Studies in Systems, Decision and Control 49

Andrey E. Gorodetskiy *Editor*

Smart Electromechanical Systems

Springer

Studies in Systems, Decision and Control

Volume 49

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Systems, Decision and Control" (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control- quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/13304

Andrey E. Gorodetskiy
Editor

Smart Electromechanical Systems

Springer

Editor
Andrey E. Gorodetskiy
Institute of Problems of Mechanical
Engineering
Russian Academy of Sciences
Saint Petersburg
Russia

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-319-27545-1 ISBN 978-3-319-27547-5 (eBook)
DOI 10.1007/978-3-319-27547-5

Library of Congress Control Number: 2015957234

Printed on acid-free paper

This Springer imprint is published by SpringerNature
The registered company is Springer International Publishing AG Switzerland

Active Data in Digital Software Defined Systems Based on SEMS Structures

V.V. Alexandrov, S.V. Kuleshov and A.A. Zaytseva

Abstract The article proposes the further development of terminal programs approach (the programs which being transmitted instead of data and executed on receiver side are restoring the data to be transmitted) to the concept of active data providing automatic reconfiguration of software defined equipment needed for self propagation control through communication environment. Active data being terminal programs are configuring the software defined equipment and controlling self propagation through communication environment.

Keywords Terminal program · Active data · Infocommunication · Software defined · Homoiconicity

1 Introduction

Modern tendencies of software defined and software configurable systems development [1] are intensifying the growth of new data representation and processing methods. The paper considers the organization and control of the adaptive communication environment if a workframe of digital programmed infocommunication [2].

For this purpose the paper proposes the further development of terminal programs approach (the programs which being transmitted instead of data and executed on receiver side are restoring the data to be transmitted) [3, 4] to the concept of

V.V. Alexandrov (✉) · S.V. Kuleshov · A.A. Zaytseva
St. Petersburg Institute for Informatics and Automation of RAS,
Russian Academy of Sciences, Saint Petersburg, Russia
e-mail: alexandr@iias.spb.su

S.V. Kuleshov
e-mail: kuleshov@iias.spb.su

A.A. Zaytseva
e-mail: cher@iias.spb.su

A.E. Gorodetskiy (ed.), *Smart Electromechanical Systems*,
Studies in Systems, Decision and Control 49,
DOI 10.1007/978-3-319-27547-5_6

active data. Active data (AD) being the terminal programs are configuring the software defined equipment and controlling self propagation through communication environment.

Providing the terminal programs with ability to take active actions not only on receiver side but also on intermediate nodes taking part in communication process will expand the abilities of data transmission networks converting them into software defined systems (the development of SDR—Soft Defined Radio principles).

This allows the real time change of data transmission formats, frequency ranges, modulation types, radio network topologies which in turn provides the ability to dynamically form the special data transmission networks from a general purpose devices temporarily reconfiguring them for data transmission task between transmitter and receiver beyond radio visibility range.

2 Terminal Programs

The example of such terminal program is a self extracting archive (executive SFX archive) despite the fact that such archive file is not a strictly terminal program it demonstrates its external behavior. In SFX archives the decompressing program is the same for any input data which in turn are input for decompressing program but are stored in the same file.

In programming the notion of homoiconicity exist—the feature of some languages to similarly represent executable code and data. This allows to treat data as executable and vice versa [5].

Unfortunately, this possibility of languages traditionally used only to facilitate metaprogramming, super-compilation [6] or the runtime virtualization techniques (virtual machines).

Homoiconicity in infocommunication tasks increases the flexibility of programmed channels by the ability to transmit decoding programs via same channels as the compressed data including the possibility to transmit individual decoding program for each transmitted data block.

In addition to mentioned above, the papers [7] are known to propose dynamically reconfigurable nodes based on automation theory.

3 From Terminal Programs to Active Data

The principle of digital content separation to the transport (initializing) stream and the generating program [8, 9] enables flexible adaptation of the content to the existing features and limitations of the physical transmission channels.

