

УДК 621.391

СЕТИ ETHERNET С ДЕТЕРМИНИРОВАННЫМИ ЗАДЕРЖКАМИ***Б.Я. Лихтциндер***¹

Поволжский государственный университет телекоммуникаций и информатики
Россия, 443100, г. Самара, ул. Льва Толстого, 23

E-mail: lixt@psuti.ru

Аннотация. Работа посвящена новой технологии, обеспечивающей получение малых и предсказуемых задержек передачи данных в промышленных сетях Ethernet. Проводится разделение трафика на критичный и некритичный к задержкам. Рассмотрены особенности функционирования TSN. Рассмотрены принцип действия и архитектура таких сетей. Показано, что снижение затрат может быть достигнуто за счет наличия только одной сети, которая отвечает требованиям детерминированного сетевого взаимодействия и совместима с современными решениями Ethernet. Приведены примеры и показано, что детерминированные сети могут быть использованы для создания систем жесткого реального времени. Особенно это относится к автоматическим системам регулирования и управления, включающим цепи обратной связи. Ключевые особенности TSN, обеспечивающие гарантированные сроки доставки сообщения, – это синхронизация времени передачи трафика и планирование. Коммутатор TSN представлен в виде двух подсистем, одна из которых выполняет стандартные функции Ethernet, а вторая обеспечивает логику управления TSN. Рассмотрен пример циклического управления трафиком по расписанию. Показана перспективность использования технологии TSN в сетях управления промышленной автоматикой.

Ключевые слова: трафик, детерминированные задержки, управление, коммутаторы, сети, цикличность

Введение

Сегодня мир переживает новый подъем цифровой трансформации, которая требует от производителей переосмысления существующих моделей промышленной автоматизации. Цифровая трансформация открывает новые возможности для автоматизации и производства. Индустрия 4.0, как еще называют эту новую волну цифровой трансформации, коренным образом меняет мировую экономику.

По мере усложнения систем они становятся все более распределенными, и объединяют их телекоммуникационные сети. Широкое распространение и положительные качества сетей Ethernet сделали их чрезвычайно привлекательными при организации связи между различными устройствами промышленной автоматизации. Однако, несмотря на высокую производительность, присущий им метод случайного доступа к общим сетевым ресурсам не

¹ Лихтциндер Борис Яковлевич, профессор кафедры «Мультисервисные сети и информационная безопасность», доктор технических наук, профессор.

обеспечивает гарантированных длительностей сетевых задержек, что для ряда систем управления технологическими процессами совершенно недопустимо.

В заводской среде существуют обычные ПК, на которых сотрудники выполняют свою работу, а также оборудование, управляемое компьютерами. Сотрудникам необходим доступ к локальной сети для работы в Интернете и интрасети. Оборудование использует сеть для связи с компьютером, управляющим им. Необходимость в подключенном оборудовании и обычных ПК может сильно различаться. Машины могут быть критически важны с точки зрения безопасности, в то время как сотрудникам требуется обычное подключение к локальной сети и для них важна в основном пропускная способность. Эти примеры использования требуют лучших решений, чем те, которые имеются на сегодняшний день. Снижение затрат может быть достигнуто за счет наличия только одной сети, которая отвечает требованиям детерминированного сетевого взаимодействия и совместима с современными решениями.

Локальные сети обычно использовались для подключения компьютеров к Интернету или внутри сети компании для подключения компьютеров друг к другу и к серверам. Поэтому возникла необходимость в отдельной сети для удовлетворения требований к детерминистской связи, например на заводах в автомобильной промышленности. Детерминизм может быть определен следующим образом: «Физическая система ведет себя детерминированно, если, учитывая начальное состояние в момент t и набор будущих синхронизированных входов, вытекают будущие состояния, а также значения и время будущих выходов» [1]. Это означает, что поведение сети можно предсказать. Детерминированные сети предсказуемы. Для предсказуемости необходимо, чтобы данные каждый раз проходили через сеть так, как ожидалось. Например, если два реплицированных компонента имеют одинаковое начальное состояние, они завершатся примерно в одно и то же время [1]. Для достижения этой цели в сети не должно быть трафика, блокирующего критичные по времени данные. Для удовлетворения этих требований необходимо зарезервировать некоторую полосу пропускания. Это означает нулевые потери при перегрузке и гарантированную задержку [2]. Благодаря гарантиям и тому, что детерминированная сеть предсказуема, потеря пакета расценивается как отказ оборудования. Детерминированные сети могут быть использованы для создания систем жесткого реального времени.

Чем больше возможностей требуется от системы, тем большее число компонентов она должна включать, и все они должны работать синхронно друг с другом, особенно в системах, критичных ко времени задержки, иначе вся система не будет работать. Особенно это относится к автоматическим системам регулирования и управления, включающим цепи обратной связи. Хорошо известно, к каким пагубным последствиям приводит наличие запаздывания в цепях обратной связи таких систем. Они не только теряют качество и точность, но при определенных условиях могут потерять и устойчивость процесса управления. Это в полной мере, например, относится к автоматизации управления транспортом.

Уровень безопасности программного обеспечения автомобиля определяется тем, насколько неуправляемым будет автомобиль, если данное программное обеспечение выйдет из строя [3].

Автономные транспортные средства, которые полагаются на быструю обработку данных от множества различных датчиков (оптических, LIDAR, приближения, GPS, инерционных и т. д.), могут иметь несколько различных устройств, обрабатывающих эти данные, соединенных сетью связи. В какой-то момент проанализированные данные от различных датчиков объединяются для принятия решений (объединение данных).

