

ОРГАНИЗАЦИЯ ПЕРЕДАЧИ СООБЩЕНИЙ В СЕТЯХ AFDX *

© 2017 г. Вдовин П.М., Костенко В.А.

*Московский государственный университет им. М.В. Ломоносова**119991 г. Москва, Ленинские горы, 1**E-mail: pavel.vdovin@gmail.com, kostmsu@gmail.com*

Поступила в редакцию 15.09.2015

В работе ставится задача организации периодически передаваемых сообщений для бортовых сетей, основанных на стандарте Avionics Full Duplex Ethernet (AFDX), и рассматривается алгоритм ее решения. Алгоритм основан на сочетании жадных стратегий и стратегий ограниченного перебора и состоит из нескольких процедур, которые поэтапно решают исходную задачу с учетом заданных ограничений. Также приводятся результаты экспериментальных исследований эффективности описанных процедур алгоритма.

1. ВВЕДЕНИЕ

В информационно управляющих системах реального времени (ИУС РВ) выполнение задач и обмен сообщениями должны происходить в строго определенные интервалы времени. Нарушение этих интервалов приводит к потере работоспособности ИУС РВ. В бортовых авиационных ИУС РВ кроме временных ограничений также накладываются ограничения на массогабаритные характеристики и предъявляются повышенные требования к надежности.

В настоящее время осуществляется переход от бортовых сетей обмена, построенных на основе каналов точка-точка и каналов множественного доступа с централизованным управлением [1, 2], к коммутируемым сетям на основе пакетной коммутации. В работе [3] был проведен сравнительный анализ подходов к построению коммутируемых сетей обмена сообщениями в бортовых ИУС РВ. В данной работе были рассмотрены сети, наиболее широко использующиеся в промышленности: Fibre Channel (FC), Avionics Full Duplex Ethernet (AFDX) и Software-Defined Networking (SDN). Сети сравнивались по следующим критериям: возможность обеспечения передачи сообщений в режиме реального времени,

возможность поддержки единого времени в системе, количество дополнительных физических ресурсов, необходимых для обеспечения требуемой надежности, возможность динамически (в ходе работы ИУС РВ) изменять маршруты передачи сообщений. По результатам обзора можно сделать вывод, что для исходно заданного набора периодических сообщений наиболее подходящими являются бортовые сети обмена, построенные на основе сетей AFDX. Сети AFDX в настоящее время широко применяются для построения бортовых сетей обмена в гражданской авиации.

В данной работе рассмотрены основные особенности сетей AFDX, приведена математическая формулировка задачи построения для исходно заданного набора периодических сообщений виртуальных каналов обмена и маршрутов для них в физической сети AFDX, предложен алгоритм решения сформулированной задачи и приведены результаты его экспериментального исследования.

2. СЕТИ AFDX

Стандарт Avionics Full Duplex Ethernet (AFDX) [4] описывает управление бортовыми сетями на основе традиционного стандарта Ethernet 802.3. Согласно стандарту AFDX, сеть состоит из следующих элементов (рис. 1):

- абоненты, передающие сообщения;

*Работа выполнена при финансовой поддержке РФФИ, грант № 16-07-01237.



Рис. 1. Типичная архитектура бортовой сети на основе стандарта AFDX.

- оконечные системы — системы, обеспечивающие интерфейс между абонентами и сетью;
- пакетные коммутаторы, соединяемые линиями передачи данных.

Стек протоколов, используемый при передаче сообщений в бортовых сетях на основе стандарта AFDX, основан на семействе протоколов TCP/IP. На канальном уровне используется протокол Ethernet, при этом на данном уровне выполняется маршрутизация кадров в сети с помощью механизма виртуальных каналов. На сетевом уровне используется протокол IP, однако маршрутизация пакетов на этом уровне не производится, так как данная функция перекладывается на канальный уровень. На транспортном уровне используется в основном протокол UDP (в некоторых случаях может быть использован протокол TCP).

Каждый абонент подключен к оконечной системе, причем к одной оконечной системе может быть подключено несколько абонентов. Соблюдение ограничений на время передачи сообщений в сети AFDX достигается за счет выделения гарантированной пропускной способности соединения между каждой парой оконечных систем. Такое соединение может проходить через несколько пакетных коммутаторов и линий передачи данных. В AFDX соединение между оконечными системами называют виртуальным каналом. Передача данных между абонентами осуществляется путем передачи сообщений по виртуальным каналам, маршруты которых в физической сети определены заранее. Для каждого виртуального канала определена только одна оконечная система-отправитель и одна или более оконечная

система-получатель. При этом по одному виртуальному каналу могут передаваться сообщения только от одного абонента.

Сообщение от абонента попадает на оконечную систему через специально выделенный порт. На оконечной системе-отправителе сообщение разбивается на порции данных (кадры), которым сопоставляются заголовки соответствующих уровней стека TCP/IP — а именно, транспортного (UDP), сетевого (IP), канального. Полученные в результате разбиения кадры выдаются на физический канал передачи данных. В каждый кадр, в поле MAC-адреса получателя записывается номер виртуального канала. Этот номер используется коммутаторами AFDX для маршрутизации кадра. Стандартная для Ethernet маршрутизация по MAC-адресу в AFDX не используется. После доставки кадра на оконечную систему-получатель (возможно, одну из нескольких) он буферизуется для последующей сборки сообщения. После прихода последнего кадра собранное сообщение направляется абонентам-получателям.

Надежность передачи данных в сетях AFDX обеспечивают за счет резервирования. Каждая оконечная система соединена с двумя независимыми идентичными сетями AFDX. Кадры передают в обе сети (в каждой сети кадр следует по идентичным маршрутам), и если в одной из них диагностируется ошибка передачи кадра (например, не совпадает контрольная сумма), то кадр берется из той сети, где не было ошибки. На оконечной системе-получателе происходит проверка целостности кадров, и если кадр уже был принят из другой сети, то кадр-дубликат сбрасывается.

Таблицы маршрутизации коммутатора AFDX настраиваются заранее. Коммутаторы AFDX

также выполняют контроль и фильтрацию трафика. При фильтрации трафика проверяется правильность следования кадров по виртуальным каналам, а также их целостность. Контроль трафика обеспечивает заданную для виртуального канала пропускную способность и не позволяет ее превышать. Для контроля пропускной способности канала в сетях AFDX применяется алгоритм текущего ведра с маркерами. Допустимую пропускную способность каждого виртуального канала устанавливают заранее при конфигурировании коммутатора. Таким образом, маршрутизация в сети AFDX задается статически, возможность динамически изменять таблицы маршрутизации стандартом не предусмотрена.

3. ПОСТАНОВКА ЗАДАЧИ ПОСТРОЕНИЯ ВИРТУАЛЬНЫХ КАНАЛОВ И МАРШРУТОВ ДЛЯ НИХ

Сеть AFDX будем описывать ориентированным графом $G = (P \cup K, E)$, где P — множество конечных систем, K — множество коммутаторов, E — множество дуг, соединяющих элементы из множества $P \cup K$. Для каждой дуги $e \in E$, $e = (k_1, k_2)$ существует парная дуга $e' \in E$, $e' = (k_2, k_1)$ ¹; для каждой дуги e задана пропускная способность R_e , причем $R_e = R_{e'}$.

Для каждой конечной системы $p \in P$ задано множество абонентов $A_p \subseteq A$ (A — множество всех абонентов), которые к ней подключены, причем каждый абонент подключен только к одной конечной системе. Каждая конечная система подключена только к одному каналу передачи данных².