Within the active data conception the decoding program can be generated on transmitter side for every data type to be sent and be transmitted before initializing stream. If predetermined standard data types are to be used (in this case, on the

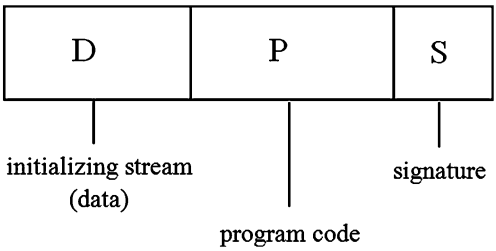

Fig. 1 The format of active data packet (container)

receiving side, there is a set of standard data recovery programs needed) it is possible to transfer only the index of the program required for recovery of digital information object.

The approach of software defined systems being configured in accordance to demands and specifics of the transmitted active data enables to create flexible virtual communication environment.

The single packet of active data can be described as a bit structure containing three components (Fig. 1): signature S, program P and initializing stream D. The only mandatory component is the signature which is needed for the active data packet (ADP) identification and the program to be executed on receiver side. Initializing stream (based on terminology in [8]) is an input data for the program P and is being transmitted to ADP only if it is needed.

In graphic description of active data behavior the designations are as shown on Fig. 2. The term of runtime environment is referring to software and hardware resources of the network node.

The basic operations for the active data processors are register operations, logical and arithmetical operations and conditional jumps. Besides, AD need access to input and output buffer memory (for the network devices). Code self modification needs also to be allowed.

The basic recommended libraries are: compression and noise immunity codecs, encryption/decryption functions.

Every device should be able to return the list of available functions and libraries to the program.

For the correct sequencing of functions to be executed from the arriving AD packets any program is advised to make all necessary checks and actions in advance

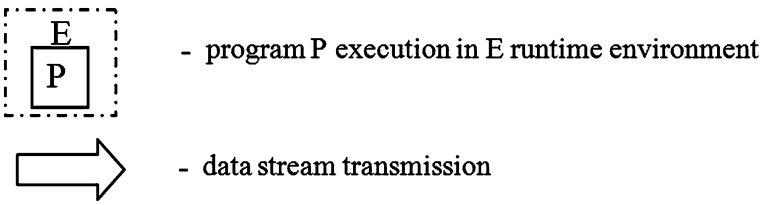

Fig. 2 Employed designations

to provide execution of its main function. This means inability for some programs to redefine available software and hardware in advance for the next arriving packets.

4 The Basic Scenarios for Active Data Usage

Hereby we consider the basic usage scenarios for active data without specification of ADP representation formats.

1. *Terminal program transmission*. The most simple usage scenario is sending the terminal program P to receiver and executing to obtain the data D* (Fig. 3). Due to insufficient theoretical research of such program generation for arbitrary bit streams, this scenario represents rather theoretical interest than practical data coding method.
2. *The decoder program usage.* The program P is being transmitted with initializing stream D (usually compressed bit stream) as a one AD packet where Pis a decoder program for D. The execution of decoding program on receiver side forms D* (Fig. 4).
3. *Identification and preparation of the environment E for the receiving node* (Fig. 5). The program P being transmitted to the receiving node detects the presence of decoding program P_d to restore the data D. In the presence of suitable decoder it is being executed on D to obtain the resulting D*. If the suitable decoder is absent the program P requests the decoding program from the sending node or the other nodes of communication environment.
4. *Using AD fragmentation*. In case of big data volume transmission it is possible to fragment it into the sequence of ADP each of which contains own copy of the program P and can be transmitted separately. After execution of program P on the receiving node it waits for the rest packets D_1, …, D_5 to be received and then executes on the merged sequence to obtain D* (Fig. 6).