Объединение данных требует, чтобы все данные либо были одинакового времени, либо имели известный момент появления. Это также требует точной синхронизации времени между всеми устройствами сбора данных. Большинство распределенных систем, критичных ко времени, нуждаются в той или иной форме синхронизации времени. Синхронизация позволяет детерминировать возникающие задержки. Появилось множество протоколов различных производителей, направленных на обеспечение гарантированных сетевых задержек. Это привело к невозможности совместной работы оборудования различных производителей или к неоправданно большим затратам на приобретение специальных шлюзов, обеспечивающих такую совместную работу. Ситуация изменилась с появлением сетей с управляемыми задержками (Tim-Sensitive Nets – сети, чувствительные к задержкам), которые все более активно начинают внедряться в промышленную автоматизацию. Эти сети, совместимые с Ethernet, способны передавать высокоприоритетные пакеты с минимально возможными задержками. Они совместили в себе все положительные свойства сетей Ethernet с возможностью получения детерминированной задержки сетей TDM, предоставляя абсолютный приоритет пакетам, наиболее чувствительным к задержкам. Появился ряд стандартов, дополняющих существующие стандарты Ethernet, в которых предусматривается получение детерминированных задержек в сетях Ethernet за счет жесткой синхронизации по времени и управления потоками трафика на основе расписания. Переключение потоков в TSN осуществляется за счет введения в коммутаторы Ethernet дополнительных аппаратных средств в виде программируемых логических матриц (ПЛМ). Необходимость получения детерминированных задержек привела также к созданию подобных беспроводных сетей.

1. Технология TSN

1.1. Гетерогенные сети

При создании сети или проектировании системы, использующей сеть, сеть должна быть выбрана в соответствии с требованиями к ней. Однако во многих случаях требуется наличие сети Ethernet, поскольку она используется для соединения компьютеров. Другие компоненты, например оборудование на заводе, могут иметь более строгие требования, и это означает необходимость в двух сетях. Как показано на рис. 1а, сеть содержит только ПК и является однородной. Сеть на рис. 1б – это сеть управления, где один компьютер управляет четырьмя промышленными роботами. Сеть должна быть детерминированной, например из соображений безопасности. Проще и дешевле иметь только одну сеть, поэтому возникает необходимость в гетерогенных сетях (разнородных). Сеть должна иметь протокол, отвечающий требованиям всех устройств, показанных на рис. 1б. Даже если в сети примера есть только один тип устройства – промышленный робот, которому требуется детерминированная сеть, вся сеть должна быть детерминированной.

Ранние сети стандарта Ethernet не могли гарантировать доставку данных

и имели значительные задержки. В результате в отраслях, требующих высокой надежности и доступности, были разработаны собственные специализированные сетевые решения (например, модифицированные сети Ethernet и полевые шины) для промышленных систем управления и автоматизации. Чтобы удовлетворить требования промышленных приложений к высокой доступности и малой задержке, традиционные технологии Ethernet с негарантированной доставкой должны были развиваться и становиться более детерминированными. TSN – это, по сути, следующий этап эволюции стандартных технологий Ethernet, отвечающий перспективным требованиям.

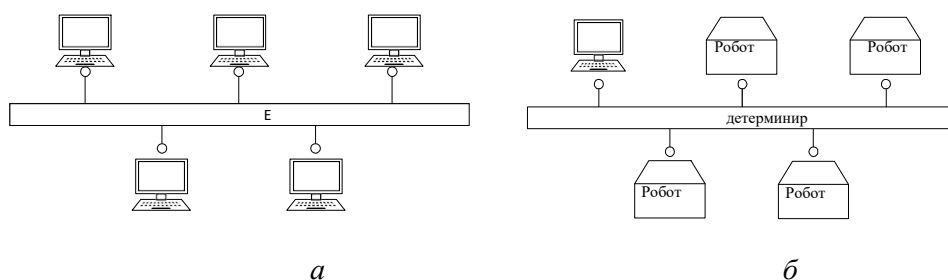


Рис. 1. Две разные сети, необходимые в заводской среде:
а – сеть Ethernet; *б* – детерминированная сеть

1.2. Создание TSN

TSN представляет собой набор стандартов, обеспечивающих детерминированный обмен сообщениями по стандартным сетям Ethernet. Согласно определению Института инженеров по электротехнике и радиоэлектронике (IEEE), TSN включает в себя форму управления сетевым трафиком, которая гарантирует строго определенное значение задержек при сквозной передаче данных. Следовательно, все устройства в сетях TSN должны синхронизироваться друг с другом и использовать единый сервер времени для обеспечения обмена данными в режиме реального времени в промышленных системах. Хотя стандарты TSN изначально разработаны IEEE, важно понимать, что TSN выходит за рамки основных стандартов IEEE и включает в себя упорный совместный труд многих международных организаций и компаний.

Ethernet как технология так же широко используется в промышленности, как и в домашних хозяйствах. Впервые он был представлен в 1980 г., а после его первой стандартизации в 1983 г. рабочей группой Института инженеров по электротехнике и электронике (IEEE) он почти заменил все другие технологии проводных локальных сетей (LAN). Начальная скорость, которую он мог достичь, составляла 10 Мбит/с. С тех пор было внесено много улучшений и введены новые стандарты. В 2017 г. скорость Ethernet достигла 400 Гбит/с. Набор всех этих стандартов, которые определяют Ethernet, – это IEEE 802.3. Они определяют физический уровень и уровни канала передачи данных Media Access Control (MAC) проводной сети. Эти два уровня также известны как первые два уровня модели взаимодействия открытых систем (OSI). IEEE 802.3 – это рабочая группа проекта IEEE 802, и все стандарты, определенные в этом проекте, относятся к локальным сетям (LAN) и городским сетям (MAN). Рабочая группа IEEE 802.1 определяет мосты LAN/MAN и управление ими. Частью этой группы является рабочая группа Time-Sensitive Networking (TSN) [4] – это бывшая

рабочая группа IEEE, и чтобы расширить работу, эта группа задач была переименована в Time-Sensitive Networking Task Group в 2012 г.

Ключевые особенности TSN, обеспечивающие гарантированные сроки доставки сообщения, – это синхронизация времени и трафика и планирование. Они определены стандартами 802.1AS и 802.1Qbv соответственно. Все устройства, участвующие в сети TSN, синхронизированы с глобальным временем и знают о сетевом расписании, которое определяет, как расставлены приоритеты сообщений при пересылке с каждого коммутатора.

Стандарт IEEE 802.1, как это показано на рис. 2, использует несколько очередей на выходе каждого порта коммутатора. Там сообщения хранятся до тех пор, пока выходы не откроются (во временном интервале) и не пропустят сообщения из очереди для передачи инфекции. Своевременный выпуск сообщений гарантирует, что задержки в сети можно будет детерминистически предсказать. Это позволяет объединить в одной той же сети критически важный и некритический трафики.

