

## ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

УДК 681.3

### ПОСТРОЕНИЕ БОРТОВЫХ КОММУТИРУЕМЫХ СЕТЕЙ МИНИМАЛЬНОЙ СЛОЖНОСТИ

© 2021 г. В. А. Костенко<sup>а,\*</sup>, А. А. Морквин<sup>а,\*\*</sup>

<sup>а</sup> *Московский государственный университет имени М.В. Ломоносова,  
119991 Москва, Ленинские горы, д. 1, Россия*

*\*E-mail: kost@cs.msu.su*

*\*\*E-mail: mr.andrej1102@yandex.ru*

Поступила в редакцию 26.10.2020 г.

После доработки 20.01.2021 г.

Принята к публикации 26.01.2021 г.

В статье сформулирована задача построения бортовой коммутируемой сети минимальной сложности необходимой для передачи периодических сообщений в реальном времени и предложены алгоритмы ее решения: построения структуры сети и системы виртуальных каналов. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей AFDX, предназначенных для передачи исходно заданного набора периодических сообщений.

DOI: 10.31857/S013234742104004X

#### 1. ВВЕДЕНИЕ

Современные информационно управляющие системы реального времени (ИУС РВ) являются распределенными и включают в свой состав: вычислительные ресурсы, датчики, контроллеры исполнительных устройств, устройства хранения и отображения информации, которые взаимодействуют между собой. В ИУС РВ можно выделить три уровня обработки данных: 0) уровень предобработки, 1) уровень первичной обработки, 2) уровень вторичной обработки. Для обеспечения выполнения функциональных программ в режиме реального времени наибольшая производительность и пропускная способность сети необходима на уровне первичной обработки данных.

В настоящее время осуществляется переход от ИУС РВ с федеративной архитектурой к ИУС РВ с интегрированной модульной архитектурой. Наиболее широко используемый подход к построению ИУС РВ с интегрированной модульной архитектурой известен как интегрированная модульная авионика (ИМА). Разработан ряд стандартов, регламентирующих построение ИУС РВ с архитектурой ИМА:

1. ARINC 651 — основные принципы построения ИУС РВ на основе ИМА [1].
2. ARINC 653 — спецификация операционных систем [2].
3. FC-AE-ASM-RT — спецификация сети информационного обмена на основе коммутируемой сети Fibre Channel [3].

4. ARINC 664 (AFDX) — спецификация сети информационного обмена на основе Ethernet [4].

Стандарт ARINC 651 регламентирует основные принципы построения ИУС РВ на основе ИМА. В соответствии с этим стандартом, единый бортовой вычислитель строится из набора стандартизованных вычислительных модулей.

Стандарт ARINC 653 регламентирует построение операционных систем, которые позволяют обеспечить выполнение программ в реальном времени и изоляцию программ различных бортовых подсистем при выполнении на едином вычислителе. Изоляция распространяется на все ресурсы, включая регистровую память, кэши центральных процессоров, шины (порты) ввода-вывода. Изоляция программ различных подсистем ИУС РВ обеспечивается введением разделов и окон. Для программ каждой бортовой подсистемы выделяется свой раздел и набор временных окон (непересекающихся интервалов времени). Расписание закрытия и открытия окон строится предварительно, до начала работы системы. Программы раздела могут выполняться только в рамках своих окон и каждому разделу выделяется необходимая память, к которой не могут обращаться программы других разделов. Программы раздела внутри окна запускаются на выполнение по мере готовности данных, в соответствии с приоритетами. Допустимо прерывание программы и ее последующее выполнение в этом окне или в одном из следующих окон раздела. Программы

различных разделов могут взаимодействовать лишь путем передачи сообщений.

Стандарт FC-AE-ASM-RT регламентирует построение сетей обмена ИУС РВ на основе Fibre Channel. Стандарт ARINC 664 (AFDX) используется при построении бортовых сетей обмена летательных аппаратов гражданской авиации. Базовая топология: коммутируемая сеть на основе стандарта Ethernet 802.3. Передача сообщений в режиме реального времени достигается введением виртуальных каналов и механизма контроля трафика по каждому каналу. Характеристики виртуальных каналов и их маршруты определяются при построении бортовой сети таким образом, чтобы гарантировано обеспечить передачу периодических сообщений по каждому виртуальному каналу в режиме реального времени, то есть сообщение должно обязательно передаваться один раз в каждом периоде.