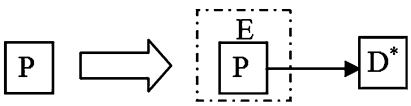

Fig. 3 Terminal program transmission

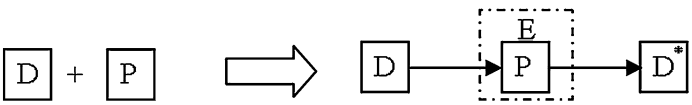

Fig. 4 The decoder program usage

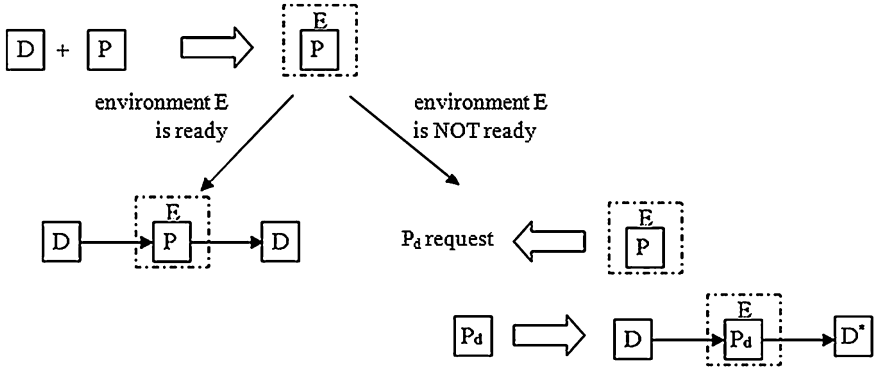

Fig. 5 Identification and preparation of the environment E for the receiving node

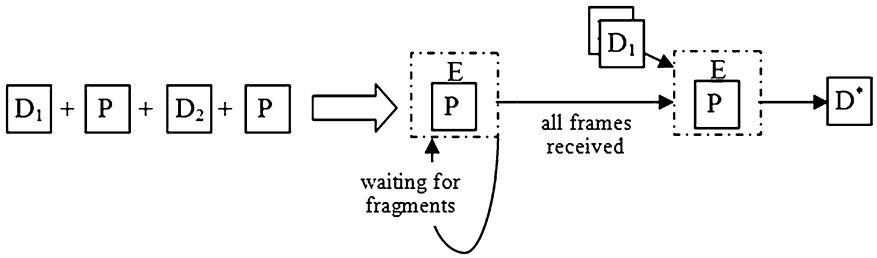

Fig. 6 Fragmentation of the AD

5. *Providing the reliability and accuracy.* When needed, the program P being executed on receiving node is able to request resending of the AD packet using the majority principle (Fig. 7).
6. *Search for the node with suitable environment for program* P *execution* (Fig. 8). This and the subsequent scenarios of AD usage are to be considered on the example of network represented as a graph where the nodes N_1, …, N_5 are the devices and the links represent physical channels.
   The node N_1 needs to transmit the AD to the receiving node N_5 given that the receiving node does not dispose the suitable environment. In this case the search for the nearest (in sense of minimizing communication expenses) node with suitable environment for execution of P (the node N_3 in this example) is

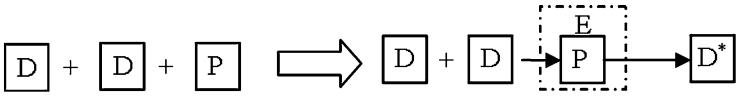

Fig. 7 Providing the reliability and accuracy

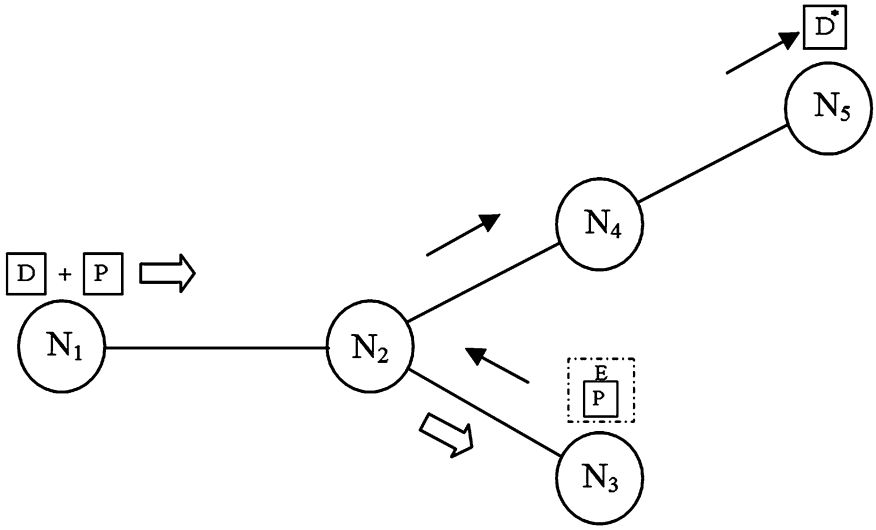

Fig. 8 Finding the node with suitable environment

performed followed by execution of P and transmission of the results D* to the receiving node N_5.