Этот стандарт помещает сообщения в разные классы в зависимости от их приоритета и использует формователи трафика для прогнозирования и предотвращения перегрузки портов коммутатора. Чтобы обеспечить априорную передачу сообщений и прерывание некритических сообщений, были внесены две поправки: 802.1Qbv и 802.1Qbu. Первая поправка вводит запланированный трафик, а вторая вводит приоритет кадра. Эти две поправки являются ключевыми средствами обеспечения связи в реальном времени в сетях TSN. Стандарты IEEE 802.1Qbv и IEEE 802.1QAS использовались для обеспечения запланированной синхронизации трафика и времени соответственно. Итак, технология Time-Sensitive Networking (TSN) решает проблему отсутствия поддержки систем реального времени в локальных сетях. Она является расширением стандарта IEEE 802.1Q и известна как 802.1Qbv. TSN создана для работы, когда трафик содержит до 75 % данных, критичных ко времени, а остальные являются best-effort [5]. Следовательно, она позволяет одной сети заменить несколько сетей с различными требованиями и таким образом обеспечить более простые сетевые решения и сократить расходы.

1.3. Принцип действия TSN

Для того чтобы передача данных могла быть приоритетной, должны существовать определенные правила. Для 802.1Q правила пиритизации определяются типами трафика и назначением данных. Имеются различные приоритеты и типы трафика, которому назначается приоритет. Существует максимум восемь приоритетов, поскольку метка, определяющая приоритет, содержит 3 бита. Все преобразования трафика в TSN производятся в выходных портах коммутаторов.

На рис. 2 показана модель со всеми различными частями исходящего порта.

Вводимые данные поступают и помещаются в исходящий порт. Данные проходят через блок выбора очереди (a) и помещаются в нужную очередь (b). Алгоритм выбора передачи (TSA) выбирает (c), какие данные взять из очереди, и если следующий шлюз Gate (e) открыт, то они могут быть переданы. Открытые шлюзы определяются списком управления шлюзами (d). Выбор передачи TS (f) выбирает, какие данные должны быть переданы с выхода данных (g). Как показано на рис. 2, существует несколько очередей для обслуживания различных классов трафика. Таким образом, всего существует максимум 8 очередей, по

одной для каждого класса трафика. Если очередей меньше восьми, каждая очередь может содержать данные от нескольких типов классов трафика. TSA решает, в каком порядке будут передаваться данные из очереди. Даже если TSA выбрала некоторые данные для дальнейшей передачи, шлюзы, следующие за ними, могут заблокировать передачу. Благодаря этому можно отдать приоритет не самому приоритетному типу трафика, а неким другим данным. *Шлюзы являются краеугольным камнем TSN.* Каждый шлюз может находиться в открытом или закрытом состоянии. Управление ими осуществляется с помощью списка управления шлюзами. Операции в списке управления шлюзами заключаются в открытии или закрытии определенных шлюзов. Селектор передачи принимает кадр с наивысшим приоритетом, который может быть выбран для передачи [6]. Если шлюз перед селектором передачи закрыт, то он будет действовать так, как будто данные для передачи отсутствуют.

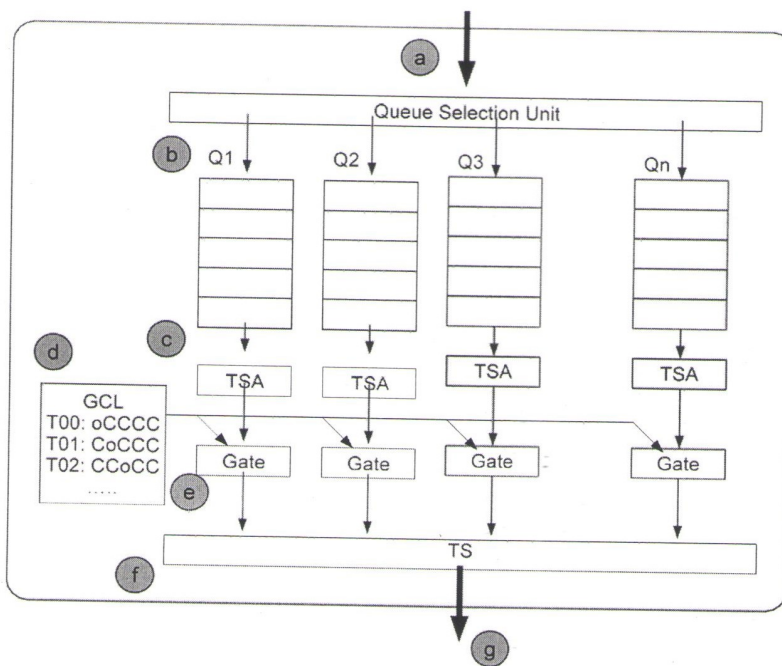


Рис. 2. Модель с различными частями исходящего порта

Передача трафика производится в соответствии с различными алгоритмами. Дисциплина обслуживания передачи может быть не простой «первым пришел – первым ушел» (FIFO). Применяются алгоритмы с относительными и абсолютными приоритетами. Особое значение приобретает возможность предварительного резервирования временных окон передачи на основе расписания для циклического трафика и резервирования на основе упреждения. Дисциплина обслуживания может быть определена производителем.

Особое значение при реализации TSN приобретают вопросы синхронизации. Синхронизация позволяет произвести планирование открытия проходов таким образом, чтобы создавались открытые соединения с минимальными задержками прохождения трафика по всему пути от источника к получателю.

1.4. Архитектура TSN

Архитектура управления средствами промышленной автоматизации является иерархической (рис. 3).

