

Токарева Е.А., Балашов В.В.

СОВМЕСТНОЕ ПОСТРОЕНИЕ МАРШРУТОВ И РАСПИСАНИЙ ПЕРЕДАЧИ СООБЩЕНИЙ В СЕТЯХ ETHERNET С ВРЕМЕННОЙ СИНХРОНИЗАЦИЕЙ¹

Введение

В данной работе рассматриваются информационно-управляющие системы реального времени, в которых взаимодействие вычислительных задач осуществляется с использованием сети Ethernet с временной синхронизацией [1]. Такие сети примечательны наличием класса сообщений с отправкой по таймеру. Для обмена данными в таких системах необходимо наличие информации о маршрутах передачи данных, а также заранее составленного расписания получения и отправки сообщений на узлах сети.

Существующие методы [2] построения расписания получают на вход предварительно построенные маршруты передачи данных. Маршруты играют ключевую роль при построении расписания, так как при наличии большого количества сообщений, передающихся по одному и тому же физическому каналу, возникают случаи, в которых невозможно построение корректного расписания. При этом зачастую при альтернативном выборе маршрутов построение корректного расписания возможно. В данной статье предложен алгоритм совместного построения маршрутов и расписания передачи данных, то есть маршруты выбираются одновременно с построением расписания и корректируются, если в какой-то момент для выбранных маршрутов построение корректного расписания неуспешно.

Структура данной статьи следующая. Раздел 1 содержит формальную постановку задачи построения маршрутов и расписания передачи данных в сетях Ethernet с временной синхронизацией. В разделе 2 представлен обзор существующих методов, решающих поставленную задачу. Раздел 3 содержит методы решения поставленной задачи, раздел 4 - описание предлагаемого алгоритма совместного построения маршрутов и расписания передачи данных. В разделе 5 приведено экспериментальное исследование алгоритма.

¹Работа выполнена при финансовой поддержке РФФИ, грант № 16-07-01237.

1. Схема функционирования сетей с временной синхронизацией

Сети с временной синхронизацией [3] расширяют классический Ethernet. Такие сети состоят из узлов сети, соединенных физическими каналами. Узлы сети представлены набором абонентов и набором коммутаторов. Абонент состоит из коммуникационного контроллера и хоста, на котором выполняются прикладные задачи. Задачи обмениваются сообщениями.

Существует три класса сообщений:

1. ТТ. Такие сообщения пересылаются сетью в предопределенное время и более приоритетны, чем все остальные типы сообщений. Для каждого такого сообщения предопределены моменты времени его отправки с каждого узла сети его маршрута. Следовательно, для него предопределена и сквозная задержка. При планировании передачи таких сообщений обеспечивается отсутствие очередей на коммутаторах, что позволяет минимизировать задержку при передаче сообщений. Для реализации такого типа сообщений требуются синхронизированные локальные часы на всех узлах сети.

2. РС. Передача сообщений организована по схеме, аналогичной схеме передачи, используемой в протоколе AFDX [4], с учетом наличия в сети более приоритетных ТТ-сообщений.

3. ВЕ. Сообщения, пересылаемые в соответствии со стандартной политикой Ethernet, имеют наименьший детерминизм задержек при передаче через сеть.

Отправка ТТ-сообщений происходит согласно заранее построенному расписанию, гарантирующему отсутствие конфликтов, по синхронизированным таймерам. РТ-сообщения отправляются в свободное от отправки ТТ-сообщений время и не влияют на ТТ-сообщения. ВЕ-сообщения используют оставшуюся пропускную способность.

В случае если на порте коммутатора уже происходит отправка сообщения РС или ВЕ на момент наступления времени отправки с него ТТ-сообщения, текущая отправка отменяется, производится отправка ТТ-сообщения, после чего повторяется отмененная отправка. Таким образом, максимальное время обработки ТТ-сообщения на коммутаторе заранее известно и учитывается на этапе построения расписания.

В данной работе рассматривается только передача сообщений класса ТТ.

Для выполнения информационного обмена с использованием ТТ-сообщений необходимо построить локальное расписание отправки ТТ-сообщений для каждого узла сети.