7. *Transmission of the AD using navigation functions.* When transmitting the AD via complex networks with the dynamically changing architecture (i.e. mobile systems of objects control) the function of navigation can be used by executing the program P on every node which it is passed through (Fig. 9). In this case special marks or tokens can be saved on the node containing results of delivery

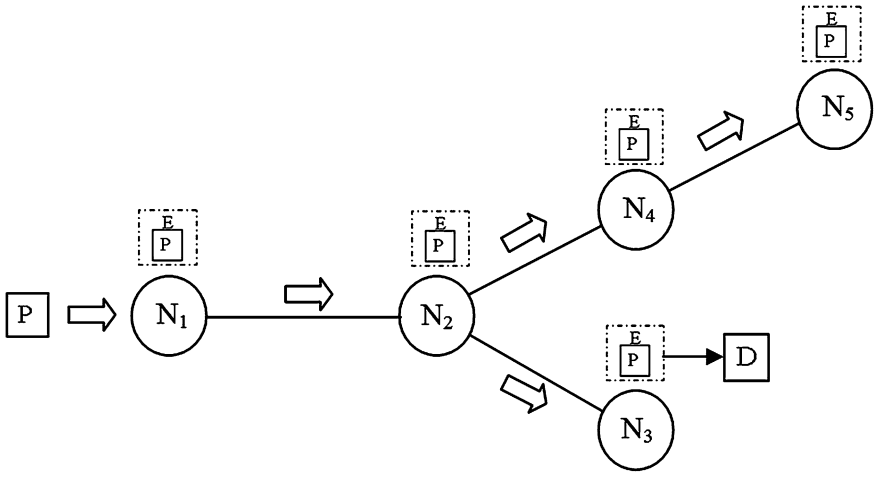

Fig. 9 The transmission of ADP to all nodes of the network

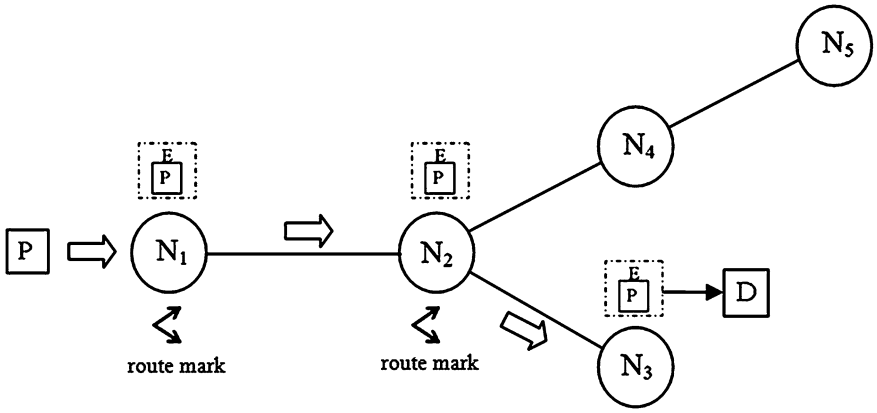

Fig. 10 The transmission of ADP using the navigation functions

of ADP to other nodes via different routes. These marks can be used by P to optimize the further delivery cost or time (Fig. 10).

8. Dynamic reconfiguration of the nodes. The AD are being transmitted from source node N_1 to node receiver N_3. The program P being executed on the node N_4 (Fig. 11) is detecting the possibility to create new communication channel to the node N_3 by reconfiguring hardware or software of this node. If the results of reconfiguration are optimal in sense of transmission time or cost then the transmission of ADP through new channel is taking place.