Входными данными задачи являются потоки данных между абонентами, которые задаются в виде множества периодически передаваемых сообщений $MSG = \{msg\}$, для каждого из которых задаются следующие параметры:

- $size_{msg}$ (байт) — размер сообщения;
- T_{msg} (мс) — период передачи сообщения;
- J_{msg} (мкс) — максимальный джиттер порождения сообщения внутри периода (интервал времени, в котором может быть порождено сообщение от начала периода);
- $src_{msg} \in A$ — абонент-отправитель сообщения;
- $\{dst_{msg}\} \subset A$ — множество абонентов-получателей сообщения, абоненты-получатели должны быть подключены к конечным системам, отличным от конечной системы-отправителя: $dst_{msg} \notin A_p : src_{msg} \in A_p$.

Для каждого сообщения заданы следующие ограничения:

- сообщение должно передаваться не менее одного раза в период³;
- τ_{msg} (мс) — максимальная длительность передачи сообщения с момента выдачи сообщения от абонента конечной системе-отправителю до момента получения сообщения всеми абонентами-получателями;
- J_{msg}^* (мс) — максимальный джиттер передачи сообщения (разность между возможными максимальной и минимальной длительностью передачи сообщения).

Задача заключается в построении множества виртуальных каналов $VL = \{vl\}$ и построении для них маршрутов в физической сети обмена данным. По одному виртуальному каналу может передаваться несколько сообщений. Сообщения на конечной системе-отправителе разбиваются на кадры, и в сеть обмена конечная система передает кадры. Сообщение считается принятым, когда получен последний его кадр. Для каждого виртуального канала должны быть определены следующие параметры:

¹Парные дуги соответствуют каналу передачи данных в полнодуплексном режиме. Далее в статье под каналом передачи данных понимаются две парные дуги.

²Согласно стандарту, каждая конечная система подключена к двум идентичным сетям, и кадры передаются одновременно через обе сети, что позволяет улучшить надежность передачи; в данной работе не рассматриваются аспекты надежности передачи, поэтому исследуется передача данных только через одну сеть, результаты для другой сети аналогичны результатам для первой сети.

³Описанный в данной работе алгоритм рассчитан на случай, при котором сообщение порождается один раз в период (с учетом джиттера); если сообщение порождается чаще одного раза в период, то передача всех его кадров не гарантируется, при этом кадры могут сбрасываться на коммутаторах.

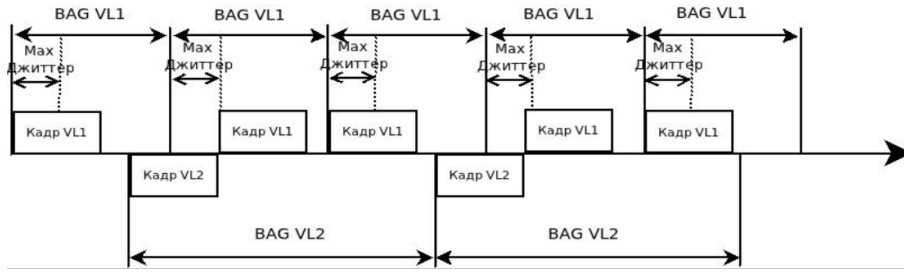


Рис. 2. Мультиплексирование виртуальных каналов с помощью джиттера.

- LM_{vl} (байт) — максимальный размер кадра, передаваемый по данному виртуальному каналу; согласно стандарту, размер кадра лежит в следующих пределах: $64 \leq LM_{vl} \leq 1518$ байт, размер заголовка кадра фиксирован и равен $c = 47$ байт, временной промежуток между кадрами равно $gap = 12$ мкс;
- BAG_{vl} (мс) — минимальный промежуток времени между передачей кадров при нулевом джиттере порождения кадров; согласно стандарту, данное значение лежит в промежутке от 1 до 128 мс и является степенью двойки;
- JM_{vl} — максимальный джиттер порождения кадров на оконечной системе-отправителе (см. рис. 2); данное значение используется при мультиплексировании виртуальных каналов и пояснено далее;
- $S_{vl} \in P$ — оконечная система-отправитель кадров данного виртуального канала;
- $D_{vl} = \{d_{vl}\} \subseteq P$ — множество оконечных систем-получателей кадров данного виртуального канала ($S_{vl} \notin D_{vl}$);
- $Tree_{vl} \subseteq E$ — маршруты передачи кадров в сети, представляет собой дерево, в котором корнем является S_{vl} , а листьями являются все элементы множества D_{vl} ;
- Множество сообщений $MSG_{vl} = \{msg\}$, передаваемых по данному виртуальному каналу; сообщения должны исходить из одного абонента, подключенного к S_{vl} , абоненты-получатели должны быть подключены к оконечным системам из множества D_{vl} .

На рис. 2 показано, что кадры виртуальных каналов $VL1$ и $VL2$ не могут быть переданы строго регулярно (с интервалом между кадрами, равным BAG). При сдвиге кадров относительно BAG на величину, не превышающую JM (Max Джиттер), кадры могут быть переданы, не нарушая регулярности передачи.

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

- Суммарная пропускная способность, зарезервированная под виртуальные каналы, проходящие через физический канал передачи данных e , не превосходит его пропускной способности:

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

здесь R_e — пропускная способность физического канала передачи данных e .

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

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

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 $\lceil size_{msg} / (LM_{vl} - c) \rceil$, получаем ограничение (2).

- Максимальный джиттер кадров на оконечных системах-отправителях не должен превосходить⁴ 500 мкс:

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

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

$$\forall msg \in MSG : Dur(msg) \leq \tau_{msg}; Jit(msg) \leq J_{msg}^* \quad (4)$$

здесь $Dur(msg)$ и $Jit(msg)$ — процедуры вычисления длительности передачи сообщений и их джиттеров соответственно, данные процедуры описаны далее в работе.

4. БЛИЗКИЕ ЗАДАЧИ И МЕТОДЫ ИХ РЕШЕНИЯ

В литературе среди работ, посвященных проектированию сетей AFDX, основное внимание уделяется задаче нахождения оценки длительности передачи кадров. В работе [5] представлен обзор методов, используемых в настоящее время. Наиболее эффективными считаются методы Network Calculus [6-8] и Trajectory Approach [9,10]. Согласно проведенным исследованиям, описанным в [5], Trajectory Approach позволяет получить более точные оценки, чем Network Calculus.

Алгоритмы, посвященные непосредственно проектированию сетей AFDX, описаны в работах [8, 11]. В работе [8] используется возможность задавать статический приоритет виртуальных каналов на портах коммутаторов для оптимизации времени передачи кадров⁵.

⁴ Данное требование исходит из рекомендаций, описанных в стандарте [4]. Согласно стандарту, вычисленный максимальный джиттер может превышать 500 мкс, но при этом его значение должно быть установлено в 500 мкс. В таком случае реальная пропускная способность виртуальных каналов может не соответствовать заявленной, поэтому в данной работе не превышение вычисленного джиттера значения в 500 мкс ставится как ограничение.

⁵ Согласно стандарту, на выходном порту коммутатора можно статически определить приоритет виртуального канала. Стандарт позволяет задать две очереди высокого и низкого приоритета, и кадры очереди низкого приоритета не могут начать передачу, пока не выданы кадры из очереди высокого приоритета.

При решении задачи оптимизации используется метод Network Calculus.