2. Формальная постановка задача построения маршрутов и расписания передачи данных в сетях с временной синхронизацией

Входными данными для задачи построения маршрутов и расписания передачи данных в системах реального времени, использующих для передачи данных сети Ethernet с временной синхронизацией, являются структура сети и поток данных в этой сети.

Структура сети задается ориентированным графом $G = (V, E)$, где

- $V = A \cup K$ - множество узлов сети, каждый узел определяется множеством портов $V.P = \{ P_1, P_2, \dots, P_{N.P_A} \}$;
- A - множество абонентов: $A = \{ A_1, A_2, \dots, A_{N_A} \}$, $|A| = N_A$;
- K - множество коммутаторов: $K = \{ K_1, K_2, \dots, K_{N_K} \}$, $|K| = N_K$;
- E - множество дуг, соединяющих элементы из множества V : $E = \{ E_1, E_2, \dots, E_{N_E} \}$, $|E| = N_E$. При этом для каждой направленной дуги $E_i = (V_a, V_b)$ существует парная дуга $E_j = (V_b, V_a)$. Такие направленные дуги называются каналами.

Поток данных в сети задается множеством однокадровых сообщений $Msg = \{ Msg_1, Msg_2, \dots, Msg_{N_{Msg}} \}$, $|Msg| = N_{Msg}$. Каждое сообщение задается четверкой параметров $Msg = \{ T_{msg}, Src_{msg}, Dst_{msg}, t_{msg} \}$, где

- T_{msg} - период отправки сообщения;
- $Src_{msg} \in A$ - абонент отправитель;
- $Dst_{msg} \in A$ - абонент получатель;
- t_{msg} - максимальная допустимая сквозная длительность передачи сообщения точка-точка.

Обозначим через Msg_i^n экземпляр сообщения Msg_i , передающийся в интервал времени $[n * T_{msg_i}, (n+1) * T_{msg_i})$.

Так как прикладные задачи отправляют данные периодически, то достаточно построить расписание только на конечном интервале времени. Далее вновь используется готовое расписание. Такой интервал называется интервалом планирования, его длительность обозначим за PI . Длительность интервала планирования равна наименьшему общему кратному периодов всех сообщений:

$$PI = \text{НОК} \{T_i\}$$

Следовательно, достаточно построить локальные расписания только на данном интервале времени.

Для упрощения задачи, интервал планирования делится на равные слоты. Слот – это интервал времени, в который может осуществляться отправка сообщения. В каждый слот с каждого порта каждого узла сети может быть отправлено не более одного сообщения. Моменты времени начала и окончания слота задаются через глобальное время и совпадают на каждом узле сети.

Длительность слота выбирается таким образом, чтобы время передачи сообщения с одного узла сети на другой не превышало время длительности слота. В случае необходимости передать большой объем данных, данные передаются несколькими сообщениями. Сквозная длительность передачи точка-точка в таком случае определяется как время с момента отправки первого сообщения с абонента-отправителя до приема последнего сообщения абонентом-получателем.

Необходимо построить маршруты передачи для каждого сообщения Msg_i , то есть найти упорядоченный набор

$$\begin{aligned} Route^i &= \{ R_1^i, R_2^i, \dots, R_{N_R^i}^i \}, |Route^i| = N_R^i, \text{ где } R_1^i = Src_{msg}, R_{N_R^i}^i \\ &= Dst_{msg}, \forall k \neq m R_m^i \neq R_k^i, (R_k^i, R_{k+1}^i) \in E \end{aligned}$$

и задать слот отправки каждого экземпляра сообщения Msg_i^n с каждого узла маршрута, то для каждой пары Msg_i^n, V_a определить значение

$\text{Time}(Msg_i^n, V_a) \rightarrow k \in \mathbb{N}$, где k – номер слота отправки, если $V \in Route^i$, иначе -1. Данное значение задает номер слота отправки экземпляра сообщения Msg_i^n с узла сети V_a .

