

ПОСТРОЕНИЕ БОРТОВОЙ КОММУТИРУЕМОЙ СЕТИ С ВРЕМЕННОЙ СИНХРОНИЗАЦИЕЙ МИНИМАЛЬНОЙ СЛОЖНОСТИ

© 2022 г. В. А. Костенко^{а,*} (ORCID: 0000-0002-7895-2322),

А. А. Морквин^{а,**} (ORCID: 0000-0001-6253-054X)

^а *Московский государственный университет имени М.В. Ломоносова*

Кафедра автоматизации систем вычислительных комплексов

119992, Россия, Москва, Ленинские горы, д. 1, строение 52

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

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

Поступила в редакцию 12.05.2022 г.

После доработки 16.07.2022 г.

Принята к публикации 20.07.2022 г.

В статье сформулирована задача построения бортовой коммутируемой сети с временной синхронизацией минимальной сложности необходимой для передачи периодических сообщений в реальном времени и предложены алгоритмы ее решения: построения структуры сети и системы расписаний передач сообщений. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей на основе стандарта Time-Sensitive Networking (TSN).

DOI: 10.31857/S013234742206005X

1. ВВЕДЕНИЕ

Современные информационно управляющие системы реального времени (ИУС РВ) являются распределенными и включают в свой состав различные устройства, которые взаимодействуют между собой. В ИУС РВ можно выделить три уровня обработки данных: 0) уровень преобработки, 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: ARINC 651 – основные принципы построения ИУС РВ на основе ИМА [1]; ARINC 653 – спецификация операционных систем [2]; FC-AE-ASM-RT – спецификация сети информационного обмена на основе коммутируемой сети Fibre Channel [3]; ARINC 664 (AFDX) – спецификация сети информационного обмена на основе Ethernet [4].

Множество стандартов Time-Sensitive Networking (TSN) [5] является расширением стандарта IEEE 802.1 Audio Video Bridging (AVB) [6], который был впервые опубликован в 2005 году, и разработано для управления сетевым трафиком, чтобы обеспечивать строго определенные временные задержки при передаче данных. Для этого все устройства в TSN сети должны быть синхронизированы друг с другом относительно единого эталона времени для поддержания связи в реальном времени. На данный момент множество стандартов и его спецификации под конкретные области использования продолжают активно разрабатываться. Однако уже есть ИУС РВ, сети обмена которых построены на основе этих стандартов: ИУС РВ автоматизированной производственной линии [7] и сеть беспилотного автомобиля [8]. Гарантия низких сквозных задержек делают TSN сети актуальными [9] для использования в ИУС РВ.

В работе [10] 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 вычислительной 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. Минимизация сложности сети с сохранением способности пере-

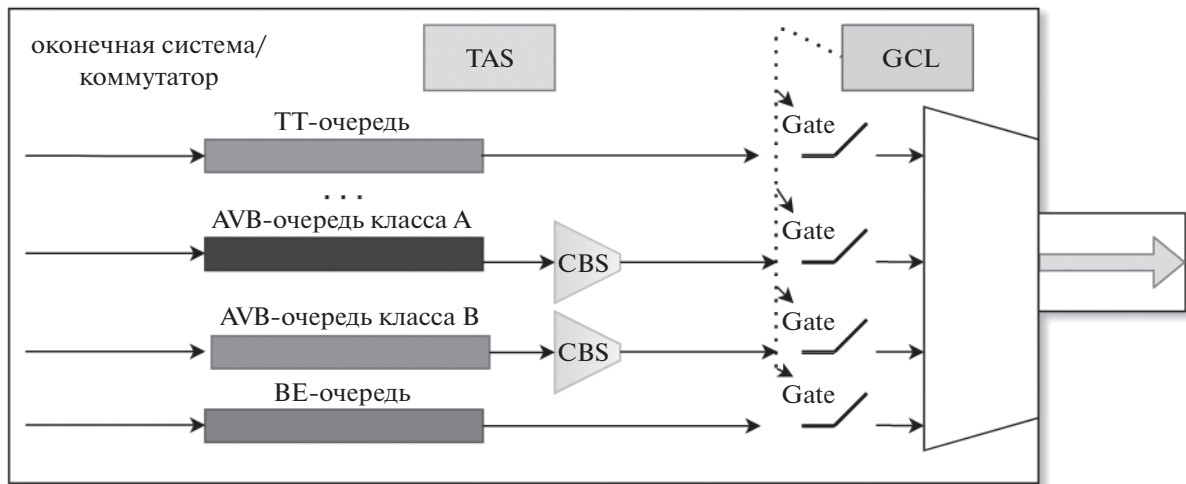


Рис. 1. Time-Aware Shaper (*TAS*).

