An SDN-Based Approach to Design of Onboard Real-Time Networks

Vasily Balashov
Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, Russia
hbd@cs.msu.su

Valery Kostenko
Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, Russia
kostmsu@gmail.com

Tatiana Ermakova
Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, Russia
tanyaerm@mail.ru

***Abstract*—In this paper we propose an approach to use of SDN for real-time data transfer in onboard networks. This approach implements virtual links technology similar to the one used in existing AFDX and Fibre Channel onboard networks, while significantly improving run-time network reconfigurability. This improvement increases onboard system flexibility, allowing support for dynamic composition of system operation modes and seamless mode change. The approach was implemented as a functional prototype and experimentally evaluated in a virtual SDN network environment.**

Keywords—software defined networks, real-time, onboard computer systems

I. INTRODUCTION

Modern aircraft, naval and spacecraft onboard equipment complexes (OEC) typically implement either federated or integrated modular architecture [1]. According to integrated modular architecture, OEC is composed of a set of standard computational modules connected by a switched data transfer network with support for virtual links. Such network can be based on one of the following standards:

- Ethernet-based ARINC 664 (AFDX) [2];
- Fibre Channel, Real-Time profile (FC-AE-ASM-RT) [3, 4].

Hardware resources of computational modules in an integrated modular OEC can be shared between different application programs (subsystems) each of which comprises a set of application tasks.

In this paper we present an overview of AFDX and FC-AE-ASM-RT networks capabilities for performing real-time data transfer, and identify the limitations specific to these networks; we also propose an approach to implementation of real-time data transfer in switched SDN networks using elements of AFDX technology. This approach was evaluated in a virtual network environment using *RUNOS* network controller [5].

Use of SDN as onboard real-time data transfer networks can remove limitations of existing AFDX and FC-AE-ASM-RT networks severely constraining implementation of dynamic OEC operation modes, including OEC reconfiguration in case of multiple hardware failures.

II. REQUIREMENTS TO REAL-TIME DATA TRANSFER IN ONBOARD NETWORKS

During OEC design, OEC operation modes are defined. Each mode features a set of application tasks to be performed in this mode, and a set of messages to be transferred between the tasks and/or onboard sensors and actuators. On OEC mode change, the sets of application tasks and messages also change.

Following attributes are specified for each message:

1) requirements to real-time transfer;
2) size;
3) maximum jitter of message generation within its deadline interval (i.e. delay from the beginning of the interval to actual message generation time);
4) message sender node;
5) message receiver node(s).

Network nodes in an integrated modular OEC can be either computational modules or legacy devices with federated architecture.

Requirements to real-time transfer of messages are typically specified by assigning to the message its transfer frequency *F* (or period – *1/F*) and phase offsets (*φ1, φ2).* Message frequency defines the set of time intervals with lengths equal to the message period. Phase offsets define for each of these time intervals a sub-interval (s_i, f_i), called deadline interval, in which the message must be transferred (see Fig. 1).

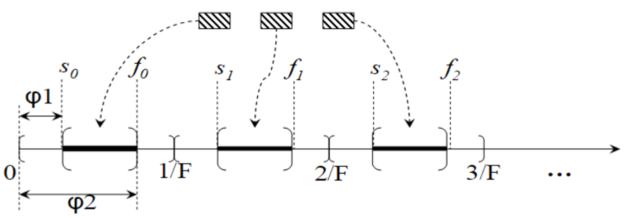

Fig. 1. Requirements to real-time transfer of messages.

A periodic message described in the way described above defines a data flow through the network.

Following constraints must be met during message transfer:

1. The message must be transferred exactly one time in a period, within the deadline interval. If the message is generated more than once in a period, its extra instances must be discarded.
2. Maximum message transfer duration from sending by the sender node to receiving by all the receiver nodes must not exceed the specified value. The message is considered received when all of its frames are received.

This work is partially supported by the Russian Foundation for Basic Research under grant № 17-07-01566.

3. Maximum message transfer jitter (difference between the maximum and minimum possible transfer durations) must not exceed the specified value.

III. Real-time Data Transfer through Switched Networks with Virtual Links

AFDX and FC-AE-ASM-RT networks used in integrated modular OECs have much common in organization of real-time data transfer. In particular, both network types use virtual links for bandwidth separation and traffic control.