Решение задачи корректно, если для любых экземпляров сообщений выполняются следующие ограничения:

- Ни на одном порту ни одного узла сети никакой паре различных сообщений не назначен один и тот же слот отправки:

$$\forall a, n \neq m, i \neq j \text{ Time}(Msg_i^n, V_a.p) \neq \text{Time}(Msg_j^m, V_a.p).$$

- Сообщение может передаваться по каналу в сети только после того, как оно было передано по предыдущему каналу на его пути:

$$\forall R_m^i, R_n^i, \forall m < n \rightarrow \text{Time}(Msg_i^k, R_m^i) < \text{Time}(Msg_i^k, R_n^i)$$

- С каждого следующего узла маршрута отправка происходит строго в следующий слот:

$$\forall n, i, a \text{ Time}(Msg_i^n, V_{a+1}) = \text{Time}(Msg_i^n, V_a) + 1$$

Данное ограничение не является обязательным для сетей с временной синхронизацией. Рассмотрение задачи без этого ограничения является предметом дальнейшего исследования.

- Сообщение должно быть доставлено абоненту-получателю за один период:

$$\text{Time}(Msg_i^k, Dst_{msg}) - \text{Time}(Msg_i^k, Src_{msg}) < T_{msg}$$

3. Методы планирования информационного обмена в сетях с временной синхронизацией

Сформулированную выше задачу можно решить разными методами. Эти методы делятся на совместные и последовательные.

Совместные методы строят маршруты и расписание одновременно. Методов совместного решения задачи построения маршрутов и расписания передачи данных обнаружено не было.

При последовательном решении задачи сначала строятся маршруты передачи данных, затем расписание передачи данных. Задача построения расписания для сетей с временной синхронизацией – это NP-полная задача выполнимости [3]. Существует множество методов построения корректного расписания:

- использование SMT-решателей (satisfiability modulo theories, SMT - задача выполнимости формул) [2] [3];
- генетические алгоритмы [5];
- целочисленное линейное программирование [6] [7];
- программирование в ограничениях [8].

Однако все эти методы используют уже готовые маршруты потоков данных. В случае, когда маршруты строятся отдельно от расписания, возможна ситуация, когда на один физический канал назначено много потоков данных, не сочетаемых по периодам. Это приводит к невозможности построить корректное расписание.

Таким образом, актуально исследование метода совместного построения маршрутов передачи сообщений и расписания их выдачи, с учетом влияния маршрутов на построение расписания.

4. Алгоритм совместного построения маршрутов передачи данных и расписания обменов

Алгоритм основан на жадных стратегиях [9] и стратегиях ограниченного перебора.

Жадная стратегия обработки сообщений выбрана по аналогии со схемой планирования вычислений Rate Monotonic, в которой в первую очередь планируются задачи с минимальным периодом.

В алгоритме используется понятие **априорной загрузки** однонаправленного канала. Она необходима для определения потенциальных узких мест в сети на этапе, когда неизвестны маршруты передачи данных. Априорная загрузка канала вычисляется как потенциальный объем проходящих через него данных. Для этого рассматривается суммарный объем данных, передающийся за интервал планирования, всех потоков данных, которые потенциально могут проходить через него. При этом учитывается каждый маршрут, в котором содержится этот канал. Предположим, что число существующих маршрутов n . Далее для каждой пары абонентов суммируется весь объем пересылаемых данных за интервал планирования – W . Считается, что по каждому маршруту проходит объем данных W/n . После чего на каждом канале маршрута увеличиваем априорную загрузку на W/n . Таким образом, определяется априорная загрузка для каждого канала сети, позволяя вычислять потенциально перегруженные каналы.

Длина маршрута L определяется как максимальная априорная нагрузка среди всех каналов, входящих в маршрут.

Вес канала определяется как сумма частот уже размещенных в ходе работы алгоритма сообщений, передающихся по данному каналу.

Общая схема алгоритма следующая:

Шаг 1. Упорядочить сообщения Msg_i по возрастанию периода отправления сообщений T_i . Алгоритм осуществляет последовательную обработку сообщений, поэтому начинать необходимо с наиболее часто передающихся сообщений, так как редко передающиеся сообщения легче добавить в расписание.

Шаг 2. Для каждого сообщения строится набор допустимых маршрутов $Route^i$ методом k-кратчайших маршрутов по метрике количество каналов в маршруте [10], причем количество маршрутов не должно превышать заранее выбранное число $MaxRoute$. Значение параметра $MaxRoute$ выбирается исходя из сложности алгоритма.