давать заранее заданный трафик реального времени является актуальной задачей проектирования ИУС РВ с архитектурой ИМА. С точки зрения надежности, чем меньше суммарная длина кабелей в сети, тем меньше перекрестные наводки, и, следовательно, меньше вероятность ошибок при передаче данных [11].

В данной работе сформулирована задача построения бортовой сети обмена минимальной сложности (минимизируется суммарная длина физических соединений) на основе стандарта TSN необходимой для передачи в реальном времени заранее заданного множества периодических сообщений и предложены алгоритмы ее решения: построения структуры сети и системы расписаний передач. Приводятся результаты апробации предложенных алгоритмов для построения бортовых сетей.

2. ОПИСАНИЕ TSN-СЕТЕЙ

TSN-сеть определяет механизм передачи трафика с требованиями к длительности передачи на основе Ethernet сетей. Ключевой особенностью является временная синхронизация всех устройств в сети на основе протокола Precision Time Protocol (PTP), который позволяет синхронизировать внутренние часы каждого устройства с разницей от 10 нс до 1 мкс. Такая точность достигается за счет “аппаратной” поддержки синхронизации, которая точно учитывает задержки. Также TSN-сеть позволяет осуществлять надежную доставку сообщений за счет возможности использования нескольких маршрутов передачи сообщений в сети.

Согласно [5] существует три вида трафика в сети:

1. Time-Triggered traffic (*TT*-трафик) – трафик с гарантией низких сквозных задержек и доставки без потерь.

2. Audio-Video-Bridging traffic (*AVB*-трафик) – трафик с ограниченными сквозными задержками. Данный тип трафика является менее критичным, чем *TT*-трафик, и состоит из двух подтипов – класса *A* и класса *B*, которые отличаются приоритетом.

3. Best-Effort traffic (*BE*-трафик) – негарантированный трафик без гарантий доставки и ограничений на задержку.

Рассмотрим подробнее вопрос планирования передачи сообщения. На рисунке 1 изображен планировщик Time-Aware Shaper (*TAS*), который ассоциируется с каждым выходным портом коммутатора. Согласно стандарту [5] для планирования передачи доступно восемь очередей (с номерами соответственно от 1 до 8), одна и более могут быть зарезервированы под *TT*-трафик, две соответственно под класс *A* и *B* *AVB*-трафика, оставшиеся могут быть зарезервированы под *BE*-трафик.

С каждой очередью ассоциированы “задвижки” (*Gate*), которые могут находиться в двух состояниях – открыты или закрыты. Состояние в каждый момент времени определяется благодаря наличию расписания Gate Control List (*GCL*), которое представляет собой циклическое расписание открытия и закрытия всех *Gate* на выходном порту и длительность соответствующего открытия или закрытия. Пример *GCL* можно увидеть на рисунке 2, где T_i – длительность интервала, а битовый массив представляет собой состояние соответствующих *Gate* в данный промежуток времени.