In this section, the organization of data transfer in AFDX networks is described in detail, and main specifics of data transfer organization in FC-AE-ASM networks are highlighted.

An AFDX network comprises following elements (Fig. 2):

- onboard systems which generate and consume messages;
- end systems which provide interfaces between the onboard systems and the network;
- packet switches;
- network links.

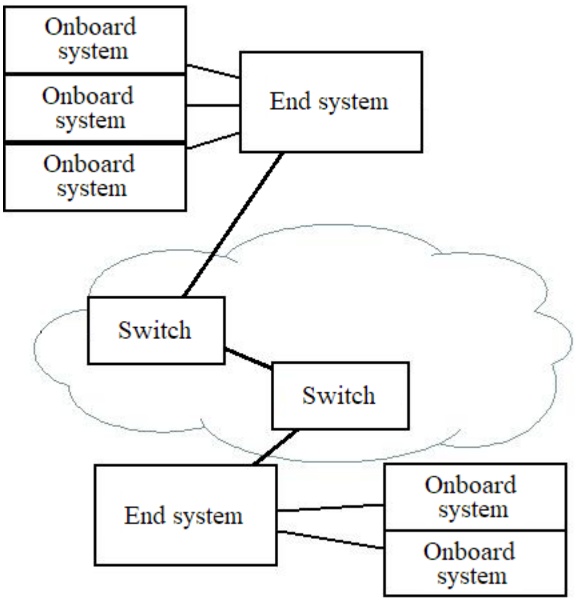

Fig. 2. Typical structure of an AFDX network.

In this paper we do not emphasize the AFDX-specific separation of onboard systems and network end systems, and assume that the network consists of nodes (which send and receive messages), switches and links.

AFDX onboard networks utilize a TCP/IP-based protocol stack. A modified Ethernet protocol is used in the data link layer, with virtual link-based (instead of MAC address-based) routing. IP protocol is used in the network layer, without routing by IP address, as routing is performed by the data link layer. UDP protocol is used in the transport layer.

Data exchange between nodes is performed via message transfer through virtual links, routes of which are defined in advance. Each virtual link has one sender node and one or more receiver nodes. Use of virtual links in AFDX networks provides bandwidth sharing and separation between different data streams, as well as guarantees meeting of constraints on duration of messages transfer through the network.

The sender node splits a message into frames of specified maximum size (or smaller), which are then sent to the physical network link. Virtual link number is written in the MAC address field of every frame. After delivery to the receiver node, the frame is buffered for farther assembly of the message. The assembled message is transferred to the application task(s) running on the node.

Data transfer reliability is provided in AFDX networks by means of redundant network hardware. Each node is connected to two independent and identical AFDX networks. Frames are sent to both networks (going then by identical routes), and if a transfer error, e.g. frame checksum error, is detected in one of the networks, the frame is taken from the network in which there was no error. The receiver node checks the frames integrity, and if a correct frame was already received from a network, its duplicate from the other network is discarded.

Routing tables of an AFDX switch (containing the information on virtual links) are set up in advance, before the start of OEC operation. Besides frame routing, AFDX switches perform traffic control and filtering. Integrity of frames and their sequence for each virtual link are checked during traffic filtering. Traffic control provides the bandwidth required for a virtual link, as well as prevents exceeding the bandwidth. The bandwidth control for an AFDX virtual link is performed using a token bucket algorithm; if a frame arrives to a switch through a virtual link and there is insufficient credit for this virtual link, the switch discards the frame. The credit replenish rate is determined by the virtual link bandwidth.

Bandwidth and route for every virtual link are set up before start of OEC operation. Therefore, routing in AFDX network is defined statically, and dynamic (in OEC run time) modification of routing tables is not provided by this network standard.

Main difference of FC-AE-ASM-RT from AFDX network regarding traffic control is that the control is performed on the message level, in contrast to the level of individual frames. FC-AE-ASM-RT networks also do not implement "sparse" sending of virtual link frames into network with intervals of BAG_{vl} (introduced later); frames of a message are sent to the FC network without pauses.