В работе [5] на примере локационной системы с фазированными антенными решетками показано, что переход от федеративных архитектур к архитектурам ИМА (то есть при переносе программ первичной обработки с вычислителя системы на единый бортовой вычислитель) приводит к увеличению потока данных в бортовой сети обмена в  $10^3$ – $10^5$  раз в зависимости от характеристик локационной системы. Это обуславливает актуальность построения бортовых сетей обмена минимально необходимой сложности.

В данной работе сформулирована задача построения бортовой коммутируемой сети минимальной сложности необходимой для передачи периодических сообщений в реальном времени и предложены алгоритмы ее решения: построения структуры сети и системы виртуальных каналов. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей AFDX.

## 2. ЗАДАЧА ПОСТРОЕНИЯ БОРТОВОЙ СЕТИ ОБМЕНА НА ОСНОВЕ СТАНДАРТА AFDX

В работе [6] сформулирована задача построения системы виртуальных каналов для заданной бортовой сети обмена и предложен алгоритм её решения. В данной работе формулируется задача построения бортовой коммутируемой сети минимальной сложности необходимой для передачи периодических сообщений в реальном времени: построения структуры сети и системы виртуальных каналов.

Пусть даны следующие множества:  $N$  – множество точек размещения оконечных систем,  $K$  – множество возможных точек размещения коммутаторов,  $E_{sw}$  – множество возможных дуг между коммутаторами,  $E_{end}$  – множество возможных дуг между оконечными системами и коммутаторами,

$V$  – множество весов дуг (длина соединения). Далее под соединением будем понимать соответствующую дугу. Для каждой дуги<sup>1</sup>  $e \in E_{sw} \cup E_{end}$  также задана пропускная способность  $R_e$ . Для каждой оконечной системы  $n \in N$  задано множество абонентов  $A_n$ , которые к ней подключены (один абонент может быть подключен только к одной оконечной системе).

Нагрузка на сеть задается *множеством периодически передаваемых сообщений MSG*, в котором каждое сообщение характеризуется следующими параметрами:

1.  $T_{msg}$  (мс) – период передачи сообщения.
2.  $size_{msg}$  (байт) – размер сообщения.
3.  $J_{msg}$  (мкс) – максимальный джиттер порождения сообщения внутри периода, то есть максимальный интервал времени от начала периода, в котором может быть порождено сообщение.
4.  $src_{msg}$  – абонент-отправитель.
5.  $\{dst\}_{msg}$  – множество абонентов-получателей, подключенных к оконечным системам.

Для передачи сообщений в реальном времени должны выполняться следующие условия:

1. Сообщение должно передаваться не менее одного раза в период.
2.  $t_{msg}$  (мс) – максимальная длительность передачи сообщения с момента выдачи от абонента оконечной системе-отправителю до момента получения всеми абонентами-получателями.
3.  $J_{msg}^*$  (мс) – максимальный джиттер передачи сообщения, то есть разность между максимальной и минимальной длительностью передачи сообщения.

*Бортовую сеть обмена* будем описывать взвешенным графом  $G^* = (N^* \cup K^*, E^*, V^*)$ , где:  $N^* \subseteq N$  – подмножество оконечных систем, абоненты которых либо являются отправителями, либо получателями сообщений;  $K^* \subseteq K$  – подмножество коммутаторов таких, что через каждый коммутатор проходит как минимум один маршрут передачи данных;  $E^* \subseteq E_{sw} \cup E_{end}$  – дуги, по которым проходит хотя бы один маршрут передачи данных,  $V^* \subseteq V$  – веса всех дуг  $e \in E^*$

Введем понятие *меры сложности сети S для бортовой сети обмена*, как суммарную длину всех соединений:

<sup>1</sup> Согласно стандарту [4], избыточность достигается путем использования двух независимых сетей. Для каждого виртуального канала оконечная система посылает по копии данных в обе сети. В данной работе исследуется передача данных только через одну сеть, результаты для другой сети будут аналогичными.