Шаг 3. Для маршрутов вычисляется их длина $L(Route^i)$, после чего они упорядочиваются по возрастанию длины.

Шаг 4. Для каждого сообщения выполняются шаги 4.1, 4.2.

Шаг 4.1. Для каждого маршрута из набора допустимых для данного сообщения осуществляется построение расписания, пока не будет найден маршрут, для которого оно завершится успешно. Построение расписания для маршрутов осуществляется по следующей схеме: на маршруте выбирается канал $E_i = (V_a, V_b)$ с наибольшим весом, для него запускается построение локального расписания. Эту операцию осуществляет процедура «Построение локального расписания».

Шаг 4.2. Если ни для одного маршрута построение локального расписания не завершилось успехом, СТОП: построить расписание не удалось.

Шаг 5. Если для каждого сообщения был найден маршрут, на котором успешно было построено локальное расписание, то СТОП: успешное завершение алгоритма.

Построение локального расписания.

Шаг Л1. Формируется множество номеров допустимых первых слотов $First = \{x \in [0, T_i - 1], x\text{-свободный}\}$. Слот называется свободным, если нет никакого экземпляра никакого сообщения Msg_k^m , для которого назначена в этот слот отправка с этого порта данного узла сети. Множество упорядочено по возрастанию номера слота.

Шаг Л2. Для каждого слота из множества допустимых слотов проверяются слоты $(First[j] + m * T_i)$, где m - целое неотрицательное число. Если все такие слоты свободны, то локальное расписание считается построенным и функция задается как $Time(Msg_i^m, V_a) = First[j] + m * T_i$. В случае успешного построения расписания на данном канале рассматриваются каналы справа и слева по маршруту.

Шаг Л3. Для этих каналов запускается построение локального расписания, где в качестве начального слота выбирается слот $First[j] - 1$, если рассматривается канал, предшествующий рассматриваемому, и $First[j] + 1$ иначе. В случае неуспешного построения алгоритм переходит на Шаг 5 и строится расписание для следующего допустимого маршрута.

Примечание: При выполнении шага 4.1. для очередного сообщения построение расписания производится с учетом загрузки каналов расписаниями, построенными для ранее обработанных маршрутов.

Сложность данного алгоритма

$$O(N_{Msg} * \log(N_{Msg})) + O(MaxRoute * \log(MaxRoute)) + MaxRoute * T_{max} * O(MaxLength * PI),$$

где

- $N_{Msg} = |Msg|$ - количество сообщений;
- $MaxLength$ - максимальная длина маршрута;
- T_{max} - максимальный период сообщения;
- PI - длина интервала планирования.

5. Экспериментальное исследование построенного алгоритма

5.1. Цель исследования

Экспериментальное исследование проводилось с целью сравнить результаты построения расписания совместно с построением маршрутов и последовательно.

Для проведения исследования было выбрано несколько конфигураций системы. Каждая конфигурация содержит информацию о структуре сети и о передающихся сообщениях.

Для каждой конфигурации строятся маршруты и расписание передачи данных двумя методами. В одном случае используется разработанное средство. Во втором - сначала строятся маршруты передачи данных, а затем расписание обменов.

Для построения маршрутов во втором случае используется САПР AFDX [11]. Эта система автоматизированного проектирования позволяет строить маршруты передачи сообщений по сети, накладывая единственное ограничение при построении, которое звучит следующим образом: «Суммарная требуемая пропускная способность на физических каналах передачи данных не должна превышать максимальную доступную пропускную способность». Так как для исследования не требуется накладывать дополнительные ограничения на маршруты, данное средство подходит для решаемой задачи.

Расписание для построенных маршрутов строится с помощью SMT-решателя, так как это наиболее широко используемый метод построения расписания для уже построенных маршрутов. Также к

достоинствам этого метода можно отнести детерминированность, которой отдается предпочтение при проектировании бортовых систем. В данной работе был использован SMT-решатель JavaSMT [12].

Два указанных выше подхода (совместное и раздельное построение маршрутов и расписания) сравниваются по критерию успешности построения расписания.