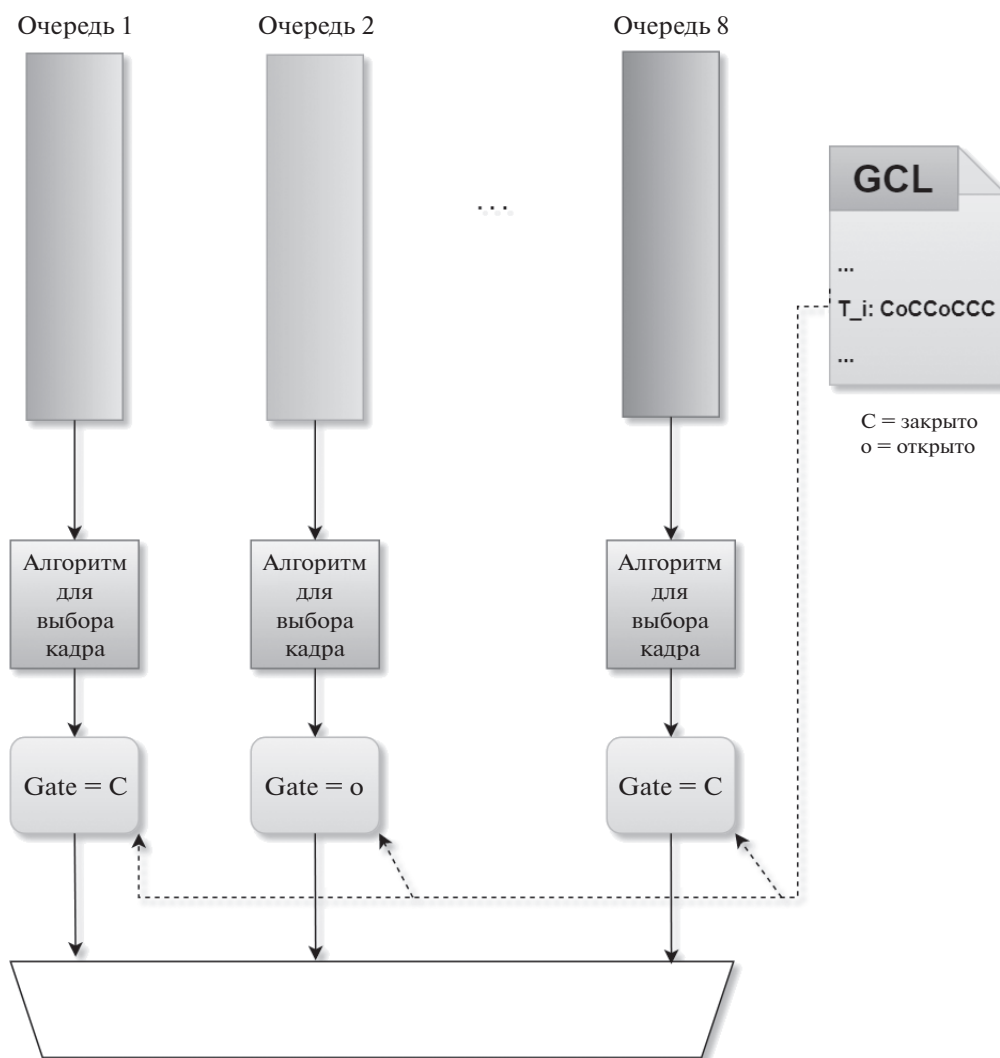


Рис. 2. Пример GCL.

Планировщик *TAS* при выборе кадра для передачи руководствуется следующим правилом — выбирается кадр из очереди с наивысшим приоритетом (соответственно с наименьшим номером) среди всех очередей, у которых *Gate* открыты и которые не пусты. Однако для проверки доступности кадров *AVB*-трафика *TAS* использует механизм, про который будет написано далее.

При открытии очереди *TT*-трафика может возникнуть ситуация наложения передачи кадров. Стандартом [5] предусмотрено два механизма предотвращения этой проблемы:

1. Встраивание защитных интервалов перед каждым открытием *TT*-очереди. Размер интервала зависит от максимального размера кадра других сообщений и в худшем случае выставляется для 1542 байт. Во время этого интервала запрещено начинать передачу кадров.

2. Прерывание передачи с последующим продолжением с места прерывания. В данном случае для второй части кадра добавляется дополнительный заголовок в 24 байта, который требуется для склейки кадра.

Для передачи *AVB*-трафика используется механизм *Credit-Base Shaper (CBS)*, который допускает выборку кадра из очереди при выполнении следующих условий:

1. Доступен кредит для передачи.
2. Не передаются кадры с более высоким приоритетом.

Этот механизм необходим для предотвращения остановки потоков с более низким приоритетом.

Кредит уменьшается на величину $sendSlop\left(\frac{\text{байт}}{c}\right)$ при передаче кадра соответствующей очереди, а увеличивается при ожидании передачи на вели-

чину $idleSlop\left(\frac{\text{байт}}{c}\right)$. При закрытии *Gate* кредит замораживается. Передача возможна только при неотрицательном кредите.

3. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Введем следующие множества: N – множество точек размещения оконечных систем, K – множество возможных точек размещения коммутаторов, E_{sw} – множество возможных соединений между коммутаторами, E_{end} – множество возможных соединений между оконечными системами и коммутаторами, V – множество длин соединений. Для каждого соединения $e \in E_{sw} \cup E_{end}$ 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. $type_{msg} \in \{TT, A, B\}$ – принадлежность сообщения к определенному типу трафика (либо *TT*-сообщение, либо *AVB*-сообщение класса *A* или *B*).
2. T_{msg} (мс) – период передачи сообщения.
3. $size_{msg}$ (байт) – размер сообщения.
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 начала периода до момента получения всеми абонентами-получателями не превосходит данное значение.

Пусть $G = (N \cup K, E, V)$ – *максимальная сеть*, которая включает в себя все коммутаторы из множества K и все соединения из множества $E = E_{sw} \cup E_{end}$. Предлагается решать задачу минимизации сложности сети путем нахождения подмножества максимальной сети достаточного для передачи множества MSG .

Сеть $G^* = (N^* \cup K^*, E^*, V^*)$ – является *подсетью* максимальной сети G : $N^* = N$ – множество оконечных систем (нельзя исключить оконечную систему без уменьшения множества передаваемых

сообщений); $K^* \subseteq K$ – подмножество коммутаторов; $E^* \subseteq E$ – подмножество соединений; $V^* \subseteq V$ – длина всех соединений $e \in E^*$.

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

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

Далее под минимальной сетью будем понимать сеть с минимальным значением S .

Построить расписание передачи множества сообщений MSG^* для сети G^* означает следующее: для каждого выходного порта коммутатора $k \in K^*$ построить GCL так, что будут выполняться ограничения на длительность передачи для каждого $msg \in MSG^*$. При этом важно понимать, что каждое сообщение из MSG^* имеет свой маршрут передачи. При построении GCL будем соблюдать следующее ограничение:

1. В любой момент времени в каждой *TT*-очереди могут находиться кадры только одного сообщения. Это ограничение вводится для изоляции потоков кадров разных сообщений, которые могут поступить на входной порт коммутатора в одно и то же время. Из-за этого может возникать джиттер и нарушаться длительность доставки.

Решение задачи заключается в построении минимальной сети G^* и расписания передач для максимального подмножества $MSG^* \subseteq MSG$, которое может быть передано через максимальную сеть.

Таким образом, имеется два критерия для оптимизации: $\max(|MSG^*|)$ и $\min(S(G^*))$. Чтобы сделать задачу оптимизации однокритериальной, оставим один критерий $\min(S(G^*))$, а другой преобразуем в ограничение $|MSG^*| = |MSG^{max}|$, где MSG^{max} – это максимальное множество сообщений для максимальной сети.

Следовательно, задача оптимизации формулируется следующим образом:

$$\begin{aligned} & \min(S(G^*)) \\ & G^* \subseteq G; \quad |MSG^*| = |MSG^{max}|; \\ & \forall msg \in MSG^* : Dur(msg) \leq t_{msg} \end{aligned}$$

Здесь $Dur(msg)$ – значение длительности передачи сообщения, которое может быть оценено по методам в следующих работах [12, 13].

4. АЛГОРИТМ ПОСТРОЕНИЯ TSN-СЕТИ МИНИМАЛЬНОЙ СЛОЖНОСТИ

Общая схема алгоритма имеет следующие шаги:

Шаг 1. Из входных данных создать максимальную сеть G .

Шаг 2. Отсортировать множество сообщений MSG (по убыванию требуемой пропускной способности).

Шаг 3. Для каждого сообщения $msg \in MSG$:

Шаг 3.1. Выполнить процедуру построения маршрута и получить маршрут, который обладает минимальной суммарной длиной используемых соединений. В случае неуспеха перейти к следующему сообщению.

Шаг 3.2. Выполнить процедуру построения расписания передачи сообщения. В случае успеха перейти на шаг 3.4.

Шаг 3.3. Выполнить процедуру ограниченного перебора для маршрутов уже назначенных сообщений, которая заключается в попытках поиска других маршрутов передачи для уже назначенных сообщений. В случае неуспеха перейти к следующему сообщению.

Шаг 3.4. Выполнить процедуру оценки длительности доставки сообщения. В случае успеха перейти к следующему сообщению.

Шаг 3.5. Выполнить процедуру ограниченного перебора соединений в маршруте сообщения, которая заключается в попытке “обхода” соединений с большим временем ожидания передачи кадров сообщения. В случае неуспеха перейти к другому сообщению.