Метод построения виртуальных каналов для приведенной задачи описан в статье [11]. В работе приведен оптимальный алгоритм проектирования виртуального канала для передачи по нему не более одного сообщения с заданными ограничениями на время выдачи последнего кадра в сеть (то есть время фактической передачи кадра по сети не учитывается). В работе также предложена процедура агрегации и алгоритм построения маршрутов виртуальных каналов на основе решения задачи Штейнера [12] с помощью смешанного целочисленного программирования (Mixed Integer Linear Programming (MILP)). Описанный в работе [11] подход обладает следующими недостатками:

1. При решении задачи построения маршрутов ставится задача целочисленного линейного программирования сразу для всех виртуальных каналов, и при неуспешном решении не строится ни одного маршрута.
2. Описанные этапы (проектирование виртуального канала, агрегация и маршрутизация) решаются последовательно и независимо, и при неуспешном построении маршрутов не производится попытка, например, по-другому агрегировать виртуальные каналы.
3. В итоговом построении не учитываются ограничения на длительность и джиттер передачи сообщений. Описанные недостатки показывают, что поставленная задача полностью не решается в указанной работе. Отсутствие подходов к решению задачи в других работах показывает, что требуется построение алгоритма ее решения.

5. ОБЩАЯ ХАРАКТЕРИСТИКА АЛГОРИТМА

Алгоритм организации передачи сообщений основан на жадных стратегиях и стратегиях ограниченного перебора. Ограниченный перебор используется в случае неуспешного завершения жадных процедур и заключается в выполнении перебора с ограниченной глубиной. Также в алгоритме используется процедура агрегации, целью которой является построение виртуальных

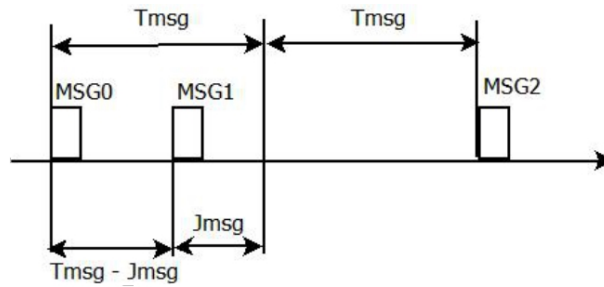


Рис. 3. Возникновение джиттера сообщения.

каналов, через которые передается более одного сообщения. Данная процедура выполняется в случае неуспешного назначения виртуальных каналов с меньшим числом сообщений.

Общая схема алгоритма следующая:

Шаг 1. Для каждого потока сообщений выделить виртуальный канал и произвести настройку параметров LM , BAG и JM с учетом ограничения (2). На данном шаге используется метод, описанный в разделе 6.1.

Шаг 2. Для каждого виртуального канала проверить выполнение ограничения (3). Если ограничение выполняется не для всех виртуальных каналов, произвести процедуру агрегации, описанную в разделе 6.2. Входом процедуры являются виртуальные каналы, для которых не выполняется ограничение. Процедура объединяет сообщения, исходящие от одного абонента, в один виртуальный канал с настройкой новых параметров LM , BAG и JM таким образом, чтобы выполнялись ограничения (2) и (3) (благодаря уменьшению количества виртуальных каналов на окончательной системе значение максимального джиттера на них может уменьшиться). Прежние виртуальные каналы этих сообщений удаляются.

Шаг 3. Для каждого виртуального канала в порядке убывания требуемой пропускной способности⁶ произвести построение маршрута с учетом ограничения (1):

Шаг 3.1. Выполнить процедуру построения маршрута, описанную в разделе 6.3. Если маршрут найден, перейти к шагу 4.

Шаг 3.2. Выполнить процедуру ограниченно-

⁶В данном случае можно использовать другие критерии для определения порядка перебора виртуальных каналов. Например, производить поиск маршрута в порядке убывания критерия $d(vl) = |MSG_{vl}| \cdot bw(vl)$ — пропорционально количеству сообщений и пропускной способности.

го перебора виртуальных каналов, описанную в разделе 6.4 и заключающуюся в проведении попыток поиска другого маршрута для уже назначенных виртуальных каналов таким образом, чтобы можно было построить маршрут и для текущего виртуального канала.

- Если маршрут построен успешно, то перейти к следующему виртуальному каналу.
- Если маршрут построить не удалось, то в случае, если в виртуальном канале передается только одно сообщение, его назначение считается невозможным. Иначе из виртуального канала убирается сообщение с наибольшим значением LM/BAG и его назначение считается неуспешным. Виртуальный канал переконфигурируется с помощью процедуры агрегации, после чего он уже может требовать меньше пропускной способности, чем исходный виртуальный канал. Для данного виртуального канала производится переход на шаг 3.1.

Шаг 4. Для каждого сообщения выполнить шаги 4.1–4.3.

Шаг 4.1. Произвести вычисление длительности и джиттера передачи сообщений (см. раздел 6.5) и проверить выполнение ограничения (4). Если ограничения выполняются, перейти к следующему сообщению.

Шаг 4.2. Произвести процедуру переконфигурации виртуального канала, по которому передается сообщение (см. раздел 6.6), заключающуюся в попытке перенастроить параметры LM , BAG и JM с учетом оценок длительности передачи, полученных на шаге 4.1. Если после проведения процедуры ограничение (4) выполняется, перейти к следующему сообщению.

Шаг 4.3. Произвести процедуру агрегации виртуальных каналов, проводя для каждого агрегированного виртуального канала построение маршрута и проверку ограничений (1)–(4). Агрегация позволяет уменьшить длительность передачи кадров, так как уменьшает количество передаваемых кадров в сети, в результате чего уменьшается время ожидания кадров в очередях на выходных портах коммутаторов. Если построение маршрута успешно и все ограничения выполняются, то прежние виртуальные каналы заменяются на новый. Если после выполнения процедуры для исходного сообщения не выполняется ограничение (4), назначение сообщения считается неуспешным и сообщение удаляется из виртуального канала.

6. ОПИСАНИЕ ПРОЦЕДУР, ИСПОЛЬЗУЕМЫХ В АЛГОРИТМЕ

6.1. Настройка параметров виртуальных каналов

Настройка параметров виртуальных каналов основана на методе, предложенном в работе [11]. В работе предлагается задавать параметры LM и BAG с помощью решения задачи минимизации пропускной способности виртуального канала. Для этого пропускная способность представляется как функция $bw_{vl}(n)$, зависящая от количества кадров n , на которое делится сообщение msg . В работе [11] задача ставится с ограничением на время передачи сообщения, которое задается в виде:

$$(n - 1)BAG \leq \delta, \quad (5)$$

здесь δ — ограничение на время выдачи последнего кадра в канал.

Так как время передачи сообщения — это время ожидания выдачи его последнего кадра δ и длительность передачи последнего кадра через сеть Δ , то верно: $\delta = \tau_{msg} - \Delta$. Величина Δ может быть вычислена только после построения маршрутов виртуальных каналов, поэтому на данном этапе ее можно задать некоторым числом (например, $1 \text{ } \mu\text{s}$), а проверку выполнения ограничений (4) проверять после построения маршрутов.

Кроме ограничений на длительность передачи сообщения, должны быть также выполнены

ограничения (2) и ограничения на максимальный джиттер J_{msg} порождения сообщения⁷. Для виртуального канала с одним сообщением ограничение (2) принимает вид:

$$n \cdot BAG \leq T_{msg} \quad (6)$$

Для формирования ограничения на джиттер порождения сообщения рассмотрим наихудшую ситуацию, при которой некоторое сообщение MSG_1 было выдано через $T_{msg} - J_{msg}$ после предыдущего сообщения MSG_0 (см. рис. 3).