5.2. Исходные данные

Наборы исходных данных состоят из описания структуры сети и описания рабочей нагрузки. Рабочая нагрузка включает в себя описание структуры сети и описание набора сообщений.

Так как бортовые сети являются сетями небольшого размера, для экспериментальных исследований были выбраны структуры сети, представленные ниже на рисунках 1-2. Кругами обозначены коммутаторы, в прямоугольниках – абоненты.

В конфигурациях используется небольшое количество сообщений, однако получатели и отправители были выбраны таким образом, чтобы возникало как можно больше маршрутов с потенциальными конфликтами, то есть маршрутов, при выборе которых построение расписания завершится с ошибкой.

В первой конфигурации используется с сети, представленная на рис.1. Данные о сообщениях представлены в таблице 1.1. *MaxRoute* равен 4.

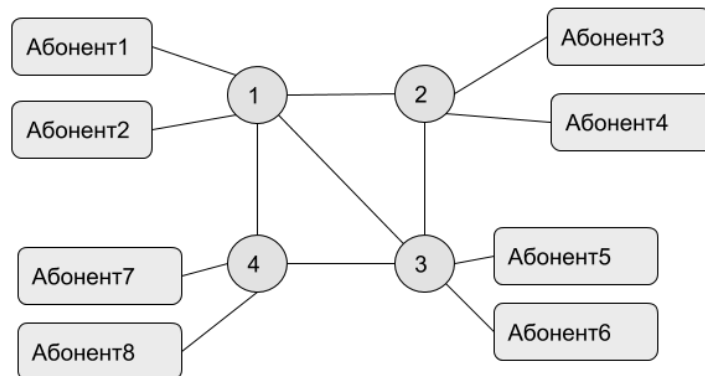


Рисунок 1. Структура сети №1

Во второй конфигурации используется структура сети, представленная на рис.1. Данные о сообщениях представлены в таблице 1.2. *MaxRoute* равен 4.

Отправитель	Получатель	Период
A3	A1	2
A4	A2	3
A5	A2	3
A6	A2	3

Таблица 1.1: Данные о сообщениях 1.

Отправитель	Получатель	Период
A3	A1	2
A4	A1	4
A5	A2	3
A6	A2	3

Таблица 1.2: Данные о сообщениях 2.

В третьей конфигурации используется структура сети, представленная на рис.2. Данные о сообщениях представлены в таблице 1.3. *MaxRoute* равен 4.

Отправитель	Получатель	Период
A1	A3	2
A2	A6	3
A5	A4	3
A7	A4	4
A8	A4	4

Таблица 1.3: Данные о сообщениях 3.

В четвертой конфигурации используется структура сети, представленная на рис.2. Зависимость по данным задана в таблице 1.4. *MaxRoute* равен 4.

Отправитель	Получатель	Период
A3	A7	3
A1	A7	3
A5	A7	3
A2	A8	4
A4	A8	4
A6	A8	4

Таблица 1.4: Данные о сообщениях 4.

Пятая конфигурация по структуре сети и зависимостям по данным идентична конфигурации 4 за исключением выбора параметра *MaxRoute*. Здесь этот параметр принимал значение 1.

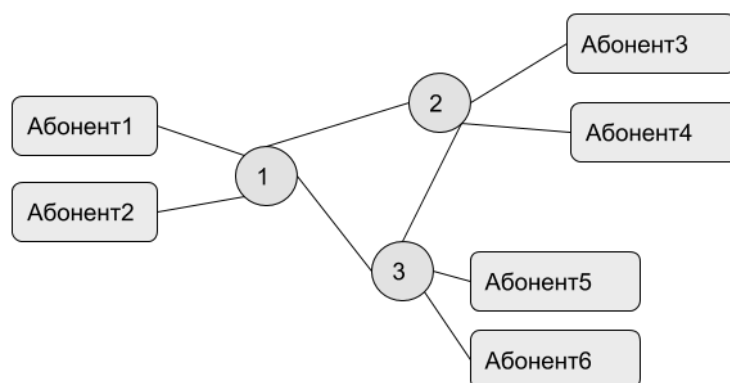


Рисунок 2. Структура сети №2

5.3. Результаты экспериментов

На таблице 1.5 представлены результаты работы последовательного построения и алгоритма для каждой конфигурации системы.