Шаг 4. Удалить из максимальной сети G неиспользуемые соединения (соединения, через которые не проходит ни один маршрут) и неиспользуемые коммутаторы (коммутаторы без подключенных соединений). Полученная сеть является результатом работы алгоритма.

Процедура построения маршрута

Маршрут представляет собой дерево, являющееся подграфом максимальной сети G , корнем дерева является оконечная система-отправитель, листьями – оконечные системы-получатели сообщения.

Для построения маршрута выполняются следующие шаги:

Шаг 1. Если в максимальной сети между оконечной системой-отправителем и хотя бы одной из оконечных систем-получателей не существует пути, то процедура возвращает неуспех.

Шаг 2. Запустить алгоритм Дейкстры от оконечной системы-отправителя. Остановить алгоритм после нахождения кратчайшего пути до одной из оконечных систем-получателей, до которых еще не найден путь. Будем использовать следующий критерий веса для ребра – $l(e) = \frac{V_e}{k+1}$, где $e \in E$, V_e – длина соединения e , а k – число уже назначенных на это соединений сообщений.

Шаг 3. Если найдены пути до всех получателей, то вернуть объединение найденных маршрутов. Иначе запомнить путь до найденного получателя и перейти на шаг 2.

Процедура построения расписания передачи сообщения

Для каждого узла (либо оконечная-система отправитель, либо коммутатор) в маршруте сообщения выполнить следующие шаги:

Шаг 1. Для каждого участвующего в передаче сообщения выходного порта узла маршрута добавить в GCL передачу этого сообщения. В случае невозможности добавить в GCL -передачу этого сообщения вернуть неуспех, иначе перейти к другому узлу маршрута.

Процедура ограниченного перебора маршрутов уже назначенных сообщений

Шаг 1. Для каждого множества назначенных сообщений мощности не больше заданного значения m (то есть рассматриваются всевозможные сочетания из множества назначенных сообщений по m элементов) выполнить шаги 1.1–1.5:

Шаг 1.1. Снять все маршруты и расписания передач сообщений из выбранного множества.

Шаг 1.2. Выполнить процедуру построения расписания передачи сообщения для msg (см. шаг 3 общей схемы алгоритма). В случае неуспеха перейти на шаг 1.5.

Шаг 1.3. Выполнить процедуру построения маршрута, процедуру построения расписания передачи и процедуру оценки длительности доставки сообщения для всех снятых сообщений. Если хотя бы одно из сообщений не может быть назначено, то снять сообщение msg и перейти на шаг 1.5.

Шаг 1.4. Выполнить назначения сообщений согласно найденным маршрутам и построенным расписаниям. Вернуть успех.

Шаг 1.5. Восстановить изначальные назначения маршрутов и расписаний сообщений. Перейти к другому множеству сообщений.

Шаг 2. Если после ограниченного перебора построить расписание для msg не удастся, то назначение сообщения считается неуспешным.

Параметр m – входной параметр алгоритма.

Процедура ограниченного перебора соединений в маршруте сообщения

Шаг 1. Сформировать $\{Paths_i\}$, $i = \overline{1, p}$ из маршрутов, которые определяются следующим образом ($Paths_i$ – i -минимальный маршрут (под минимальность понимаем сумму критериев веса ребер в маршруте), а p – заранее заданное число):

Шаг 1.1. Для каждого множества соединений из маршрута сообщения мощности не больше заданного значения h , то есть рассматриваются все-

возможные сочетания из множества соединений маршрута по h элементов:

Шаг 1.1.1. Принять длины соединений из выбранного множества за достаточно большое число.

Шаг 1.1.2. Выполнить процедуру построения маршрута для данного сообщения msg . Сохранить маршрут в $\{Paths_i\}$, если новый маршрут является новым i -минимумом.

Шаг 1.1.3. Восстановить изначальные длины соединений из выбранного множества. Перейти к следующему множеству.

Шаг 2. Для каждого маршрута из $\{Paths_i\}$:

Шаг 2.1. Выполнить процедуру построения расписания передачи для данного сообщения msg . В случае успеха перейти на шаг 2.3.

Шаг 2.2. Выполнить процедуру ограниченного перебора маршрутов уже назначенных сообщений. В случае неуспеха перейти к другому маршруту.