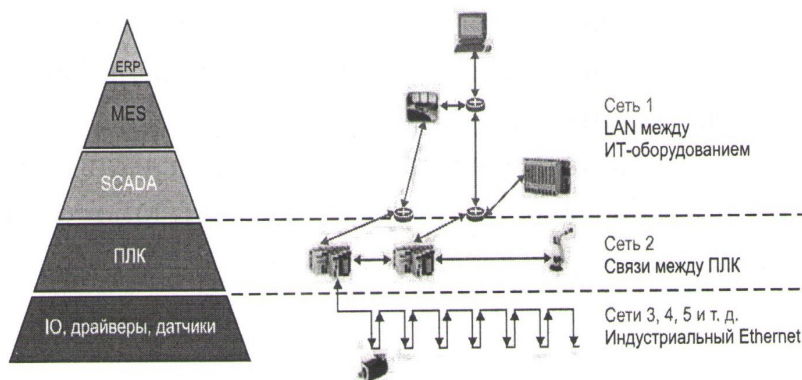


Рис. 3. Архитектура управления средствами промышленной автоматизации

Приложения, ответственные за планирование ресурсов предприятия ERP (Enterprise Resource Planning), находятся на самом верху иерархии и обеспечивают общее управление и планирование работы бизнес-процесса, вплоть до автоматизированных систем управления технологическими процессами (MES – Manufacturing Execution System), которые контролируют непосредственно производственный процесс [7]. Непосредственный производственный процесс управляется специализированными контроллерами, получающими информацию от различных датчиков и управляющими различными исполнительными механизмами, находящимися на самом нижнем уровне иерархии. Различные уровни соединяются телекоммуникационными сетями и предъявляют к ним различные требования. На высоких уровнях требуется большая пропускная способность, в то время как нижние уровни нуждаются в детерминированном поведении и передаче данных с малой вариацией временной задержки. Технология TSN позволяет объединить указанные разрозненные системы. Перед разработчиками TSN встала задача сохранить все преимущества сетей Ethernet, но обеспечить возможность гарантированной передачи данных в течение заданного промежутка времени. Поэтому любая система, применяемая в TSN, может быть представлена в виде объединения двух подсистем (рис. 4). На первой подсистеме показаны входной и выходной порты, подуровень MAC и система коммутации, предусмотренные обычными стандартами Ethernet. Реализация функций TSN предусматривает весьма жесткую синхронизацию всех узлов сети. Поэтому предусмотрено программное обеспечение, обеспечивающее синхронизацию, в соответствии со стандартом IEEE 802.1AS. Подсистема, обозначенная как TSN, предусматривает введение дополнительного программного обеспечения, часть из которого (CUC) доступна пользователю и позволяет настроить работу системы в соответствии с задачами технологического процесса. Часть программного обеспечения, недоступная пользователю, осуществляет сопряжение обеих подсистем. Реализуется управление, как правило, с использованием комбинированного процессора и программируемой логики архитектуры FPGA (Field Programmable Gate Array), которые подключены к высокоскоростному каналу. Фактически сеть состоит из двух различных узлов – сетевых коммутаторов (NS) и конечных

систем (ES). На рис. 4 показан узел типа NS. Конечные системы либо отправляют, либо получают данные, но не передают их. Сетевые коммутаторы только передают данные, но не потребляют, не создают и не изменяют их.

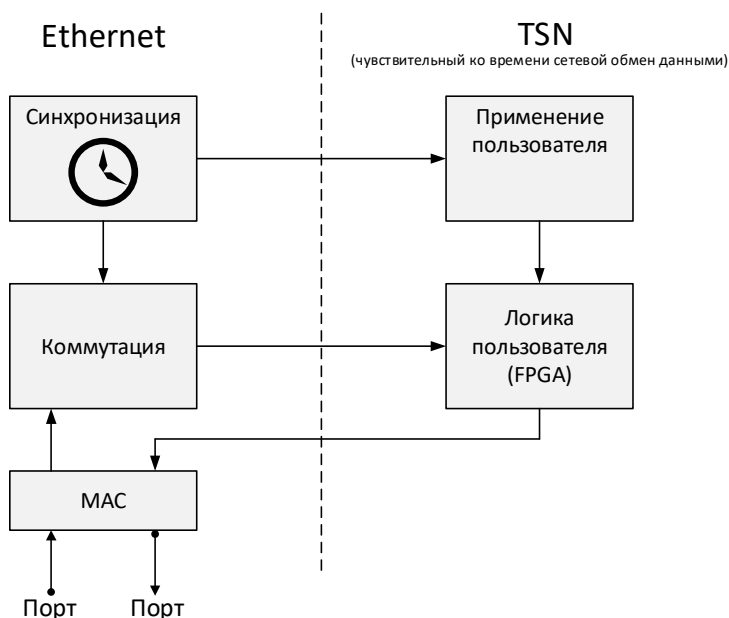


Рис. 4. Подсистемы TSN

Соединение с другими узлами осуществляется через исходящий порт. Каждый исходящий порт имеет свои собственные очереди, и каждый порт подключен только к одному другому узлу. Поэтому в каждом узле может быть более одного исходящего порта. Количество принимающих и передающих портов может отличаться. Одной из важных частей TSN являются часы внутри узла.

2. Планирование графика

Вопросам получения трафика с детерминированными задержками посвящено значительное количество зарубежных публикаций, краткий обзор которых приведен в [11].

Планирование критического трафика в детерминированных сетях с использованием программируемых систем было предложено Штайнером в [12]. Работы [13, 14] включали планирование приоритетных задач, выполняемых на узлах конечных систем.

Планирование передачи ограниченного по скорости трафика с разными классами критичности в TTEthernet при оптимизации сквозной задержки изучалась в [15]. Задачи планирования с оптимизацией и без нее для других технологий, таких как PROFINET, FlexRa9 и TTP, были рассмотрены в [16, 17, 18, 19]. Методы определения наихудшего случая сквозных задержек для незапланированных сетей (например, AFDX или AVB) были разработаны, например, в [20–22].

Асинхронный класс трафика введен в [23] в контексте TSN, где цель состоит в том, чтобы предложить возможности композиционного анализа связи, поддерживая схему сквозного прохождения трафика между выходными портами

по маршруту. Этот подход рассматривает поведение стандартных потоков AVB в предположении, что все синхронизированные ворота очередей выходных портов открыты одновременно. Существует значительное число промышленных устройств и технологических процессов, работающих в циклических режимах, когда информационные потоки возникают в определенных местах и должны доставляться получателям в строго регламентированные промежутки времени. В работе [24] вводится класс трафика с гарантией доставки в реальном времени, называемый «планируемый (Scheduled) график (ST)», который еще называют «трафиком высокой критичности». Технология TSN позволяет создать сквозной открытый путь для передачи таких потоков, обеспечив исключение влияния на их задержку со стороны других, менее срочных потоков.