FC-AE-ASM-RT networks provide limited support for run-time reconfiguration: several routing tables can be loaded into every switch, and changing between them is supported with temporary (for several dozen of milliseconds) interruption of data transfer through the network.

A virtual link *vl* in an AFDX network is described by the following attributes [2, 6]:

- LM_{vl} – maximum size of a frame transferred through this virtual link;
- BAG_{vl} – minimum time interval between sending of sequential frames in case of zero frame generation jitter; according to the AFDX specification, this

value (in ms) must be a power of two, in the range from 1 to 128;

- JM_{vl} – maximum jitter of frame generation in the sender node;
- S_{vl} – sender node for the frames of the virtual link;
- $D_{vl} = \{d_{vl}\}$ – set of receiver nodes for the frames of the virtual link ($S_{vl} \notin D_{vl}$);
- $Tree_{vl}$ – route for frames' transfer through the network; the route is a tree with root in S_{vl} and leaves in all members of D_{vl} ;
- $MSG_{vl} = \{msg\}$ – set of messages transferred through the virtual link; the messages are generated by S_{vl} node and received by nodes from D_{vl} .

A set of virtual links for an AFDX network must meet the following constraints:

- total bandwidth reserved for all virtual links going through a specific physical link *e* must not exceed its bandwidth:

$$\sum_{vl \in e} \frac{LM_{vl}}{BAG_{vl}} \le R_e \quad (1)$$

where R_e is the bandwidth of the physical link *e*;

- the frame sending frequency necessary for sending the messages through the virtual link must not exceed the maximum frame sending frequency for this link:

$$\forall vl \in VL: \sum_{msg \in MSG_{vl}} \left(\left\lceil \frac{size_{msg}}{LM_{vl} - c} \right\rceil / T_{msg} \right) \le \frac{1}{BAG_{vl}} \quad (2)$$

This constraint is derived from the fact that all frames of one instance of a message *msg* must be sent to the network before sending of the next message instance begins, i.e. in time no greater than the message period T_{msg} . Considering that *msg* is split into $\lceil size_{msg} / (LM_{vl} - c) \rceil$ frames, where *c* is the size of service headers in the frame, we get the constraint (2).

- maximum frame sending jitter on the sender node must not exceed the specified limit (0.5 ms for AFDX networks):

$$\forall vl \in VL: JM_{vl} \le 0.5 \quad (3)$$

- maximum jitter and maximum duration of message transfer must not exceed the specified limits (determined by OEC operation logic rather than the network standard):

$$\forall msg \in MSG \; Dur(msg) \le \tau_{msg}; Jit(msg) \le J*_{msg} \quad (4)$$

where $Dur(msg)$ and $Jit(msg)$ are the message transfer duration and jitter, respectively.

Similar constraints apply to FC-AE-ASM-RT networks, with the distinction that for these networks they are introduced on the level of whole messages, not individual frames.

Several approaches to design of AFDX virtual link systems are known [6, 7] which take a set of messages as an input and construct a set of virtual links which meets the constraints (1) – (4) and guarantees message transfer within the real-time constraints 1 – 3 from Section II. These approaches estimate message transfer duration and jitter with such techniques as network calculus [8] and trajectory approach [9]. The forementioned approaches and techniques can be tailored for FC-AE-ASM-RT networks.

IV. Limitations on Dynamic Composition of OEC Operation Modes Imposed by Use of AFDX and FC-AE-ASM Networks

An example of OEC operation mode change is switching of an aircraft OEC from pre-flight preparation mode to take-off mode. A routine response of OEC to a single hardware failure (e.g. that of a computational module, switch, or network link) also includes a change into one of predefined failure modes, in which the failed component is not used, and its workload is redistributed to intact components.

Dynamic (during OEC operation) composition of the OEC operation mode is required in certain circumstances, for instance:

- in case of a new goal for the system under control: corresponding application tasks are to be loaded into OEC and started *without* interruption of OEC operation;
- in case of multiple OEC hardware failures.

Both cases are specifically relevant for autonomous vehicles with long uninterrupted operation time, such as spacecraft and unmanned air vehicles.