Если $n \cdot BAG > T_{msg} - J_{msg}$, то к моменту выдачи MSG_1 не все кадры сообщения MSG_0 поступили в канал. В таком случае выдача последнего кадра сообщения MSG_1 превосходит величину $(n - 1) \cdot BAG$ на величину, равную длительности передачи оставшихся кадров сообщения MSG_0 , которую можно оценить как $n \cdot BAG - (T_{msg} - J_{msg})$. Исходя из описанных рассуждений, ограничение (5) принимает следующий вид:

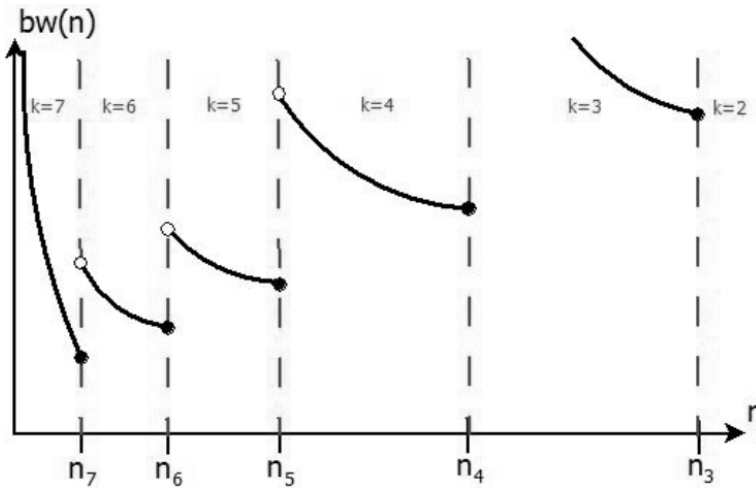
$$\begin{cases} (n - 1)BAG \leq \delta, \\ \text{при } n \cdot BAG \leq T_{msg} - J_{msg} \\ (2n - 1) \cdot BAG - (T_{msg} - J_{msg}) \leq \delta, \\ \text{иначе.} \end{cases} \quad (7)$$

Таким образом, LM и BAG формируются с помощью минимизации величины $bw(n)$, вычисляемой как $bw(n) = \frac{LM}{BAG}$, с учетом ограничений (6) и (7). Величины LM и BAG в данной формуле зависят от n и задаются следующим образом:

- $LM(n) = \max(64, \lceil \frac{size_{msg}}{n} \rceil + c)$, так как размер кадра не меньше 64 байт при размере заголовка c ,
- $BAG(n) = 2^{k(n)}$, $k(n) = 0 \dots 7$, где $k(n)$ определяет значение степени двойки BAG .

Согласно исследованиям, проведенным в работе [11], значения функции $bw(n)$ в зависимости от n ведут себя согласно графику, изображенному на рис. 4.

⁷Указанные ограничения не учитываются в работе [11], поэтому описанные выкладки имеют отличия от оригинального подхода.

Рис. 4. График зависимости функции $bw(n)$ от n .

Значения n_i , изображенные на рисунке, вычисляются следующим образом:

$$\begin{cases} n_i = \min(\lfloor \frac{\delta}{2^i} \rfloor + 1, \lfloor \frac{T_{msg}}{2^i} \rfloor), \\ \text{при } n_i \cdot 2^i \leq T_{msg} - J_{msg} \\ n_i = \min(\lfloor \frac{\delta + T_{msg} - J_{msg}}{2^{i+1}} + 1/2 \rfloor, \lfloor \frac{T_{msg}}{2^i} \rfloor), \\ \text{иначе.} \end{cases}$$

n_i соответствует количеству кадров, при которых $k(n)$ принимает граничное значение, то есть при $n > n_i$ для $k = i$ не выполняется одно из ограничений (6), (7), причем при $n = n_i$ оба ограничения выполняются. С другой стороны, при фиксированном k график является убывающим.

Согласно данному графику, для выбора оптимального значения n достаточно взять наименьшее значение n_i , при которых выполняются ограничения (6) и (7), и вычислить соответствующим образом $LM(n)$ и $BAG(n)$. Полученные значения являются оптимальными с точки зрения минимизации пропускной способности виртуального канала.

После вычисления параметров LM_{vl} и BAG_{vl} для всех виртуальных каналов возможно вычисление максимальных джиттеров порождения кадров JM_{vl} . Данная величина равна времени ожидания отправки кадров со всех виртуальных каналов, исходящих из той же оконечной системы-отправителя es :

$$JM_{vl} = \sum_{v \in VL_{es}, v \neq vl} (\frac{LM_v}{R_e} + gap), \quad (8)$$

где VL_{es} — множество виртуальных каналов, исходящих из es ; R_e — пропускная способность физического канала передачи данных, исходящего из es ; gap — временной промежуток между кадрами.

6.2. Процедура агрегации

Процедура агрегации заключается в проектировании виртуального канала, по которому передаются несколько сообщений, исходящих от одного абонента. Процедура вызывается в следующих случаях:

1. При невыполнении ограничения (3) для некоторого виртуального канала.
2. При невыполнении ограничения (4) для некоторого сообщения и неуспешной процедуре переконфигурации.

6.2.1 Общая схема процедуры агрегации

Пусть процедура проводится для виртуального канала vl , исходящего из оконечной системы es . Процедура агрегации выполняется по следующей схеме:

Шаг 1. Для каждого абонента src , подключенного к es , сформировать множество виртуальных каналов VL_{src} , состоящее из виртуальных каналов с сообщениями, исходящими из src .

Шаг 2. Если каждое из множеств VL_{src} содержит не более одного элемента или для всех пар виртуальных каналов попытка агрегации уже проведена, вернуть неуспех.

Шаг 3. Выбрать нерассмотренную пару виртуальных каналов vl_1 и vl_2 среди множеств VL_{src} , таких что $vl_1, vl_2 \in VL_{src}$ и значение $r(vl_1) \cdot r(vl_2)$ максимально среди всех подобных пар. $r(vl)$ — критерий, который предлагается задавать одним из следующих способов:

- $r(vl) = bw(vl)/|MSG_{vl}|$, $bw_{vl} = LM_{vl}/BAG_{vl}$ — критерий прямо пропорционален требуемой пропускной способности и обратно пропорционален количеству сообщений в виртуальном канале; данный критерий исходит из того принципа, что следует объединять сообщения из виртуальных каналов с наибольшей резервируемой пропускной способностью, чтобы попытаться получить наибольший выигрыш в итоговой экономии пропускной способности; кроме того, слишком большое количество сообщений значительно увеличивает длительность их передачи (т.к. сообщения могут долго ждать отправки кадров других сообщений), поэтому предлагается обратная пропорциональность количеству сообщений;
- $r(vl) = (\alpha \cdot bw(vl) + \beta \cdot \min_{msg \in MSG_{vl}} \tau_{msg}) / |MSG_{vl}|$, $\alpha + \beta = 1$ — кроме резервируемой пропускной способности в данном критерии добавляется требование длительности доставки сообщения; так как агрегация увеличивает длительность доставки сообщения, целесообразно выбирать сообщения с большим значением требуемой длительности передачи.

Шаг 4. Провести агрегацию элементов vl_1 и vl_2 согласно алгоритму, описанному в разделе 6.2.2.

Шаг 5. Если требуемая пропускная способность агрегированного виртуального канала больше суммарной требуемой пропускной способности каналов vl_1 и vl_2 , восстановить исходные виртуальные каналы⁸ и перейти на шаг 2.

Шаг 6. При невыполнении требования (3) для агрегированного виртуального канала, восстано-

⁸Требуемая пропускная способность агрегированного виртуального канала может быть как больше, так и меньше суммы пропускных способностей исходных виртуальных каналов. Алгоритм основан на жадном предположении, что при уменьшении пропускной способности достигнут положительный результат.