Такой трафик возникает в сетях, объединяющих циклически работающие устройства и системы. Весь цикл работы таких устройств разделяется на равные между собой слоты, измеряемые временными тактами (тиками). Целью планирования является определение временных слотов, в течение которых указанный трафик должен проходить через выходные порты коммутаторов, и, следовательно, для него должны быть открыты сквозные проходы от источника возникновения трафика до всех его получателей.

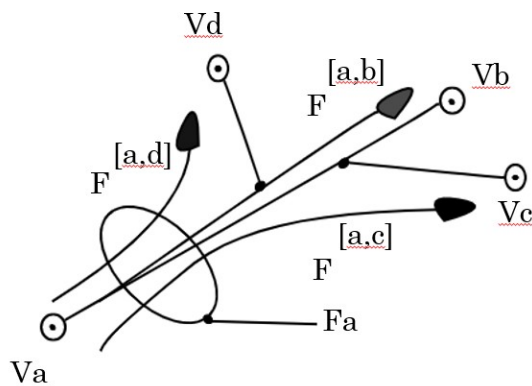


Рис. 5. Фрагмент потоков в сети

Параметры сети

На рис. 5 изображен фрагмент сети, где все конечные узлы рассматриваются как вершины графа и обозначаются как v_a, v_b, v_c, v_d .

Рассматриваемая сеть работает по полнодуплексным многоскоростным физическим каналам и моделируется как направленный граф $G(V, L)$, где узлами (коммутаторами и конечными системами) является множество вершин графа (V), связи между узлами являются двунаправленными и представляют ребра графа $L \subset (V \times V)$. Вершины графа соответствуют следующим устройствам: V_e – конечные системы; V_s – коммутаторы; V_{e+s} – все устройства.

Каналы, представляемые направленными ребрами, обозначаются как $[v_a, v_b] \subset L, [v_b, v_a] \subset L$. Физическая связь между узлами v_a и v_b приводит к двум направлениям логических каналов, каждый из которых обозначается упорядоченным кортежем, а именно $[V_a, V_b]$ и $[V_b, V_a]$ соответственно.

Путь от отправителя v_a к получателю v_b , маршрутизируемый через промежуточные узлы $v_1, v_2 \dots v_n$, обозначим через кортеж $s_i = [[v_a, v_1], [v_1, v_2], \dots, v_n, v_b]$.

Физическая связь $[v_a, v_b]$ определится набором величин $\langle [v_a, v_b]..s, [v_a, v_b]..d, [v_a, v_b]..mt, [v_a, v_b]..c \rangle$, где $[v_a, v_b]..s$ – скорость в канале, $[v_a, v_b]..d$ – объявленная задержка, $[v_a, v_b]..mt$ – слот времени (макротик), $[v_a, v_b]..c$ – количество доступных очередей в канале (в порту).

Параметры трафика

Планируемый трафик возникает в источнике v_a и передается в конечные вершины v_b, v_c, v_d . Весь поток трафика, исходящего из вершины v_a , обозначим через F_a . Указанный поток представляет сумму составляющих потоков, поступающих от источника v_a ко всем получателям v_b, v_c, v_d .

Обозначим составляющие потоки через $F^{[a,b]}$, $F^{[a,c]}$, $F^{[a,d]}$ соответственно. Индекс $[a,b]$ обозначает канал, первая буква которого соответствует номеру выходного порта источника, а вторая – номеру выходного порта промежуточного коммутатора или конечного получателя. В этом разделе мы пользуемся обозначениями, принятыми в работе [1].

Потоки образуют в выходных портах коммутаторов очереди. Конфигурация очереди $G(Q)$ характеризуется следующими параметрами:

$$G(Q) = (N; N_u; N_{prio}),$$

где N – общее количество очередей на порте; N_u – количество очередей, работающих в запланированном отрезке времени ТТ; N_{prio} – количество оставшихся очередей с приоритетами (запланированные очереди всегда имеют наивысший приоритет).

Набор параметров, определяющих конфигурацию всей системы, определяется как $(G(Q); G(E))$.

Поскольку любой исходящий порт подключен только к одному звену, можно установить эквивалентность между портом и связанным с ним каналом. Следовательно, можно обращаться к переменной $[v_a, v_b]:c$ как к количеству исходящих очередей, присутствующих в соответствующем порту.

Определим время передачи кадра между двумя узлами через длину кадра и скорость соединения, которая представляет скорость передачи по физическому сетевому кабелю. Например, при передаче по каналу со скоростью 1 Гбит/с фрейма Ethernet IEEE 802.1Q размером 1542 байта время передачи будет $12,36 \mu$ sec.

Объявленная дополнительная задержка обусловлена задержкой распространения по физической среде, зависящей от расстояния.

Макротик определяет размер дискретной единицы времени, характеризующей степень детализации временных событий, которые протекают по данному каналу.

Несмотря на то, что поток F_a имеет несколько составляющих, без уменьшения общности можно рассмотреть лишь одну из его составляющих $F^{[a,b]}$, проходящих по каналу $[v_a, v_b]$.

В модели различают поток и экземпляр потока, проходящего через канал. Экземпляр потока, проходящего через канал $[v_a, v_b]$, обозначим через $s_i^{[v_a, v_b]}$.

Экземпляр $s_i \subset S$ потока, исходящего от узла v_a к получателю v_b , проходящий через промежуточные коммутаторы $v_a, v_1, v_2, \dots, v_{n-1}, v_n$, обозначим через $s_i = [[v_a, v_1], [v_1, v_2], \dots, [v_{n-1}, v_n], [v_n, v_b]]$.

Множество допустимых максимальных и минимальных сквозных задержек, длин кадров в байтах и периоды действия экземпляра s_i обозначим через $\langle s_i.e2e, s_i.L, s_i.T \rangle$ соответственно.

Переменную экземпляра очереди в порту i , соответствующую каналу $[v_a, v_b]$, обозначим через $s_i^{[v_a, v_b]}.p$. Наличие очереди является синонимом того, что в пределах данного выходного порта i соответствующего устройства имеется приоритетный поток.

Поскольку размер данных может превышать размер MTU – Ethernet, каждый экземпляр потока фреймов не должен превышать этот размер.