Use of AFDX and FC-AE-ASM-RT networks imposes following limitations on composition and change of OEC operation modes:

1. Dynamic composition of OEC operation modes cannot be supported. Only static modes are allowed, in which the sets of application tasks and messages (including tasks distribution to modules) are defined in the design or upgrade phase.
2. Flexible OEC reconfiguration in response to multiple hardware failures is impossible.
3. Seamless change between static modes is possible only if the network bandwidth is reserved for virtual links of all modes simultaneously (in a single routing table on every switch). A mode change is seamless if it does not interrupt the transfer of messages that must be transferred in both the old and the new mode.

Let us explain why the limitations listed above take place.

OEC mode change leads to a change of the executed application tasks set and the transferred messages set; the set of virtual links used for messages transfer also changes. For AFDX networks, which do not support run-time change of routing tables in switches, bandwidth must be reserved for virtual links of all modes at once, as if it was required to support simultaneous operation of virtual links from all modes. This severely limits the total bandwidth of virtual links for all supported modes and leads to inefficient utilization of physical network bandwidth. FC-AE-ASM-RT switches support run-time change between several pre-loaded routing tables, so that different tables can correspond to different modes. Meanwhile during the change of OEC operation mode (and thus of routing tables in all or some switches) the connection is temporarily lost between application tasks which operate in both old and new modes.

To support dynamic composition of OEC modes, it is necessary to include in the routing tables the information on all virtual links that can be used in any of potentially possible modes.

If an OEC mode is dynamically composed due to arrival of new application tasks and data flows, it is impossible to pre-load the information on the corresponding virtual links into switches, as this information is not available at the time of OEC operation start. Therefore, in this case dynamic composition of operation modes in OECs with AFDX or FC-AE-ASM-RT networks is impossible.

In case of computational module failure in an integrated modular OEC, it is possible to redistribute its tasks between remaining modules (task migration). In this case, the routes are changed for virtual links connecting these tasks. Support for task migration in AFDX networks leads to exceptionally inefficient bandwidth utilization, as the bandwidth must be reserved multiple times for a virtual link connecting the same pair of tasks – once for each of routes corresponding to possible bindings of tasks to modules. For FC-AE-ASM-RT networks it is possible to pre-calculate routing tables for all single module failures; meanwhile, support for multiple (double and more) failures leads to rapid growth of the number of possible task migrations which leads to impossibility to pre-load all the necessary routing tables into the switches. Requirement to tolerate failures of network devices (switches, links) only makes the things worse. Thus, for AFDX or FC-AE-ASM-RT based OECs with over 10 components it is impossible to support dynamic composition of OEC operation modes in order to respond to multiple hardware failures.

Even for static modes only, seamless OEC mode change requires simultaneous reservation of bandwidth for virtual links of all modes for which the seamless change is required. For AFDX networks, it is the only way to support mode change; FC-AE-ASM-RT networks support change between pre-loaded routing tables during mode change, which leads to temporary interruption of data exchange between OEC modules, and cannot be considered seamless.

V. Real-Time Data Transfer through Virtual Links in an SDN Network

To eliminate the limitations described in the Section IV, imposed by use of AFDX and FC-AE-ASM-RT networks in OEC, it is reasonable to use SDN instead of these networks. The rationale for SDN usage is that these limitations are caused by constrained (for FC-AE-ASM-RT) or completely absent (for AFDX) support for run-time change of network configuration, i.e. the set of virtual links and their attributes, including routes through the network. From network reconfiguration point of view, the key features of SDN are the support for dynamic, during the network operation, re-programming of switches, and presence of a network controller, the applications on which can be responsible for this re-programming.

Authors of this paper propose an approach to use of SDN as onboard networks, built upon an SDN-based implementation of real-time data transfer, and characterized by following features:

- use of virtual links for traffic control and sharing the network bandwidth between data flows;
- organization of traffic control in virtual links by analogy with AFDX and FC-AE-ASM-RT networks;
- preservation of applicability of existing approaches to message transfer latency and jitter estimation, developed for AFDX networks;
- network controller operation in passive mode;
- support for dynamic change of the virtual links set (including their routes and other attributes) without interrupting message transfer through unmodified virtual links.