вить исходные виртуальные каналы и перейти на шаг 2.

Шаг 7. Если хотя бы для одного виртуального канала vl_1 или vl_2 уже построен маршрут, провести попытку построения маршрута агрегированного виртуального канала. При неуспешном построении восстановить исходные виртуальные каналы и перейти на шаг 2.

Шаг 8. Убрать vl_1 и vl_2 из VL_{src} и вставить в VL_{src} полученный виртуальный канал. Если нарушаемые свойства для vl все еще не выполняются, перейти на шаг 2. Иначе завершить процедуру.

6.2.2 Агрегация выбранного множества сообщений

Агрегация двух виртуальных каналов, выполняемая на шаге 4 алгоритма из раздела 6.2.1, в целом схожа с методом настройки параметров виртуального канала, описанного в разделе 6.1. Отличие заключается в том, что в одном виртуальном канале формируются кадры нескольких сообщений. Пусть дано множество потоков сообщений $MSG_{src} = \{msg\}$, исходящих из одного абонента src , для которых требуется построить виртуальный канал. Максимальное количество кадров в очереди виртуального канала равно суммарному количеству кадров:

$$N = \sum_{msg \in MSG_{src}} \lceil size_{msg} / (LM_{msg} - c) \rceil.$$

Как и в случае с одним сообщением, ставится задача минимизации пропускной способности, которая задается функцией от N :

$$bw(N) = LM(N) / 2^{k(N)}.$$

При этом должны быть выполнены ограничения, аналогичные (6) и (7). Отличием является то, что из ограничений T_{msg} , δ_{msg} выбирается наименьшее, так как эти ограничения должны выполняться для всех сообщений виртуального канала:

$$N \cdot BAG \leq \min_{msg} T_{msg}, \quad (9)$$

$$\left\{ \begin{array}{l} (N - 1)BAG \leq \min_{msg} \delta_{msg}, \\ \text{при } N \cdot BAG \leq \min_{msg} (T_{msg} - J_{msg}) \\ (2N - 1) \cdot BAG \leq \min_{msg} (\delta_{msg} + T_{msg} - J_{msg}), \\ \text{иначе.} \end{array} \right. \quad (10)$$

Оптимальное значение N определяется аналогично случаю с одним сообщением, путем выбора наименьшего из значений N_i , для которых верны ограничения (9) и (10):

$$\begin{cases} N_i = \min\left(\left\lfloor \frac{\min_{msg} \delta_{msg}}{2^i} \right\rfloor + 1, \left\lfloor \frac{\min_{msg} T_{msg}}{2^i} \right\rfloor\right), \\ \text{при } N_i \cdot 2^i \leq \min_{msg}(T_{msg} - J_{msg}) \\ N_i = \min\left(\left\lfloor \frac{\min_{msg}(\delta + T_{msg} - J_{msg})}{2^{i+1}} \right\rfloor + 1/2\right), \\ \left\lfloor \frac{\min_{msg} T_{msg}}{2^i} \right\rfloor, \text{ иначе.} \end{cases}$$

По выбранному значению N_i определяется значение $k(N_i) = i$. Основным вопросом является вычисление функции $LM(N)$, которая представляет собой максимальный размер кадра, при котором суммарное количество кадров равно N .

Следует отметить, что точного значения данной функции может и не иметь. Например, если имеется два сообщения одинакового размера и $N = 3$, то невозможно определить максимальное значение кадра, так как оба сообщения разобьются на одинаковое количество кадров и, соответственно, количество кадров чётно.

Предлагаемым решением является такое значение $LM(N)$, при котором сообщения разделяются на максимальное количество кадров, не превышающее N . Так как конфигурация позволяет передавать N кадров с промежутком BAG между кадрами, то за то же время конфигурация гарантирует передачу любого количества кадров, не превышающего N (при фиксированных BAG и LM). При таком вычислении $LM(N)$ может резервироваться больше пропускной способности, чем необходимо виртуальному каналу, однако значение зарезервированной пропускной способности удовлетворяет заданным ограничениям.

Значение $LM(N)$ предлагается вычислять итеративно, начиная со значения $LM(N_0) = \max_{msg}(size_{msg} + c)$, $N_0 = |MSG_{src}|$. Данное значение определяет размер кадра при условии, что количество кадров равно количеству сообщений. В этом случае каждое сообщение должно попасть ровно в один кадр (ограничения на размер кадра в данном слу-

чае не учитываются) и размер кадра равен максимальному размеру сообщения.

На каждом последующем шаге производится увеличение общего количества кадров на 1 при условии, что каждое сообщение разбито на кадры одинакового размера. В реальности сообщения не всегда разбиваются на равные части, и общее количество кадров может быть меньше получаемого, поэтому данное условие является жадным. Для разбиения выбирается сообщение, у которого размер кадра на предыдущей итерации максимален.

Описанный алгоритм вычисления $LM(N)$ заключается в выполнении следующих шагов:

Шаг 1. $F_{msg}^{N_0} = 1, \forall msg \in MSG_{src}$ — данная величина показывает, на сколько кадров может быть разбито данное сообщение msg на данной итерации.

Шаг 2. Для $k = N_0 + 1 \dots N$ выполнить шаги 2.1, 2.2:

Шаг 2.1. $l = \arg \max_{msg}(c + size_{msg}/F_{msg}^{k-1})$ — данной величине присваивается сообщение, которое на предыдущем шаге разбивалось на кадр максимального размера; если таких сообщений несколько, то берется любое;

Шаг 2.2. $F_{msg}^k = F_{msg}^{k-1}, \forall msg \in MSG_{src}/\{l\}$, $F_l^k = F_l^{k-1} + 1$ — количество кадров увеличивается только для выбранного сообщения l ;

Шаг 3. $LM(N) = \max_{msg}(c + size_{msg}/F_{msg}^N)$ — функция возвращает максимальный размер кадра с учетом получившихся разбиения сообщений на кадры.

Следует отметить, что во многих случаях выполнено: $f(k) = f(k-1)$. Например, если имеется два сообщения одинакового размера $size$, то верны следующие выкладки:

- $f(2) = (size + c)$;
- $f(3) = (size + c)$ — одно сообщение разбито на два одинаковых кадра, второе имеет исходный размер;
- $f(4) = (size/2 + c)$ — оба сообщения разбиты на два одинаковых кадра.

Значение LM для агрегированного виртуального канала определяется, таким образом, путем вычисления функции $LM(N_i)$.

Вычисление джиттера для агрегированного виртуального канала производится согласно формуле (8).

6.3. Построение маршрутов виртуальных каналов

Маршрут виртуального канала представляет собой дерево, являющееся подграфом исходного графа сети G , корнем дерева является оконечная система-отправитель, листьями — оконечные системы-получатели кадров виртуального канала. Построение маршрута происходит с учетом ограничений на пропускную способность каналов передачи данных. Алгоритмы для построения маршрутов виртуальных каналов со схожей постановкой также можно найти в работах [13, 14], посвященных задаче назначения запросов в центрах обработки данных.

Для построения маршрута используется алгоритм поиска дерева групповой передачи, основанный на алгоритме, описанном в работе [15]. Схема построения маршрута виртуального канала vl заключается в выполнении следующих шагов:

Шаг 1. Преобразовать граф G , убрав из него все дуги $e \in E$, для которых остаточная пропускная способность, вычисляемая как разность $R_e - \sum_{R_v \in VL: e \in Tree_v} R_v$, меньше, чем требуемая пропускная способность $bw(vl)$. Полученный граф G' состоит только из тех дуг, через которые может быть проложен маршрут для данного виртуального канала.