Каждый составляющий поток может состоять из нескольких потоков, имеющих наивысший приоритет. Обозначим через $F_i^{[a,b]} \subset F^{[a,b]}$ набор фреймов $f_{ij}^{[a,b]}$ из экземпляра потока $s_i^{[a,b]}$. Кроме того, первый и последний кадры этого набора, заказанные по расписанию, указанному по ссылке, обозначим как $f_{i1}^{[a,b]}$ и $last(F_i^{[a,b]})$ соответственно. Фреймы из набора $f_{ij}^{[a,b]} \subset F_i^{[a,b]}$ различают на

основании данных пула переменных $\langle f_{i,j}^{[v_a, v_b]}.p, f_{i,j}^{[v_a, v_b]}.T, f_{i,j}^{[v_a, v_b]}.L \rangle$, где

$f_{i,j}^{[v_a, v_b]}.p \in [0, f_{i,j}^{[v_a, v_b]}.T]$ – установлен в макротиках длины кадра в канале,

$f_{i,j}^{[v_a, v_b]}.T = \left[\frac{s_i.T}{[v_a, v_b].mt} \right]$ – период времени нахождения потока в канале $[V_a, V_b]$,

$f_{i,j}^{[v_a, v_b]}.L = \left[\frac{L_i[v_a, v_b, s]}{[v_a, v_b].mt} \right]$ – продолжительность передачи кадра в макротиках

в канале $[V_a, V_b]$.

Принятые обозначения могут быть использованы при моделировании и планировании трафика TSN.

3. Циклическое обслуживание трафика по расписанию

Многие процессы, требующие передачи информации, носят циклический характер. Моменты появления кадров, несущих срочную информацию, в определенных участках сети для таких процессов заранее предопределены, и передача кадров может осуществляться по заранее запрограммированному расписанию [1]. Однако наряду с таким трафиком существует трафик весьма высокой критичности, появление которого носит случайный характер и не может быть запланировано в строго установленные интервалы времени. Примером

такого трафика являются кадры, сигнализирующие о возникновении предаварийной ситуации, или управляющие сигналы, предотвращающие такую ситуацию. Эти кадры имеют самый высокий приоритет, и если они поступят в очередь в период передачи высокочритичного ранее запланированного трафика, то должны быть обслужены первыми. В остальные периоды указанный трафик обнаруживается в процессе циклического опроса очередей по расписанию.

Расписание задается в виде последовательности очередей [8] ($g = 1, 2 \dots G$). Номер очереди равен $1, \dots, H$. Расписание носит циклический характер, где G – длина цикла. Например, последовательность 1, 3, 1, 4, 2, 1, 3, 1, 4, 2... свидетельствует, что первой обслуживается очередь с номером 1, длина цикла равна 5, а последняя очередь в цикле имеет номер 2. В течение цикла очередь с номером 1 будет обслужена 2 раза, а остальные – по одному разу. Всего различных классов очередей 4. В одной очереди могут находиться пакеты различных соединений, имеющих одинаковый приоритет (приоритеты соединений устанавливаются заранее).

Обозначим через g_h число обращений к очереди с номером h в течение одного цикла (в рассмотренном примере $g_1 = 2$). Величину $\pi_h = \frac{g_h}{G}$ можно назвать вероятностью обращения к h той очереди. Обозначим через h^* номер очереди, число обращений к которой в цикле максимально. В нашем примере $h^* = 1$. Подциклом называют обращения к очередям, заключенные между двумя последовательными обращениями к очереди h^* , включая одно обращение к этой очереди. Значение g_{h^*} определяет число подциклов в расписании. Расписание представляется в виде матрицы (табл. 1). Число строк в матрице равно числу подциклов в расписании, а число столбцов – числу различных групп соединений (числу очередей на рис. 5).

Если в подцикле $i = 1 \dots g_{h^*}$ должно производиться обращение к данной очереди, $h = 1, \dots, H$, то элемент $s_{ik} = 1$, в противном случае $s_{ik} = 0$. Здесь число единиц в каждом столбце g_h равно числу обращений к соответствующей этому столбцу очереди h в течение одного цикла.

Расписание

Очереди		1	2	3	4
Подциклы в цикле	№ 1	1	0	1	0
	№ 2	1	1	0	1

Процесс управления обработкой заявок по расписанию, представленный в [8, 9], заключается в следующем: из матрицы S выбирается строка S_g , соответствующая очередному подциклу. В строке единицами отмечены классы заявок, которые могут быть обслужены в данном подцикле. Если заявки указанных классов отсутствуют, то производится переход к следующему очередному подциклу расписания. Если заявки классов, обслуживаемых в данном подцикле, имеются, то они обслуживаются в порядке следования этих классов в подцикле, а затем осуществляется переход к следующему подциклу.

Введем следующие обозначения: весь трафик разделим на классы $h (h=1, \dots, H)$ согласно приоритетам. В соответствии с технологическим процессом в сети должно образовываться множество соединений $[A]$. Для передачи трафика h должно образовываться подмножество $[A_h]$ соединений $a_{ih} (i=1, \dots, I_h)$. Каждое соединение a_{ih} в данный момент времени обеспечивает передачу только одного класса трафика h с интенсивностью λ_{ih} . В сети имеется множество портов P . Любой порт p_j с номером $j (j=1, \dots, J)$ имеет по N выходов p_{jh} . Каждый выход предназначен для передачи трафика только одного класса h .

Процесс передачи является циклическим, весь цикл разделяется на временные слоты, в течение каждого из которых может передаваться трафик только одного класса h (рис. 6). Если в соответствии с технологическим процессом в течение данного слота должна производиться передача трафика класса h , то в сети из всего запланированного множества соединений выделяется подмножество соединений A_h , обеспечивающих одновременную передачу указанного трафика. Это подмножество характеризуется матрицей, столбцы которой соответствуют соединениям, а каждая строка соответствует одному из выходных портов $p_j (j=1, \dots, J)$ коммутаторов сети. Каждый порт p_j характеризуется интенсивностью μ_j , с которой может обрабатываться проходящий через него трафик. Столбцы матрицы соединений содержат единицы в клетках тех портов, которые задействованы в данном соединении, а единицы в строках матрицы свидетельствуют о том, что соответствующие выходы p_{ih} класса h должны быть открыты в течение всего рассматриваемого слота времени. Одновременно определяется загрузка r_{ij} каждого выхода p_j трафиком соединения $a_i = \frac{\lambda_i}{\mu_j}$. Суммируя загрузку от всех соединений, трафик

которых проходит через рассматриваемый выход, получим суммарную загрузку данного выходного порта, как это показано на рис. 6.