Use of the existing traffic control scheme, practically proven in AFDX and FC-AE-ASM-RT networks, aims to simplify upgrade of existing OECs and development of new OECs on the base of existing ones, as well as to allow application of known approaches to network design verification against real-time requirements to data transfer.

Passive mode of controller operation guarantees absence of auxiliary network traffic between the switches and the controller, except for traffic necessary for switch (re)programming. This is essential both for network load minimization and for data transfer timings predictability, as there is no need to modify existing techniques for latency and jitter estimation to incorporate the influence of auxiliary traffic. Choice between static and dynamic controller operation modes is considered in detail in [10].

The proposed scheme for SDN usage utilizes TCP/IP protocol stack without routing by IP address. The message to be transferred through a virtual link is split into UDP packets, each of which fits in a single frame of data link layer (e.g. Ethernet). These packets are sent into the network by the sender node according to the scheme used in AFDX, with intervals equal the BAG_{vl} attribute of the virtual link. A service field located in the beginning of the packet's data area contains the virtual link number. The packet is routed by the switches based on the virtual link number, as in AFDX and FC-AE-ASM-RT networks. To perform the routing, the switches are loaded with rules that, depending on the value of the respective field of incoming UDP packet, send the packet to specified output ports of the switch. Packets of size exceeding the specified limit for the virtual link, or arriving to a physical port not assigned to the virtual link, are discarded by the switch.

Control of traffic's real-time properties is performed by meters – a technology introduced in OpenFlow 1.3 [11]. A meter is associated to a flow; in our case, a flow is defined as a sequence of received packets with a specified virtual link number, i.e. packets of a virtual link. A set of flow rate ranges is defined for each meter in a switch. Depending on the range to which the current measured flow rate belongs, one of specified actions is performed with a received packet, e.g. farther transmission or discarding.

The technique for flow rate calculation is not defined in OpenFlow 1.3 and depends on implementation of meters in a specific switch. The data exchange scheme proposed in this paper requires the switch to support flow rate calculation using token bucket algorithm similar to the one used in AFDX. This enables not only to control the flow rate for not exceeding the virtual link bandwidth limit, but also to control the packet transfer jitter. An example of a switch supporting the required technique for flow rate calculation is the *Ofsoftswitch13* software switch.

Meters are set up on the switches by means of OpenFlow protocol in the following way. For every switch, for every virtual link passing through the switch:

- associate a meter to a virtual link by a *FlowMod* message;
- pass the maximum credit value and the value of LM_{vl} / BAG_{vl} to the meter by a *MeterMod* message (in its *burst_size* and *rate* fields, respectively).

According to the proposed approach, a virtual link in SDN has the same set of attributes and the same set of constraints as an AFDX virtual link (see Section III). As the scheme for sending packets to the network from a node and the traffic control scheme for a switch are inherited from AFDX, the techniques for data transfer delay and jitter estimation [8, 9] developed for AFDX are applicable within the proposed SDN-based approach, as well as the methods for synthesis of virtual link sets [6].

VI. Evaluation of the Proposed Approach

The authors performed an experimental evaluation of the proposed approach to real-time data exchange in SDN networks. The experiments were performed in a network simulation environment based on *mininet*, with *RUNOS* as the network controller and *Ofsoftswitch13* as software switches.

A service application was implemented for the *RUNOS* environment that performs programming the switches to support the specified set of virtual links, and can remove/add virtual link records from/to the switches. The experiment scenarios were implemented as network controller applications that invoke the service application in order to set up and modify specific sets of virtual links in the switches.

The goals of experimental evaluation were as follows:

- estimate the delays and jitters of message transfer through the network, confirm their accordance to the upper estimates acquired by techniques developed for AFDX networks;
- confirm that the proposed data exchange scheme allows transfer of multiple data flows with guaranteed bandwidth for each of them;
- confirm that attempts by a network node to send data through a virtual link with a rate exceeding the bandwidth limit are blocked by the switch;
- perform dynamic modification of the set of virtual links according to different scenarios typical for OEC mode changes, and confirm that data transfer through unmodified virtual links is not interrupted.

The number of virtual links and values of their attributes were taken from the paper [6] on design of AFDX networks. The number of virtual links in the experiments reached several dozen.