Шаг 2. Если в графе G' между оконечной системой-отправителем $S_{vl} \in N$ и хотя бы одной из оконечных систем-получателей $d_{vl} \in D_{vl} \subset P$ не существует пути, то алгоритм возвращает неуспех.

Шаг 3. Запустить алгоритм Дейкстры нахождения кратчайшего пути в графе G' от оконечной системы-отправителя S_{vl} согласно заданному критерию $l(n)$, $n \in (P \cup K)$. Остановить алгоритм после нахождения кратчайшего пути $path(d_{vl})$ до одной из оконечных систем-получателей из множества D_{vl} , до которых еще не найден путь. Предлагается использовать один из следующих критериев:

- $l(n) = |path(S_{vl}, n)|$ — поиск пути минимальной длины; каждой дуге в процессе рабо-

ты алгоритма Дейкстры присваивается вес, равный 1.

- $l(n) = \sum_{e \in path(S_{vl}, n)} \frac{\varepsilon + \sum_{v \in VL: e \in Tree_v} bw(v)}{R_e}$ — поиск пути “максимальной ширины”.

Второй критерий предложен в работе [15] и заключается в том, что каждой дуге сопоставляется вес, равный занятой ширине канала и выражающейся в виде отношения зарезервированной пропускной способности к максимальной пропускной способности канала. Величина $\varepsilon > 0$ задается для сравнения весов дуг в случае, если через дугу не проходит ни один виртуальный канал. Данный критерий является более предпочтительным, так как учитывает загруженность каналов передачи данных.

Шаг 4. Если найдены пути до всех получателей, то вернуть множество маршрутов $path(d_{vl})$ до всех оконечных систем-получателей. Объединение построенных путей $Tree_{vl} = \bigcup_{d_{vl} \in D_{vl}} path(d_{vl})$ составляет маршрут виртуального канала. Иначе запомнить пути до найденных получателей, присвоить всем дугам данных путей вес, равный 0 (для любого из критериев), и перейти на шаг 3. Дуги, которым присвоен вес 0, соответствуют элементам дерева маршрута виртуального канала, для которых уже построен путь, поэтому на последующих этапах построения маршрута учитывать веса данных дуг не нужно.

6.4. Процедура ограниченного перебора виртуальных каналов

Процедура ограниченного перебора виртуальных каналов заключается в попытке повторного построения маршрутов уже назначенных виртуальных каналов таким образом, чтобы данный канал vl мог быть назначен. Процедура ограничивается максимальным размером множества рассматриваемых виртуальных каналов для повторного поиска маршрутов. Алгоритм заключается в выполнении следующих шагов:

Шаг 1. Для каждого множества назначенных виртуальных каналов, мощность которого не больше заданного значения, выполнить шаги 1.1–1.7:

Шаг 1.1. Снять все маршруты виртуальных каналов из выбранного множества.

Шаг 1.2. Выполнить алгоритм построения маршрута для данного канала vl с помощью алгоритма, описанного в разделе 6.3.

Шаг 1.3. Если маршрут виртуальный канал не может быть построен, то перейти на шаг 1.7.

Шаг 1.4. Выполнить алгоритм построения маршрута для всех снятых виртуальных каналов.

Шаг 1.5. Если хотя бы один из виртуальных каналов не может быть назначен, снять виртуальный канал vl , перейти на шаг 1.7.

Шаг 1.6. Выполнить назначения виртуальных каналов согласно найденным маршрутам и вернуть успех.

Шаг 1.7. Восстановить изначальные назначения маршрутов виртуальных каналов.

Шаг 2. Если после ограниченного перебора маршрут виртуального канала не может быть построен, вернуть неуспех.

6.5. Вычисление максимальной длительности и джиттера передачи сообщений

Максимальная длительность передачи сообщения msg , передаваемого по виртуальному каналу vl , вычисляется следующим образом:

$$Dur(msg) = \mu + \delta + \Delta, \text{ где}$$

- μ — константа, описывающее время, требуемое на разбиение сообщения на кадры и сборку сообщения из кадров (и, возможно, прохождение сообщения через некоторые обязательные этапы обработки, занимающие не более чем некоторое известное постоянное время);
- δ — максимальное время, вычисляемое с момента постановки кадров сообщения в очередь виртуального канала до момента выдачи последнего кадра из этой очереди;
- Δ — максимальная длительность передачи последнего кадра.

Величина δ , как было показано в разделах 6.1 и 6.2, вычисляется как $\delta = BAG_{vl} \sum_{msg \in MSG_{vl}} [size_{msg} / (LM_{vl} - c)]$.

Величина Δ может быть вычислена с помощью одного из известных методов оценки длительности передачи кадров [5–10]. Данная величина может быть вычислена после построения маршрутов виртуальных каналов.

Максимальный джиттер передачи сообщения вычисляется как разница между максимальной и минимальной длительностью передачи сообщения:

$$Jit(msg) = Dur(msg) - Dur_{\min}(msg).$$

Минимальная длительность передачи сообщения вычисляется следующим образом: $Dur_{\min}(msg) = \mu + \delta_{\min} + \Delta_{\min}$, где

- δ_{\min} — минимальное время, вычисляемое с момента постановки кадров сообщения в очередь виртуального канала до момента выдачи последнего кадра из этой очереди;
- Δ_{\min} — минимальная длительность передачи последнего кадра.

Величина δ_{\min} вычисляется как время ожидания сообщения в случае, если на момент поступления кадров сообщения в очередь виртуального канала данная очередь пуста. Данная величина, таким образом, вычисляется как $\delta_{\min} = BAG_{vl} |size_{msg} / (LM_{vl} - c)|$.

Величина Δ_{\min} вычисляется как длительность передачи кадров в случае, когда при прохождении через коммутаторы кадр поступает в пустую очередь выходного порта коммутатора и сразу передается дальше. Данная величина, таким образом, вычисляется следующим образом:

$$\Delta_{\min} = \max_{path \in Tree_{vl}} \left(\sum_{e \in path} \frac{LM_{vl}}{R_e} + \sum_{k \in path} t_k \right).$$

Длительность передачи в данном случае вычисляется как время передачи по каналам, равное $\frac{LM_{vl}}{R_e}$ для каждого канала e , и время задержки кадра в коммутаторах, равное некоторой фиксированной величине t_k , соответствующей длительности обработки кадра в коммутаторе k .