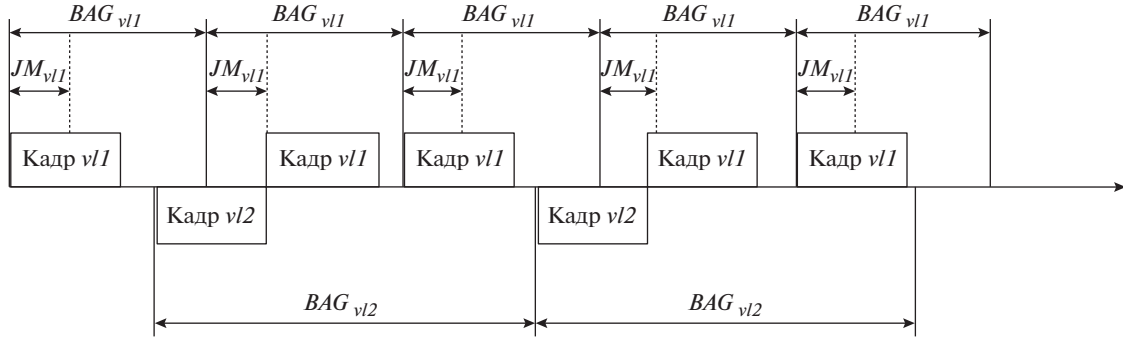


Рис. 1. Мультиплексирование виртуальных каналов с помощью джиттера.

$$S(G^*) = \sum_{e \in E^*} V^*(e)$$

Тогда под *максимальной сетью* (полным графом) будем понимать бортовую сеть обмена максимальной сложности, то есть бортовую сеть  $G$ , которая включает все коммутаторы из множества  $K$  и все дуги из множества  $E_{sw} \cup E_{end}$ .

Построить *виртуальный канал*  $vl$  значит определить для него следующие параметры:

1.  $LM_{vl}$  (байт) – максимальный размер кадра, передаваемого по данному виртуальному каналу.  $64 \leq LM_{vl} \leq 1518$  байт; размер заголовка  $c = 47$  байт; расстояние между кадрами равно  $gap = 12$  мкс.

2.  $BAG_{vl}$  (мс) – минимальный промежуток времени между передачей кадров при нулевом джиттере порождения кадров; по стандарту это значение является степенью двойки и лежит в промежутке от 1 до 128 мс.

3.  $JM_{vl}$  – максимальный джиттер порождения кадров на оконечной системе-отправителе, то есть разность между реальным временем начала выдачи кадра и временем выдачи, определенным согласно периоду. Используется при мультиплексировании виртуальных каналов.

4.  $S_{vl}$  – оконечная система-отправитель кадров данного канала.

5.  $D_{vl}$  – множество оконечных систем-получателей кадров данного канала.

6.  $Tree_{vl}$  – маршруты передачи кадров в сети.

7.  $MSG_{vl}$  – множество сообщений, передаваемых по данному виртуальному каналу и исходящих от одного абонента  $S_{vl}$ .

На рисунке 1 показано, что кадры виртуальных каналов  $vl_1$  и  $vl_2$  не могут быть переданы строго регулярно (с интервалом между кадрами, равным  $BAG$ ). При сдвиге кадров виртуального канала  $vl_1$  относительно  $BAG_{vl_1}$  на величину, не превышающую  $JM_{vl_1}$ , кадры могут быть переданы, не нарушая регулярности передачи.

При построении виртуальных каналов и маршрутов для них должны выполняться следующие ограничения:

1. Суммарная пропускная способность, зарезервированная под виртуальные каналы, проходящие через дугу  $e$ , не превосходит его пропускной способности:

$$\forall e \in E^* : \sum_{vl \in e} \frac{LM_{vl}}{BAG_{vl}} \leq R_e \quad (1)$$

2. Частота передачи кадров каждого виртуального канала не превосходит частоты выдачи кадров в канал:

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

Данное ограничение возникает из того, что все кадры одного сообщения  $msg$  должны поступить в канал до выдачи следующего сообщения  $msg$  то есть за период  $T_{msg}$ . Учитывая, что сообщение  $msg$  делится на количество кадров, равное  $\left\lceil \frac{size_{msg}}{LM_{vl} - c} \right\rceil$ .