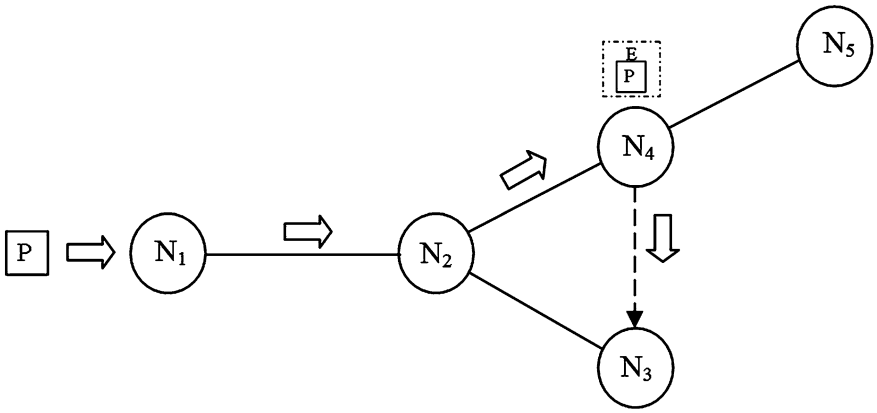

Fig. 11 Dynamic reconfiguration of the nodes

5 Infocommunication Environment and Reconfigurable Systems

Some of the considered above usage scenarios have analogies in the technologies of packet data transmission and global communication networks.

The example of the most similar approach for developing self organizing reconfigurable infocommunication networks are the mesh grids.

In contrast to mesh grids the proposed approach of active data not only operate on the level of content independent network reconfiguration but is capable of configuring networks for optimal and reliable delivery of specified content, providing network monitoring activity, unauthorized access detection etc.

6 Conclusion

The paper considers the concept of the active data transmission systems development. Being terminal programs the active data are able to configure the software defined equipment on the nodes of the network in order to control self propagation through communication environment.

The main advantage of proposed approach is an ability of fast deployment of specific communication systems on the basis of available shared facilities within the radio availability range with active data conception support.

The additional result is an increase in radio resources usage efficiency and providing electromagnetic compatibility by utilization of unused at the moment bandwidth of digital communication channels without introducing new transmitters. Also this conception increases the reliability of communication systems in general.

Further research will refine the formalization of active data description language including the demands for the set of operators necessary for active data functioning, the formats of active data representation, and standards for converting active data into executable code.

Also the equipment architecture and requirements will be proposed which includes the processor (or virtual machine) extension capable of active data code execution.

References

1. Alexandrov, V.V., Kuleshov, S.V., Tsvetkov, O.V., Zaitseva, A.A.: The conception of infotelecommunication building (the SDR prototype). Tr. SPIIRAN **6**, 51–57 (2008)
2. Alexandrov, V.V., Kuleshov, S.V., Tsvetkov, O.V.: Tsifrovaya tekhnologiya infokommunikatsii. Peredacha, khraneniye i semanticheskiy analiz teksta, zvuka, video, p. 244. SPb, Nauka (2008)

3. Alexandrov, V.V., Kuleshov, S.V.: Narrotive representation of information processes. Inf. Processes **4**(2), 160–169 (2004). www.jip.ru
4. Kuleshov, S.V.: The Terminal programs of «digital» transmission and processing of data, energy and information equivalence. Inf. Measuring Control Syst. **9** (2007)
5. Homoiconic—http://en.wikipedia.org/wiki/Homoiconic
6. Nemytykh, A.P.: about supercompilation. http://conf.nsc.ru/files/conferences/Lyap-100/fulltext/69293/69928/nemytykh_supercompilation_Lyapunov100.pdf
7. Attie, P.C., Lynch, N.A.: Dynamic input/output automata: a formal and compositional model for dynamic systems. In: Computer Science Artificial Intelligence Laboratory Technical Report, July 2013
8. Kuleshov, S.V.: Metody i tekhnologiya postroyeniya tsifrovykh programmiruyemykh infokommunikatsionnykh sistem. Diss. na soiskaniye stepeni d-ra tekhnicheskikh nauk, p. 240. SPb (2011). (Uchrezhdeniye Rossiyskoy akademii nauk Sankt-Peterburgskiy institut informatiki i avtomatizatsii RAN)
9. Kuleshov, S.V.: Hybrid codecs and application in digital programmed data transmission channels. Inf. Measuring Control Syst. **10**(5), 41–45 (2012)