Both network topologies from [6] and those typical for integrated modular OECs were considered. Examples of the latter are a multiple star (Fig. 3) and a topology with connection of every node to two switches (Fig. 4).

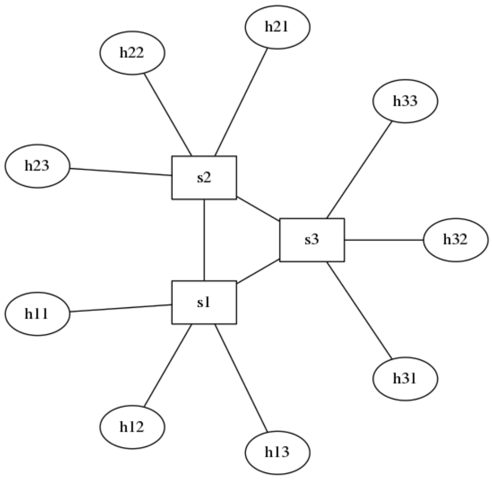

Fig. 3. Multiple star network topology.

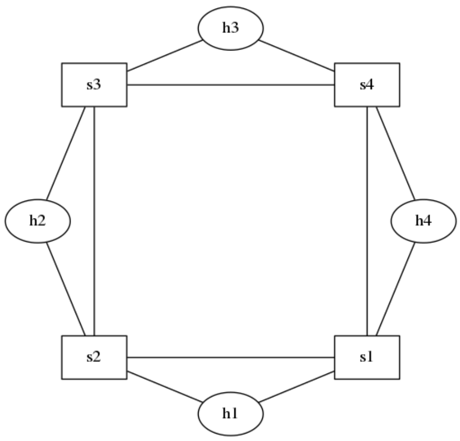

Fig. 4. Network topology with every node connected to two switches.

The performed experiments indicated that for the considered data flows and network topologies the proposed SDN-based data exchange scheme guarantees the required bandwidth for every data flow, with delay and jitter within estimated bounds. In the experiments that involved network nodes generating traffic with oversized UDP packets and/or too small inter-packet intervals, the switches discarded incorrect packets of respective virtual links, thus preventing violation of bandwidth constraints.

Following scenarios of dynamic modification of virtual links sets were included in the experiments:

1) “regular” OEC mode change: some virtual links remain intact, others are removed, and new ones are added instead;

2) task migration, e.g. in case of computational module failure: virtual links leading to one of the nodes change their routes to lead to another node;

3) network reconfiguration in case of switch or physical link failure: virtual links passing through the failed component change their routes to bypass that component.

In all the listed scenarios, new routes of virtual links were specified in the input data. On-the-fly synthesis of new routes is a matter of farther research.

Experiments with the scenario 1 indicated that during removal/adding of virtual links the data exchange through other links was not interrupted. There was also no data exchange interruption for unmodified virtual links in the experiments with scenarios 2 and 3.

Main distinction of the scenarios 2 and 3 from the scenario 1 is that sending of data to the virtual links being modified did not stop. Re-programming the switches for a new virtual link route started from the first switch (counting in the direction of data transfer) after which the new route differs from the old one. The experiments indicated that during re-programming of the switches for a new virtual link route only several packets were lost; the number of lost packets depends on packet send frequency, defined by BAG_{vl} value. There was no data transfer interruption in other virtual links.

The conclusion from the experiment results is that the proposed approach to SDN-based design of onboard real-time networks preserves timing predictability characteristic for AFDX and FC-AE-ASM-RT networks, and removes the limitations of these networks regarding composition and change of OEC operation modes.

VII. Necessary Adaptation of SDN Technology for Onboard Use

OEC is a critical counterpart of the controlled system, such as an aircraft or a spacecraft. OEC hardware, including network components, is subject to strict requirements to operation in harsh environmental conditions, such as high temperature, vibration, radiation. To meet these requirements, it is necessary to implement SDN controllers and switches using technologies with no lesser tolerance to such conditions than other OEC components, including computational modules. It should be noted that this condition could be automatically fulfilled by using the resources of regular computational modules of OEC to implement software switches and controllers. However, the transition to hardware SDN switches will raise again the issue of reliability.