5.4. Выводы

Экспериментальное исследование показало, что использование маршрутов, построенных совместно с расписанием обменов, позволяет в большем количестве конфигураций успешно построить расписание. Однако важно правильно задавать параметр *MaxRoute*, так как чем меньше значение этого параметра, тем меньше рассматривается маршрутов, тем меньше вероятность построить корректное расписание, однако большое значение этого параметра увеличивает сложность алгоритма.

Номер кон- фигурации	Успех сов- местного построения	Успех после- довательного построения
1	+	-
2	+	+
3	+	-
4	+	+
5	-	+

Таблица 1.5: Результаты экспериментального исследования.

В данной серии экспериментов можно сделать вывод, что отсутствие перебора маршрутов ($MaxRoute = 1$) снижает возможности алгоритма по построению расписаний.

Подход к выбору значения $MaxRoute$ в зависимости от характеристик входных данных может быть предметом дальнейшего исследования.

6. Заключение

Сети Ethernet с временной синхронизацией позволяют обеспечить высокий временной детерминизм обмена данными в информационно-управляющих системах реального времени. При планировании обмена по таким сетям необходимо построить маршруты передачи сообщений и сформировать расписания выдачи сообщений для всех узлов сети, включая абонентов и коммутаторы. В существующих методах планирования обмена для сетей рассматриваемого класса вначале строятся маршруты передачи сообщений, затем для полученного набора маршрутов строятся расписания выдачи сообщений.

Подобное разделение этапов приводит в ряде случаев к невозможности построения расписаний в связи с перегрузкой отдельных участков сети.

Предложенный в данной работе подход совмещает построение маршрутов и расписаний выдачи сообщений, что позволяет учитывать загрузку сетевых каналов при построении маршрутов. Экспериментальное исследование показало преимущество предложенного подхода над отдельным выполнением этапов планирования. К направлениям дальнейшего исследования можно отнести модификацию разработанного алгоритма с тем, чтобы при неуспешном построении решения выполнялось достраивание маршрутов и расписаний

с частичным нарушением ограничений; для построенных таким образом маршрутов будет применяться известный метод построения расписания при помощи SMT-решателя.

Литература

1. Obermaisser R. *Time-triggered communication*. – CRC Press, 2011.
2. Steiner W. *An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks*. // Real-Time Systems Symposium (RTSS), 2010 IEEE 31st. – IEEE, 2010. – P. 375-384.
3. Steiner W. *TTEthernet dataflow concept* // Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on. – IEEE, 2009. – P. 319-322.
4. AFDX / ARINC 664 Tutorial (1500-049) // Condor Engineering, Inc. – 2005.
5. Monnier Y., Beauvais J.P., Deplanche A.M. *A genetic algorithm for scheduling tasks in a real-time distributed system* // Euromicro Conference, 1998. Proceedings. 24th. – IEEE, 1998. – vol. 2. – P. 708-714.
6. Jones J.G., Meneghin B.J., Kirby M. W. *Formulating adjacency constraints in linear optimization models for scheduling projects in tactical planning* // Forest science. – 1991. – vol. 37. – No. 5. – P. 1283-1297.
7. Goswami D. *Time-triggered implementations of mixed-criticality automotive software* – EDA Consortium, 2012. – P. 1227-1232.
8. Marriott K., Stuckey P.J. *Programming with constraints: an introduction*. – MIT press, 1998.
9. Вдовин П.М. *Жадные алгоритмы и стратегии ограниченного перебора для планирования вычислений в системах с жесткими требованиями к качеству обслуживания* : дис. ... канд. физ.-мат. наук: 05.13.11. – М., 2016. – 127 с.
10. Yen J. Y. Finding the k shortest loopless paths in a network // Management Science. – 1971. – vol. 17. – № 11. – P. 712-716.

11. Вдовин П.М. *Инструментальная система проектирования сетей AFDX* // труды МФТИ. 2015. Т. 7. No 2. С.131 – 136.
12. JavaSMT [HTML] <https://github.com/sosy-lab/java-smt>