3. Максимальный джиттер на оконечных системах-отправителях не должен превосходить 500 мкс<sup>2</sup>:

$$\forall vl \in VL : JM_{vl} \leq 0.5 \text{ мкс} \quad (3)$$

4. Максимальная длительность передачи каждого сообщения и максимальный джиттер не превосходят заданных ограничений:

$$\forall msg \in MSG : \begin{cases} Dur(msg) \leq t_{msg} \\ Jit(msg) \leq J_{msg}^* \end{cases} \quad (4)$$

Здесь  $Dur(msg)$  и  $Jit(msg)$  – процедуры вычисления оценок длительности передачи сообщений и их джиттеров соответственно.

*Решение задачи* заключается в построении минимальной бортовой сети обмена  $G^*$  и множества

<sup>2</sup> Данное ограничение было добавлено в задачу исходя из рекомендаций, описанных в стандарте [4].

виртуальных каналов  $VL$ , построенного для максимального подмножества  $MSG^* \subseteq MSG$ . Таким образом имеется две задачи условной оптимизации с ограничениями (1)–(4):

$$\begin{array}{ll} \max(|MSG^*|) & \min(S(G^*)) \\ MSG^* \subseteq MSG & G^* \in U \\ \text{ограничения (1)–(4)} & \text{ограничения (1)–(4)}, \end{array}$$

где  $U$  – множество всех подграфов максимальной сети. Поэтому в качестве первого критерия выберем максимизацию числа передаваемых сообщений, а вторым критерием выберем минимизацию сложности бортовой сети.

Из такого выбора критериев оптимизации следует наличие двух классов задач:

1. Такие входные данные задачи, что все сообщения можно разместить в максимальной сети.
2. Такие входные данные задачи, что все сообщения нельзя разместить даже в максимальной сети.

### 3. АЛГОРИТМ ПОСТРОЕНИЯ БОРТОВОЙ СЕТИ ОБМЕНА МИНИМАЛЬНОЙ СЛОЖНОСТИ

Алгоритм состоит из следующих шагов:

Шаг 1. Из входных данных создать максимальную сеть  $G$ .

Шаг 2. Для каждого сообщения из  $MSG$  создать виртуальный канал с такими параметрами  $LM$ ,  $BAG$  и  $JM$ , чтобы выполнялось ограничение (2).

Шаг 3. Для каждого виртуального канала проверить выполнение ограничения (3). Если ограничение выполняется не для всех виртуальных каналов, то воспользоваться процедурой агрегации. Данная процедура объединяет сообщения, исходящие от одного абонента, в один виртуальный канал (при этом прежние виртуальные каналы этих сообщений удаляются) таким образом, что будут выполняться ограничения (2) и (3).

Шаг 4. Для каждого виртуального канала произвести построение маршрута с учетом ограничения (1) и суммарной длины уже используемых для передачи данных соединений, где перебор виртуальных каналов будем выполнять в порядке убывания требуемой пропускной способности:

Шаг 4.1. Выполнить процедуру построения маршрута. Если маршрут найден, перейти к шагу 5.

Шаг 4.2. Выполнить процедуру ограниченного перебора для маршрутов уже назначенных виртуальных каналов.

Шаг 4.2.1. Если маршрут построен успешно, то перейти к следующему виртуальному каналу.

Шаг 4.2.2. Если маршрут построить не удалось, назначение считается неуспешным, если в виртуальном канале передается только одно сообще-

ние. Иначе из виртуального канала убирается сообщение с наибольшим значением  $LM/BAG$  (его назначение считается неуспешным), и виртуальный канал переконфигурируется с помощью процедуры агрегации. После этого для нового виртуального канала выполняется шаг 4.1.

Шаг 5. Для каждого сообщения выполняются следующие шаги:

Шаг 5.1. Произвести вычисление длительности и джиттера передачи сообщений, затем проверить ограничение (4) и, если ограничение выполняется, то перейти к следующему сообщению.

Шаг 5.2. Произвести процедуру переконфигурации виртуального канала – попробовать перенастроить параметры  $LM$ ,  $BAG$  и  $JM$  с учетом оценки полученной, на шаге 5.1. Если после выполнения процедуры, ограничение (4) выполняется, то перейти к следующему сообщению.