6.6. Процедура переконфигурации виртуального канала

Процедура переконфигурации выполняется в случае, когда не выполняется ограничение (4) для некоторого виртуального канала vl . Процедура заключается в попытке заново сконфигурировать параметры LM_{vl} и BAG_{vl} на основе вычисленной длительности передачи кадра. В разделах 6.1 и 6.2 задавалась некоторая начальная

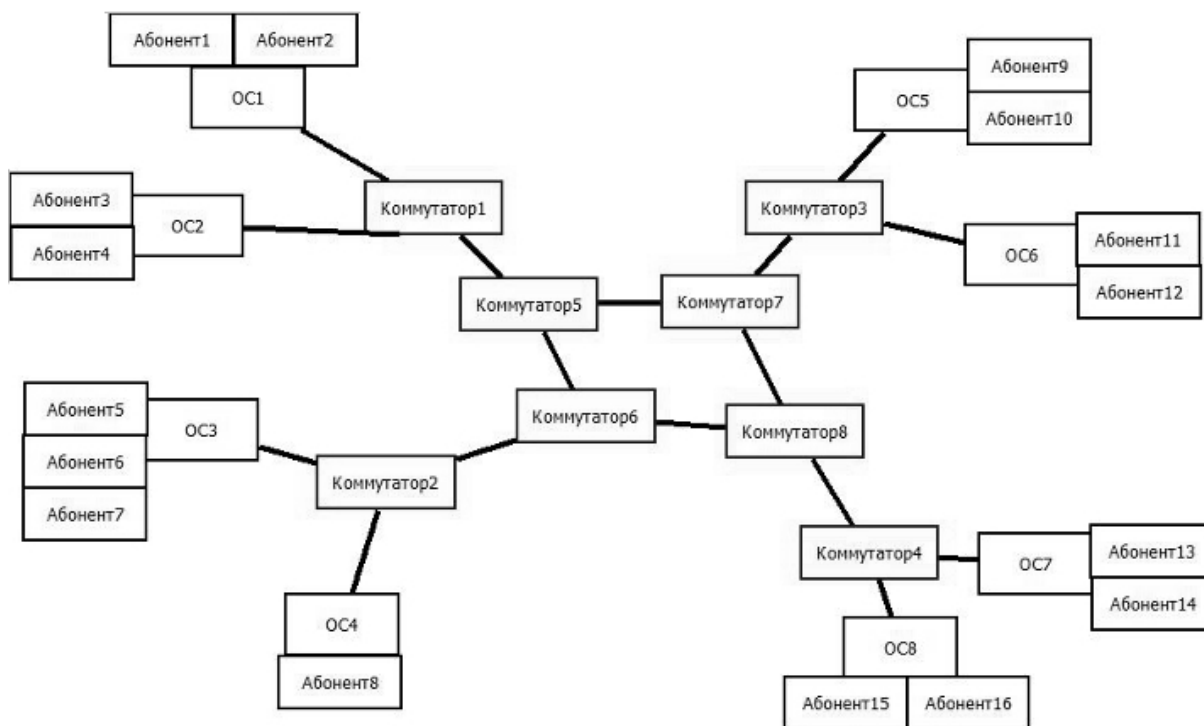


Рис. 5. Сеть AFDX, используемая при проведении экспериментальных исследований.

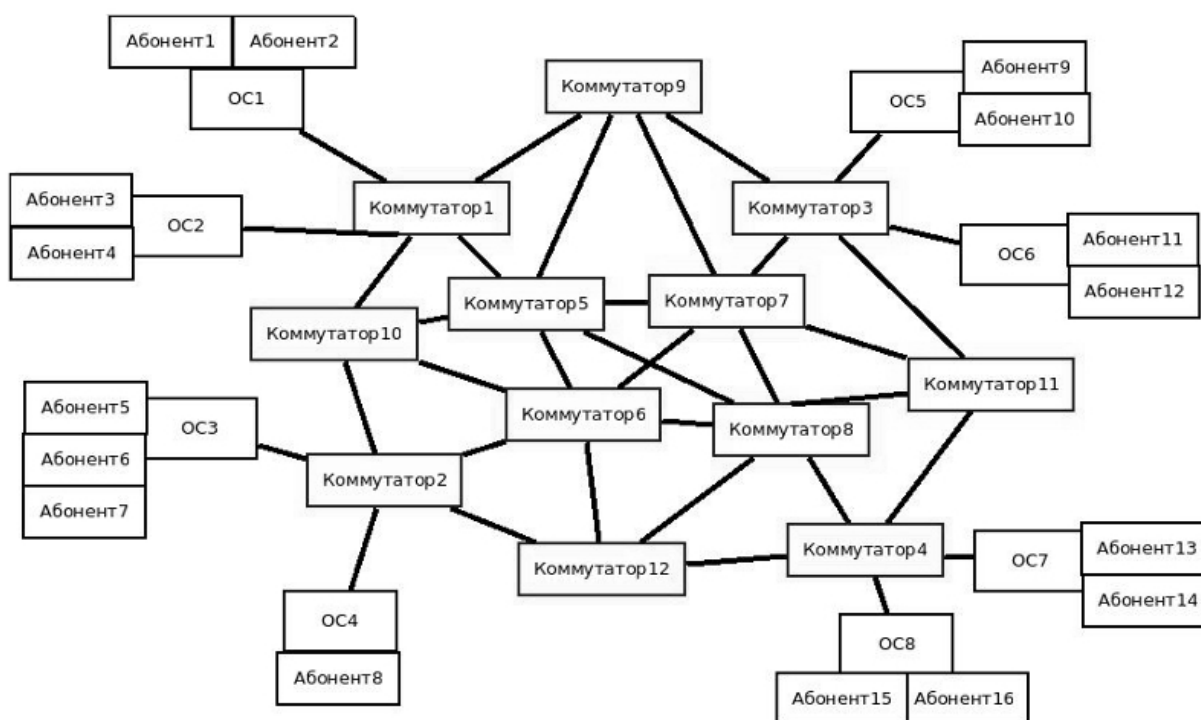


Рис. 6. Сеть AFDX с избыточными каналами передачи данных.

оценка величины Δ . Пусть данная оценка равна Δ_0 (например, $\Delta_0 = 1mc$). После построения маршрутов виртуальных каналов данная величина уже может быть оценена точнее, как указа-

но в разделе 6.5. С помощью новой оценки можно заново сконфигурировать параметры виртуального канала с помощью метода, указанного в разделах 6.1 или 6.2.2 (в зависимости от количе-

ства сообщений, передаваемых через виртуальный канал).

Общая схема процедуры переконфигурации следующая:

Шаг 1. Вычислить необходимую максимальную длительность кадра Δ :

- В случае нарушения длительности передачи сообщения данная величина вычисляется как $\Delta_0 + (Dur(msg) - \tau_{msg})$, то есть требуемая длительность передачи увеличивается на величину, на которую она нарушается;
- В случае нарушения максимального джиттера передачи сообщения данная величина вычисляется как $\Delta_0 + (Jit(msg) - J_{msg}^*)$, то есть требуемая длительность передачи увеличивается на величину, на которую нарушается максимальный джиттер;

Шаг 2. Параметры виртуального канала пересчитываются с помощью метода, описанного в разделе 6.1 или 6.2.2 с новой оценкой длительности передачи кадра. Если метод не может получить конфигурацию, удовлетворяющую новым требованиям, вернуть неуспех и отказать в назначении данного сообщения⁹.

Шаг 3. Проверить, выполняются ли ограничение (3) для виртуального канала на оконечной системе-отправителе. Если данные требования не выполняются, отказать в назначении данного сообщения.

Шаг 4. Проверить, выполняются ли ограничение (1) на маршруте передачи виртуального канала. Если ограничение выполняется, перейти на шаг 5, иначе выполнить шаги 4.1–4.2:

Шаг 4.1. Отменить маршрут виртуального канала и выполнить построение маршрута согласно алгоритму из раздела 6.3.

Шаг 4.2. Если процедура построения маршрута возвращает неуспех, отказать в назначении данного сообщения.

Шаг 5. Пересчитать максимальные длительности и джиттеры сообщений, для которых данные требования уже были проверены и выполнялись.

⁹В случае, когда в виртуальном канале передаются кадры нескольких сообщений, неуспех возвращается для сообщения, у которого нарушалось ограничение (4). Для остальных сообщений необходимо проверить выполнимость требований и провести процедуру переконфигурации в случае их невыполнимости.

Если хотя бы для одного из них требование нарушается, отказать в назначении данного сообщения.