Шаг 2.3. Выполнить процедуру оценки длительности доставки сообщения msg . В случае успеха вернуть новый маршрут и новое расписание передачи.

Шаг 3. Если после ограниченного перебора добиться выполнения ограничения на длительность доставки для msg не удастся, то назначение сообщения считается неуспешным.

Параметры p и h – входные параметры алгоритма.

Процедура оценки длительности доставки сообщения

Длительность передачи сообщения msg вычисляется следующим образом:

$$Dur(msg) = \mu + \Delta$$

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 – максимальная длительность передачи последнего кадра.

Для TT -трафика величина Δ может быть вычислена при построении расписания передач для данного сообщения, так как для TT -трафика на каждом порту известно время получения первого кадра и время ухода последнего кадра из очереди.

Для AVB -трафика величина Δ может быть вычислена с помощью одного из методов в работах [12, 13]. Рекомендуется использовать метод из работы [12], так как в данной работе учитывается влияние TT -трафика на длительность передачи

AVB -трафика, поэтому оценка получается точнее, чем в других работах.

Процедура выполняет проверку ограничения на длительность передачи сообщения и возвращает успех или неуспех.

5. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ СВОЙСТВ АЛГОРИТМА

Цель экспериментального исследования – проверить эффективность предложенного алгоритма по критерию минимизации сложности сети. При этом по сети должны передаваться в реальном времени все изначальные заданные периодические сообщения. Для достижения этой цели достаточно формировать такие наборы сообщений, для которых известно, какая подсеть достаточна для передачи всех сообщений.

Наборы данных для экспериментов строились следующим образом:

1. Выбирается базовая топология сети.
2. Создается набор сообщений, который может быть передан через базовую сеть, создавая достаточно высокую нагрузку на сеть.
3. Базовая сеть расширяется до максимальной сети с дополнительными коммутаторами и соединениями так, чтобы для передачи сообщений было доступно несколько дополнительных маршрутов.

Работа алгоритма считается успешной, если для такого набора сообщений максимальная сеть была сокращена до базовой сети или до другой сети со значением сложности равным сложности базовой сети.

Три пары (базовая; максимальная) сетевых топологий, которые использовались в исследовании, изображены на рисунке 3 и 4. Сплошные квадраты, круги и линии изображают соответственно коммутаторы, оконечные системы и соединения базовой сети. Вместе с пунктирными они образуют максимальную сеть. Третья топология (рисунок 4) взята из [14] и сокращена до базовой, удалив некоторые избыточные связи между коммутаторами. Топологии типичны для бортовых сетей, в частности, третья используется в Airbus A380.

Для каждой из трех топологий сети были сгенерированы наборы сообщений следующих типов:

1. Размер: 16 Байт – 1 КБ; период: 10 мс – 1 с; максимальная длительность доставки: 10 мс – 100 мс.
2. Размер: 1 КБ – 100 КБ; период: 100 мс – 10 с; максимальная длительность доставки: 10 мс – 1 с.
3. Произвольная смесь 1 и 2 класса.

Класс 1 соответствует трафику управления и навигации в бортовых сетях. Класс 2 соответствует “медиа”-трафику, такому как изображения с метеорологического радара. Класс 3 представляет

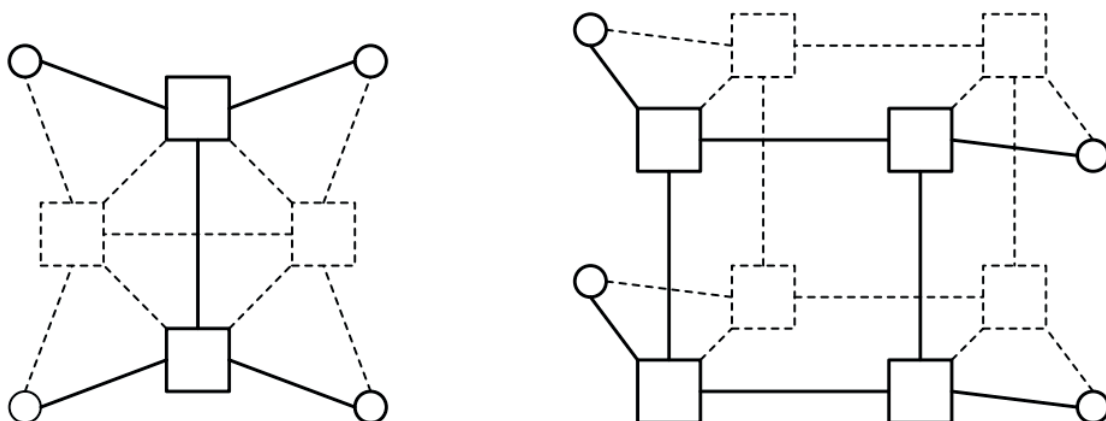


Рис. 3. Первая и вторая пара (базовая; максимальная) сетевых топологий.