Шаг 5.3. Выполнить процедуру агрегации виртуальных каналов, с построением нового маршрута и проверки ограничений (1)–(4). В случае успеха прежние виртуальные каналы заменяются на новый канал. Иначе назначение сообщения считается неуспешным.

Шаг 6. Выполнить минимизацию полного графа  $G$ .<sup>3</sup>

Шаг 6.1. Для каждого соединения проверить, используется ли он в каком-либо маршруте назначенных виртуальных каналов. Если нет, то удалить это соединение из графа.

Шаг 6.2. Для каждого коммутатора проверить число соединений с другими коммутаторами. Если их число равно 0, то удалить этот коммутатор из графа.

Процедура настройки параметров виртуальных каналов [7]:

*Вход:* виртуальный канал  $vl$ .

*Выход:*  $vl$  с заданными параметрами  $LM$ ,  $BAG$  и  $JM$ .

Процедура агрегации:

*Вход:* виртуальный канал  $vl$ .

*Выход:* агрегированный виртуальный канал для  $vl$  с заданными параметрами  $LM$ ,  $BAG$  и  $JM$ , либо неуспех.

Процедура ограниченного перебора виртуальных каналов:

<sup>3</sup> На данном этапе работы алгоритма не требуется делать попытки удаления используемых в сети передачи данных соединений с последующим перестроением маршрутов виртуальных каналов, так как процедура построения маршрута учитывает не только длину соединений, но и их участие в передаче уже назначенных сообщений, то есть на каждом этапе работы процедуры сложности сети либо не изменяется, либо увеличивается на минимальную величину.

**Вход:** множество назначенных виртуальных каналов; виртуальный канал  $vl$ , для которого не удается построить маршрут.

**Выход:** виртуальный канал  $vl$  с построенным маршрутом, либо неуспех.

Процедура вычисления максимальной длительности и джиттера передачи сообщений [8–10]:

**Вход:** сообщение  $msg$ , передаваемое по виртуальному каналу  $vl$ .

**Выход:**  $Dur(msg)$ ,  $Jit(msg)$ .

Процедура переконфигурации виртуального канала:

**Вход:** виртуальный канал  $vl$  с  $msg$ , для которого не выполняется (4).

**Выход:** виртуальный канал  $vl$  с новыми параметрами  $LM$ ,  $BAG$  и  $JM$ , либо неуспех.

Алгоритмы выполнения этих процедур приведены в работе [6].

Процедура построения маршрута для виртуального канала:

**Вход:** граф  $G$ ; виртуальный канал  $vl$ .

**Выход:** виртуальный канал  $vl$  с маршрутом передачи, либо неуспех.

Шаг 1. Преобразуем граф  $G$ , убрав из него все соединения, остаточная пропускная способность которых меньше, чем требуемая пропускная способность.

Шаг 2. В полученном подграфе, запустить алгоритм Йена, который использует следующие 2 критерия:

- Критерий веса для ребра:

$$I(e) = \frac{V_e}{k+1},$$

где  $e \in E$ ,  $V_e$  – длина соединения  $e$ , а  $k$  – число уже назначенных на это соединений виртуальных каналов. Данный критерий, используется при нахождении одного маршрута алгоритмом Дейкстры.

- Критерий для сравнения маршрутов между собой:

$$I(path) = C_1 * \sum_{path} I(e) + C_2 * cost(path, vl),$$

где  $I(e)$  – критерий веса для ребра,  $cost(path, vl) =$

$$= \sum_{e \in path(S_{vl}, n)} \frac{LM_{vl}}{R_e} + \sum_{v \in path(S_{vl}, n), v \neq vl} \frac{LM_{vl}}{\max_{e \in path} (R_e)} - \text{эвристика}$$

[6], которая оценивает стоимость пути из длительности передачи кадра текущего виртуального канала по всем каналам передачи данного пути и из оценки ожиданий передачи кадров других виртуальных каналов,  $C_1, C_2 \in [0, 1]$  – коэффициенты, что  $C_1 + C_2 = 1$ .

Среди найденных  $k$  путей будем выбирать маршрут с минимальным  $I(path)$ .

Алгоритм остановим после нахождения кратчайшего пути до одной из конечных систем получателей, до которых путь еще не найден.