Значения этой загрузки могут быть использованы для управления потоками трафика. Управление открытием выходов портов каждого из коммутаторов может происходить на основании информации, получаемой от всей сети, поэтому наиболее целесообразным следует считать централизацию функций управления в едином устройстве, которое получает данные от всех коммутаторов и на основании заложенной в этом устройстве программы расписания определяет выходы портов всех коммутаторов, которые должны быть активированы в течение данного слота времени. При организации TSN на коммутируемом в данный момент выходе порта коммутатора присутствует трафик только одного класса. Однако очередь одного класса может содержать кадры чувствительного к задержкам трафика нескольких различных приоритетов.

При подключении данного выхода очередь находящихся в нем кадров будет обслуживаться согласно указанным приоритетам, и первыми из этой очереди будут переданы кадры, имеющие наивысший приоритет. Мы уже отмечали, что, например, кадры, свидетельствующие о возможности возникновения аварийной ситуации, имеют наивысший приоритет, должны обладать наименьшей

задержкой и поэтому должны опрашиваться в течение цикла опроса наибольшее число раз.

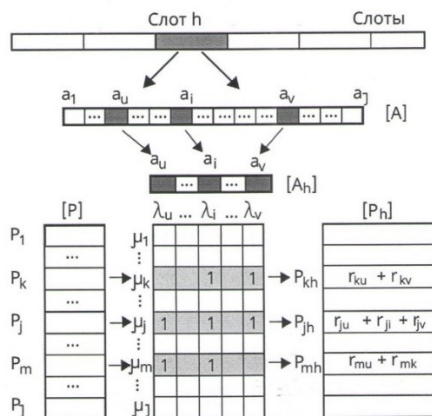


Рис. 6. Выбор открытия выходов в течение слота h

Но кадры данного типа возникают крайне редко и большая часть временных слотов опроса указанного трафика окажутся пустыми, что привело бы к крайне нерациональному использованию пропускной способности канала. Для улучшения пропускной способности наряду с рассматриваемым трафиком, имеющим наивысший приоритет, в очереди размещают другие виды чувствительного к задержкам трафика, имеющие меньший приоритет. Именно поэтому предусматривается возможность размещения в одном классе трафика кадров, имеющих различные приоритеты.

Заключение

TSN представляют новую технологию, обеспечивающую получение детерминистских свойств обычными сетями Ethernet. Технология уже прошла основные этапы научных исследований и стандартизации и в настоящее время широко внедряется в промышленную автоматику [10]. Однако, несмотря на достаточно жесткую стандартизацию, имеется весьма широкое поле деятельности для разработки алгоритмов и программного обеспечения пользователей. Каждый технологический процесс, обеспечиваемый TSN, специфичен и требует тщательной подготовки программного обеспечения пользователей. В настоящее время ряд компаний разрабатывают специальное программное обеспечение, позволяющее автоматизировать указанный процесс, и работы в этом направлении еще много.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Kopetz H.* Real-time systems: design principles for distributed embedded applications. Springer Verlag, 2011.
2. *Finn N.* Detnet problem statement. <http://www.ietf.org/proceedings/91/slides/slides-91-detnet-9.pptx>
3. *Redmill F.* Understanding safety integrity levels. MEASUREMENT and CONTROL, 32(7): 197–200, 1999.
4. *Society I.C.* IEEE 802.1Q Standard. IEEE, 2014. Audio Video Bridging (AVB).

5. *Fiedler M., Hossfeld T., Tran-Gia P.* A generic quantitative relationship between quality of experience and quality of service // IEEE NETWORK, 24(2): 36–41, 2010.
6. *Frances F., Fraboul C., Grieu J.* Using network calculus to optimize the AFDX network. In Proc. ERTS (2006).
7. *Тейлор А., Запке М.* TSN: конвергентные сети для улучшения работы ПоТ. Интернет вещей.
8. Основы теории вычислительных систем: Учеб. пособие для вузов. Под ред. С.А. Майорова. М.: Высшая школа, 1978. 408 с.
9. *Лихтциндер Б.Я.* Особенности TSN // Вестник связи. 2021. 7. С. 32–41.
10. *Brooks' S., Uludag E.* Time-Sensitive Networking: From Theory to implementation in Industrial Automation // White Paper, Intel, WP-01279-1.0. Pp. 1–7.
11. *Silviu S., Craciunas R., Oliver S., Chmelik M. Steiner W.* TTEch Computertechnik AG Schonbrunner StraBe 7, 1040 Vienna, Austria. Presented at the 24th International Conference on Real – Time Networks and Systems (RTNS), ACM, 2016.
12. *Steiner W.* An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks // In Proc. RTSS (2010), IEEE Computer Society.
13. *Craciunas S.S., Serna Oliver R.* SMT-based task and network-level static schedule generation for time-triggered networked systems // In Proc. RTNS (2014), ACM.
14. *Craciunas S.S., Serna Oliver R.* Combined task- and network-level scheduling for distributed time-triggered systems // Real-Time Systems 52, 2 (2016), 161(200).
15. *Tamas-Selicean D., Pop P., Steiner W.* Synthesis of communication schedules for TTEthernet-based mixed-criticality systems // In Proc. CODES+ISSS (2012), ACM.
16. *Hanzalek Z., Burget P., Sucha P.* Pro_net IO IRT message scheduling // In Proc. ECRTS (2009), IEEE.
17. *Huang J., Blech J.O., Raabe A., Buckl C., Knoll A.* Static scheduling of a time-triggered network-on-chip based on SMT solving // In Proc. DATE (2012), IEEE.
18. *Pop P., Eles P., Peng Z.* Schedulability-driven communication synthesis for time triggered embedded systems // Real-Time Syst. 26, 3 (2004), 297 (325).
19. *Zeng H., Zheng W., Di Natale M., Ghosal A., Giusto P., Sangiovanni-Vincentelli A.* Scheduling the xray bus using optimization techniques // In Proc. DAC (2009), ACM.
20. *Bauer H., Scharbarg J., Fraboul C.* Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach // Industrial Informatics, IEEE Transactions on 6, 4 (2010).
21. *Diemer J., Thiele D., Ernst R.* Formal worst-case timing analysis of ethernet topologies with strict-priority and AVB switching // In Proc. SIES (2012), IEEE Computer Society.
22. *Frances F., Fraboul C., Grieu J.* Using network calculus to optimize the AFDX network // In Proc. ERTS (2006).
23. *Specht J., Samii S.* Urgency-based scheduler for time-sensitive switched ethernet networks. In Proc. ECRTS (2016), IEEE Computer Society.
24. *Alderisi G., Patti G., Bello L.L.* Introducing support for scheduled traffic over IEEE audio video bridging networks // In Proc. ETFA (2013), IEEE Computer Society.