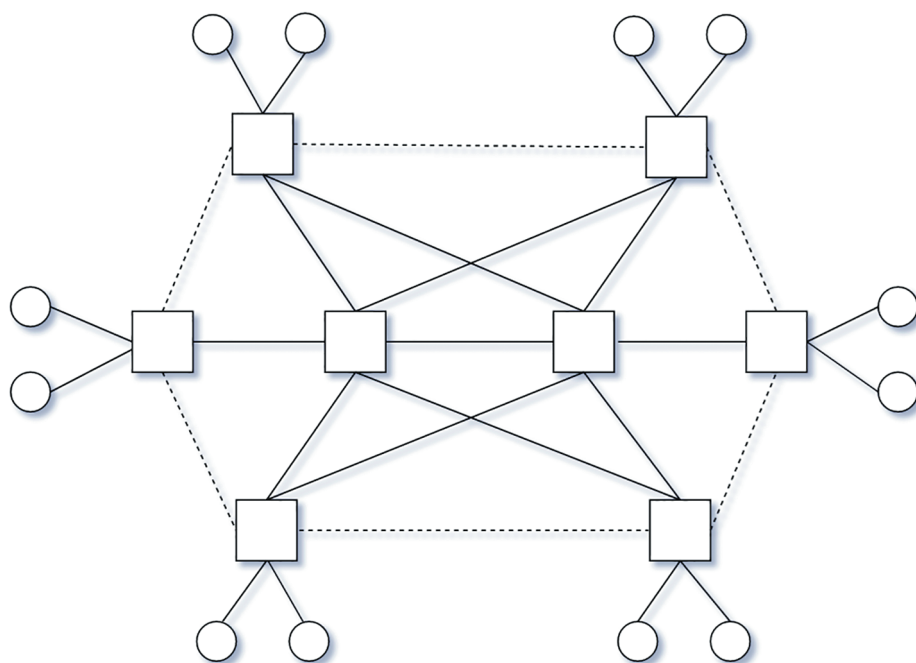


Рис. 4. Третья пара (базовая; максимальная) сетевых топологий.

гетерогенный трафик. Для каждого типа набора сообщений параметры были выбраны случайным образом в указанных диапазонах.

Для каждой пары (топология сети, тип набора сообщений) было сгенерировано и использовано в экспериментах 10 наборов сообщений. В зависимости от типа набора сообщений и конфигурации сети набор сообщений включал от 100 до 300 периодических сообщений, что согласуется с имеющимися данными о характеристиках трафика в бортовых сетях [14].

Эксперименты проводились с параметрами алгоритма: $m = 2$, $h = 4$, $p = 3$. Параметр m был выбран так, чтобы алгоритм не превращался в пе-

реборный. Параметр h был выбран таким из эвристических соображений, чтобы рассматривались достаточно “различные” маршруты передачи. Параметр p также был выбран из эвристических соображений, чтобы процедура рассматривала как максимум три различных маршрута.

Результаты экспериментального исследования представлены на рисунке 5. На рисунке видно, что в большинстве случаев для 100% сообщений успешно строится расписание передач. Максимальная сеть в большинстве случаев также успешно упрощается до базовой сети. Даже для третьей топологии, которая имеет большое количество альтернативных маршрутов для передачи