Шаг 6. Пересчитать максимальную длительность и джиттер данного сообщения. Если требования снова не выполняются, присвоить $\Delta_0 = \Delta$ и перейти на шаг 1. Если данный шаг в процессе работы процедуры выполнен больше некоторого заданного числа раз, вернуть неуспех и отказать в назначении сообщения.

7. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ СВОЙСТВ АЛГОРИТМА

Так как бортовые сети являются сетями небольшого размера и, кроме того, обычно представляют собой топологию типа “звезда”, для экспериментальных исследований были выбраны сети AFDX, изображенные на рис. 5 и 6.

Сеть AFDX, изображенная на рис. 5, является стандартной сетью, соединяющей абонентов из различных частей самолета. Сеть на рис. 6 при этом расширяет стандартную, добавляя избыточные коммутаторы и каналы передачи данных.

Исследования проводились для исследования следующих свойств алгоритма¹⁰:

- эффективность алгоритма без использования процедуры агрегации;
- эффективность алгоритма без использования процедуры переконфигурации виртуального канала;
- эффективность алгоритма без использования процедуры ограниченного перебора;
- эффективность алгоритма с использованием всех процедур. В данном случае использовалась процедура ограниченного перебора глубины 2.

¹⁰В следствие того, что авторами не было найдено в свободном доступе каких-либо доступных реализаций существующих подходов к решению задачи (и, кроме того, как было показано в разделе 4, не было найдено подходов, полностью решающих поставленную задачу), эксперименты ставились только для исследования свойств разработанного алгоритма.

Таблица 1. Исследования свойств алгоритма для стандартной топологии

Свойства запускаемого алгоритма	Средний процент назначенных сообщений (%)			
	1-ый класс сообщений	2-ой класс сообщений	3-ий класс сообщений	4-ый класс сообщений
Без процедуры агрегации	17	54	61	54
Без процедуры переконфигурации	100	99	55	83
Без процедуры ограниченного перебора	100	99	68	87
С использованием всех процедур	100	99	68	87

Таблица 2. Исследования свойств алгоритма для топологии с избыточными каналами передачи данных

Свойства запускаемого алгоритма	Средний процент назначенных сообщений (%)			
	1-ый класс сообщений	2-ой класс сообщений	3-ий класс сообщений	4-ый класс сообщений
Без процедуры агрегации	17	54	64	55
Без процедуры переконфигурации	100	99	64	88
Без процедуры ограниченного перебора	100	99	77	92
С использованием всех процедур	100	99	77	92

Исследования проводились для трех классов автоматически генерируемых наборов сообщений:

1. Большое количество сообщений с низкими требованиями к длительности передачи. Генерировалось 2000 сообщений с размером от 1 до 1000 байт, передаваемые с периодом от 1 до 1000 сек и с ограничением длительности передачи от 0.1 до 100 сек.
2. Сообщения большого размера. Генерировалось 100 сообщений размером от 1кб до 1мб с периодом от 1 до 10 сек и с ограничением длительности передачи от 1 до 10 сек.
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.
4. Смешанный класс сообщений. Генерировалось 100 сообщений из 1 класса и по 50 сообщений из 2 и 3 класса соответственно.

Для каждого исследуемого свойства проводился запуск на 100 случайно сгенерированных наборов сообщений, после чего результаты усреднялись. Результаты исследования указанных выше свойств для всех классов для стан-

дартной топологии указаны в таблице 1, для топологии с избыточными каналами передачи данных — в таблице 2.

Исходя из представленных результатов, можно сделать выводы, что существенно улучшить результат помогает процедура агрегации. Для каждого класса сообщений эта процедура позволила значительно увеличить среднее число назначаемых сообщений.

Процедура переконфигурации позволяет улучшить качество работы алгоритма для 3-его (и, соответственно, для смешанного) класса сообщений, так как именно в этом случае более всего нарушаются ограничения на длительность передачи сообщений. Добавление избыточных каналов передачи данных позволило улучшить результат для 3-его и 4-ого классов сообщений благодаря большей возможности строить маршруты в обход уже построенных маршрутов с целью уменьшения загрузки каналов передачи данных и длительности передачи сообщений.

Процедура ограниченного перебора не позволила улучшить результаты для проводимых тестов. Это можно пояснить тем, что предоставляемая пропускная способность является в большинстве случаев наименее ограничивающим фактором в сравнении с ограничением на максимальный джиттер и на длительность передачи сообщений.

8. ЗАКЛЮЧЕНИЕ

Предложенный в работе алгоритм для исходно заданного множества периодических сообщений осуществляет построение множества виртуальных каналов и построение для них маршрутов в физической сети обмена данными. В алгоритме может быть использован любой из известных методов для расчета значений параметров виртуального канала и любой из известных методов для вычисления оценки максимальной длительности и джиттера передачи сообщений. Это также позволяет настраивать алгоритм на особенности коммутаторов AFDX различных производителей.

СПИСОК ЛИТЕРАТУРЫ

1. *Костенко В.А., Гурьянов Е.С.* Алгоритм построения расписаний обменов по шине с централизованным управлением и исследование его эффективности // Программирование. 2005. № 6. С. 67–76.
2. *Kostenko V.A., Plakunov A.V.* An Algorithm for Constructing Single Machine Schedules Based on Ant Colony Approach. *J. of Computer and Systems Sciences International*, 2013, vol. 52, no. 6, pp. 928–937.
3. *Balashov V., Kostenko V., Vdovin P., Smeliansky R., Shalimov A.* An Analysis of Approaches to Onboard Networks Design. *Proceedings of the International Science and Technology Conference Modern Networking Technologies (MoNeTec)*, IEEE. Moscow, Russia, MAKS Pess, 2014, pp. 20–24.
4. Specification A. R. I. N. C. 664: Aircraft data network. Part 7. Aeronautical Radio, Inc., 2005.
5. *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. pp. 413–438.
6. *Guangyu H. et al.* Towards Tightening Worst Case End-to-End Delay Estimates for AFDX Network, 2011.
7. *Boyer M., Fraboul C.* Tightening end to end delay upper bound for afdx network calculus with rate latency fifo servers using network calculus. *Factory Communication Systems*, 2008. WFCS 2008. IEEE International Workshop on. IEEE, 2008, pp. 11–20.
8. *Frances F., Fraboul C., Grieu J.* Using network calculus to optimize the AFDX network, 2006.
9. *Bauer H., Scharbarg J.L., Fraboul C.* Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network. *Emerging Technologies & Factory Automation*, 2009. ETFA 2009. IEEE Conference on. IEEE, 2009, pp. 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. vol. 48, no. 1, pp. 101–133.
11. *Al Sheikh A. et al.* Optimal design of virtual links in AFDX networks. *Real-Time Systems*, 2013. vol. 49, no. 3, pp. 308–336.
12. *Gilbert E., Pollak H.* Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 1968, vol. 16, no. 1, pp. 1–29.
13. *Vdovin P.M., Kostenko V.A.* Algorithm for Resource Allocation in Data Centers with Independent Schedulers for Different Types of Resources. *J. of Computer and Systems Sciences Intern*, 2014, vol. 53, no. 6. pp. 854–866.
14. *Zotov I.A., Kostenko V.A.* Resource Allocation Algorithm in Data Centers with a Unified Scheduler for Different Types of Resources. *J. of Computer and Systems Sciences International*, 2015, vol. 54, no. 1, pp. 59–68.
15. *Seok Y. et al.* Explicit multicast routing algorithms for constrained traffic engineering. *Computers and Communications*, 2002. *Proceedings. ISCC 2002. Seventh International Symposium on. IEEE*, 2002, pp. 455–461.