Статья поступила в редакцию 05 марта 2022 г.

ETHERNET NETWORKS WITH DETERMINISTIC DELAYS

*B.Ya. Lichtsinder*¹

Volga State University Telecommunications and Informatics
23, L. Tolstogo st., Samara, 443100, Russian Federation

E-mail: lixt@psuti.ru

Abstract. *The work is devoted to a new technology that provides low and predictable data transmission delays in industrial Ethernet networks. The traffic is divided into critical and non-critical to delays. The features of TSN functioning are considered. The operating principle and architecture of such networks are considered. It is shown that cost reduction can be achieved by having only one network that meets the requirements of deterministic networking and is compatible with modern Ethernet solutions. Examples are given and it is shown that deterministic networks can be used to create hard real-time systems. This is especially true for automatic regulation and control systems that include feedback circuits. The key features of TSN that provide guaranteed message delivery times are traffic transmission time synchronization and scheduling. The TSN switch is presented as two subsystems, one of which performs standard Ethernet functions, and the second provides the TSN control logic. An example of cyclic traffic control according to the schedule is considered. The prospects of using TSN technology in industrial automation control networks are shown.*

Keywords: *traffic, deterministic delays, control, switches, networks, cyclicity*

REFERENCES

1. *Kopetz H.* Real-time systems : design principles for distributed embedded applications. Springer Verlag, 2011.
2. *Finn N.* Detnet problem statement. <http://www.ietf.org/proceedings/91/slides/slides-91-detnet-9.pptx>
3. *Redmill F.* Understanding safety integrity levels. MEASUREMENT and CONTROL, 32(7): 197–200, 1999.
4. *Society I.C.* IEEE 802.1Q Standard. IEEE, 2014. Audio Video Bridging (AVB).
5. *Fiedler M., Hossfeld T., Tran-Gia P.* A generic quantitative relationship between quality of experience and quality of service // IEEE NETWORK, 24(2): 36–41, 2010.
6. *Frances F., Fraboul C., Grieu J.* Using network calculus to optimize the AFDX network. In Proc. ERTS (2006).
7. *Tejlor A., Zapke M.* TSN: konvergentnye seti dlya uluchsheniya raboty IIoT. Internet veshchej.
8. *Osnovy teorii vychislitel'nyh sistem.* Uchebnoe posobie dlya vuzov. Pod red. *S.A. Majorova.* M.: Vysshaya shkola, 1978. 408 s. [In Russian].
9. *Lihtcinder B.Ya.* Osobennosti TSN // Vestnik svyazi. 2021. 7. C. 32–41. [In Russian].
10. *Brooks' S., Uludag E.* Time-Sensitive Networking: From Theory to implementation in Industrial Automation // White Paper, Intel, WP-01279-1.0. Pp. 1–7.
11. *Silviu S., Craciunas R., Oliver S., Chmelik M.* Steiner W.TT Tech Computertechnik AG Schonbrunner StraBe 7, 1040 Vienna, Austria. Presented at the 24th International Conference on Real – Time Networks and Systems (RTNS), ACM, 2016.
12. *Steiner W.* An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks // In Proc. RTSS (2010), IEEE Computer Society.
13. *Craciunas S.S., Serna Oliver R.* SMT-based task and network-level static schedule generation for time-triggered networked systems // In Proc. RTNS (2014), ACM.
14. *Craciunas S.S., Serna Oliver R.* Combined task- and network-level scheduling for distributed time-triggered systems // Real-Time Systems 52, 2 (2016), 161(200).

¹ *Boris Ya. Lichtsinder (Dr. Sci. (Techn.)), Professor.*

15. *Tamas-Selicean D., Pop P., Steiner W.* Synthesis of communication schedules for TTEthernet-based mixed-criticality systems // In Proc. CODES+ISSS (2012), ACM.
16. *Hanzalek Z., Burget P., Sucha P.* Pro_net IO IRT message scheduling // In Proc. ECRTS (2009), IEEE.
17. *Huang J., Blech J.O., Raabe A., Buckl C., Knoll A.* Static scheduling of a time-triggered network-on-chip based on SMT solving // In Proc. DATE (2012), IEEE.
18. *Pop P., Eles P., Peng Z.* Schedulability-driven communication synthesis for time triggered embedded systems // Real-Time Syst. 26, 3 (2004), 297 (325).
19. *Zeng H., Zheng W., Di Natale M., Ghosal A., Giusto P., Sangiovanni-Vincentelli A.* Scheduling the exray bus using optimization techniques // In Proc. DAC (2009), ACM.
20. *Bauer H., Scharbag J., Fraboul C.* Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach // Industrial Informatics, IEEE Transactions on 6, 4 (2010).
21. *Diemer J., Thiele D., Ernst R.* Formal worst-case timing analysis of ethernet topologies with strict-priority and AVB switching // In Proc. SIES (2012), IEEE Computer Society.
22. *Frances F., Fraboul C., Grieu J.* Using network calculus to optimize the AFDX network // In Proc. ERTS (2006).
23. *Specht J., Samii S.* Urgency-based scheduler for time-sensitive switched ethernet networks. In Proc. ECRTS (2016), IEEE Computer Society.
24. *Alderisi G., Patti G., Bello L.L.* Introducing support for scheduled traffic over IEEE audio video bridging networks // In Proc. ETFA (2013), IEEE Computer Society.