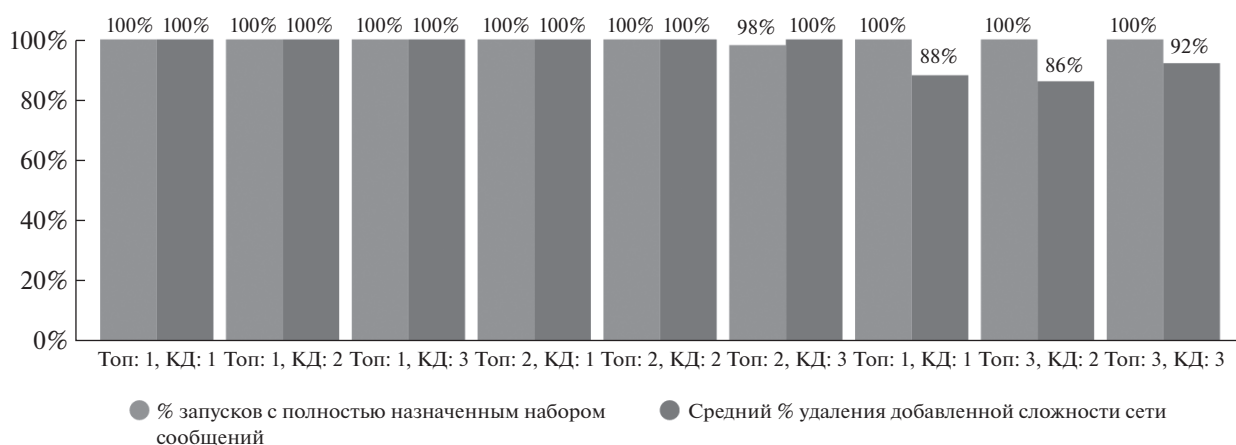


Рис. 5. Результаты экспериментального исследования для TSN сетей. (Топ:N – топология N. КД:K – класс данных K).

каждого сообщения, имеем более чем 85% удаление избыточной сложности сети.

Для 2 топологии и 3 класса данных имеем 2% запусков, где не удалось назначить полностью набор сообщений, что обуславливается тем, что алгоритм является жадным и процедуры ограниченного перебора не всегда помогают получить точное решение за оптимальное время.

6. ЗАКЛЮЧЕНИЕ

Предложенный в работе алгоритм для исходно заданного множества периодических сообщений осуществляет построение коммутируемой сети обмена с временной синхронизацией минимальной сложности необходимой для передачи сообщений в реальном времени и строит систему расписаний передач.

Результаты экспериментального исследования показали, что представленный алгоритм успешно удаляет избыточную сложность сети (не менее 85% сокращение избыточности).

СПИСОК ЛИТЕРАТУРЫ

1. ARINC Specification 651. Design Guidance for Integrated Modular Avionics. Airlines Electronic Engineering Committee, 1997.
2. ARINC Specification 653. Avionics Application Software Standard Interface. Airlines Electronic Engineering Committee, 2007.
3. INCITS 373. Information Technology – Fibre Channel Framing and Signaling Interface (FC-FS). International Committee for Information Technology Standards, 2003.
4. ARINC Specification 664. Aircraft Data Network, Part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network. Airlines Electronic Engineering Committee, 2005.
5. IEEE 802.1Q – Local and Metropolitan Area Networks – Bridges and Bridged Networks. Institute of Electrical and Electronics Engineers, 2018.
6. IEEE 802.1BA-2011 – IEEE Standard for Local and metropolitan area networks – Audio Video Bridging (AVB) Systems. Institute of Electrical and Electronics Engineers, 2011.
7. D. Bruckner et al. An Introduction to OPC UA TSN for Industrial Communication Systems. in Proceedings of the IEEE. 2019. V. 107. № 6.
8. Lee J., Park S. Time-Sensitive Network (TSN) Experiment in Sensor-Based Integrated Environment for Autonomous Driving. Sensors, 2019.
9. Lin Zhao, Feng He, Ershuai Li, Jun Lu. Comparison of Time Sensitive Networking (TSN) and TTEthernet. IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), 2018.
10. Костенко В.А. Архитектура программно-аппаратных комплексов бортового оборудования. Изв. вузов. Приборостроение. 2017. Т. 60. № 3.
11. Al-Kuwaiti M., Kyriakopoulos N., Hussein S. A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability. IEEE Communications Surveys & Tutorials. 2009. V. 11. № 2.
12. Zhao L., Pop P., Zheng Z., Li Q. Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus. 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2018.
13. Sune Mølgaard Laursen, P. Pop, W. Steiner. Routing optimization of AVB streams in TSN networks. ACM SIGBED Review, 2016.
14. Amari A., Mifdaoui A. Specification and Performance Indicators of AeroRing – A Multiple-Ring Ethernet Network for Avionics Embedded Systems. Sensors. 2018. V. 18.