To use the SDN technology in OEC of such vehicles as cars and passenger aircraft, hardware and software counterparts of SDN, as well as processes of their development, must be qualified according to respective industry standards, such as DO-178C [12] for aircraft.

VIII. Conclusion

In this paper, we analyzed the virtual link based approach to real-time data transfer in onboard networks implemented in AFDX and FC-AE-ASM-RT networks. Major limitation of these networks is their limited or lacking support for dynamic reconfiguration of virtual channels, leading to impossibility to dynamically compose operation modes for an onboard equipment complex (OEC), particularly in response to multiple hardware failures.

We propose an approach to use of SDN as onboard real-time networks to remove this limitation. The approach is based on implementation, by means of SDN and OpenFlow 1.3, of a virtual link technology similar to one used in AFDX and FC-AE-ASM-RT. The approach was implemented as a functional prototype and experimentally evaluated in a virtual network environment based on *Ofsoftswitch13* software SDN switches and *RUNOS* network controller. The experiments indicated that the proposed data exchange scheme allows transfer of messages within specified constraints on delay and jitter, as well as prevents violation of bandwidth constraints for virtual links. The experiments also confirmed that dynamic reconfiguration of virtual links in the SDN network does not interrupt data transfer through unmodified virtual links.

Directions for future work include:

- development of algorithms for dynamic synthesis of new routes for virtual links during OEC reconfiguration, particularly in response to multiple failures of computational modules and network components;
- implementation of these algorithms in a network controller application;
- integration of the application with tools for OEC hardware status monitoring, including network monitoring tools.

The unifying goal for the work is to create an SDN-based network technology for onboard use, which provides both real-time data transfer and automatic network reconfiguration during OEC mode change, including support for dynamically composed OEC operation modes.

References

[1] T. Gaska, C. Watkins, and Y. Chen, “Integrated Modular Avionics – Past, present, and future”, IEEE Aerospace and Electronic Systems Magazine, vol. 30, No. 9, pp. 12–23, 2015.

[2] Aircraft Data Network. Part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network: Aeronautical Radio, Inc., 2012.

[3] INCITS 373. Information Technology – Fibre Channel Framing and Signaling Interface (FC-FS): International Committee for Information Technology Standards, 2003.

[4] Y. Osipov, A. Pershin, Y. Pustovoy, “Method for Real-Time Information Transmission Using Small-Scale Local Area Networks Based on FC-AE-ASM Protocol Modification,” Patent RU2536659C1, NTC GREK, JSC, 2013.

[5] A. Shalimov, S. Nizovtsev, D. Morkovnik, and R. Smeliansky., “The RUNOS OpenFlow Controller,” Proc. 4th European Workshop on Software Defined Networks, pp. 103–104, IEEE, 2015.

[6] P. Vdovin and V. Kostenko, “Organizing Message Transmission in AFDX Networks,” Programming and Computer Software, vol. 43, No. 1, pp. 1–12, 2017.

[7] A. Al Sheikh, O. Brun, and P.-E. Hladik, “Optimal Design of Virtual Links in AFDX Networks,” Real-Time Systems, vol. 49, No. 3, pp. 308–336, 2013.

[8] M. Boyer and C. Fraboul, “Tightening End to End Delay Upper Bound for AFDX Network Calculus with Rate Latency FIFO Servers Using Network Calculus,” Proc. IEEE International Workshop on Factory Communication Systems, pp. 11–20, IEEE, 2008.

[9] H. Bauer, J. L. Scharbarg, and C. Fraboul, “Applying and Optimizing Trajectory Approach for Performance Evaluation of AFDX Avionics

Network," Proc. IEEE Conference on Emerging Technologies and Factory Automation, pp. 1–8, IEEE, 2009.

[10] V. Balashov, V. Kostenko, P. Vdovin, R. Smeliansky, and A. Shalimov, "An Analysis of Approaches to Onboard Networks Design," Proc. 2014 International Science and Technology Conference on Modern Networking Technologies, pp. 20–24, IEEE, 2014.

[11] OpenFlow Switch Specification, Version 1.3.0: Open Networking Foundation, 2012.

[12] DO-178C Software Considerations in Airborne Systems and Equipment Certification: RTCA, Committee SC-205, 2011.