Шаг 3. Если найдены пути до всех получателей, то вернем объединение полученных маршрутов, которое и является маршрутом виртуального канала. Иначе перейти на шаг 2.

#### 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ СВОЙСТВ АЛГОРИТМА

Цель экспериментального исследования – проверить эффективность предложенного алгоритма на различных классах исходных данных. Критерии эффективности: суммарная длина используемых в сети передачи данных соединений и число назначенных сообщений.

Внутри каждого класса данных, описанных в разделе 1, генерировались три типа наборов сообщений:

1. Большое количество сообщений с низкими требованиями к длительности передачи. Генерировалось 1500 сообщений размером от 1 до 1000 байт с периодом от 1 до 1000 секунд и с ограничением длительности передачи от 0.1 до 100 секунд.

2. Малое число больших сообщений. Генерировалось 100 сообщений размером от 1 кб до 1 мб с периодом от 1 до 10 секунд и с ограничением длительности передачи от 1 до 10 секунд.

3. Сообщения с высокими требованиями к длительности передачи. Генерировалось 100 сообщений размером от 1 до 1000 байт с периодом от 10 до 100 мс и с ограничением длительности передачи от 1 до 10 мс.

Запуски проводились на топологиях, используемых в реальных ИУС РВ, в которые была добавлена избыточность. Например, топология из работы [11].

Для каждой топологии генерировались все классы данных со всеми подтипами. Для каждого подтипа генерировалось 100 случайных наборов, после чего результаты усреднялись. Основные результаты исследований указаны для всех классов данных на диаграмме 1.

На основе полученных результатов экспериментов можно сделать следующие выводы:

- 1) Алгоритм позволяет минимизировать исходную сеть в среднем на 23%.
- 2) Для второго класса данных алгоритм в большинстве случаев дает максимальную сеть.

#### 5. ЗАКЛЮЧЕНИЕ

Предложенный в работе алгоритм для исходно заданного множества периодических сообщений



**Диаграмма 1.** Результаты для двух классов данных (слева: 1, 2, 3 типы данных для 1 класса данных; справа: 1, 2, 3 типы данных для 2 класса данных).

осуществляет построение коммутируемой сети обмена ИУС РВ минимальной сложности необходимой для передачи сообщений в реальном времени и строит множество виртуальных каналов, по которым передаются сообщения.

Алгоритм допускает настройку на особенности коммутаторов AFDX различных производителей за счет изменения алгоритмов для расчета значений параметров виртуального канала и использования различных методов для вычисления оценки максимальной длительности и джиттера передачи сообщений.

## 6. БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке РФФИ, грант № 19-07-00614.

## СПИСОК ЛИТЕРАТУРЫ

1. ARINC 651-1 "Design Guidance for Integrated Modular Avionics", 1997.
2. ARINC Specification 653. Airlines Electronic Engineering Committee. [PDF] (<http://www.arinc.com>).
3. INCITS 373. Information Technology – Fibre Channel Framing and Signaling Interface (FC-FS), International Committee for Information Technology Standards, 2003.
4. Aircraft DataNetwork. Part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network // Aeronautical Radio, Inc. 2012.
5. Костенко В.А. Архитектура программно-аппаратных комплексов бортового оборудования // Изв. вузов. Приборостроение. 2017. Т. 60. № 3. С. 229–233.
6. Вдовин П.М., Костенко В.А. Организация передачи сообщений в сетях AFDX // Программирование. 2017. № 1. С. 5–20.
7. Al Sheikh A. et al. Optimal design of virtual links in AFDX networks. // Real-Time Systems, 2013. V. 49. № 3. P. 308–336.
8. Scharbarg J.L., Fraboul C. Methods and tools for the temporal analysis of avionic networks. // New trends in technologies: control, management, computational intelligence and network systems, 2010. P. 413–438.
9. Frances F., Fraboul C., Grieu J. Using network calculus to optimize the AFDX network // European Congress ERTS Embedded real-time software, 25–27 Jan 2006. P. 1–8.
10. Bauer H., Scharbarg J.L., Fraboul C. Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources // Real-time systems. 2012. V. 48. № 1. P. 101–133.
11. AFDX from component to application // URL: <https://www.prosoft.ru/cms/f/467416/Презентация+про+AFDX+продуктам.pdf> (15.02.2020)