

Научная статья

УДК 621.395.44

<https://doi.org/10.31854/1813-324X-2026-12-3-72-80>

EDN:EMQKGN



## Оценка предельного количества потребителей в домене TSN при заданных параметрах качества синхронизации

Николай Леонидович Сторожук, storozhuk.nl@sut.ru

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

### Аннотация

**Актуальность.** Для функционирования активно развивающегося в последние годы промышленного интернета вещей необходимо обеспечение обмена данными в режиме реального времени. Решить эту задачу оптимальным образом позволяет применение технологии построения сетей, чувствительных ко времени, которая является расширением классического Ethernet и призвана обеспечить более высокое качество передачи трафика, для которого задержки в сети недопустимы. Качественные параметры передачи определяются точностью временной синхронизации сетевых устройств, которая, в свою очередь, зависит и от их числа в домене, увеличение числа сетевых устройств в синхронизационном домене обычно ее снижает, поэтому исследование влияния количества потребителей на точность синхронизации является актуальной научно-практической задачей.

**Цель:** исследовать влияния количества потребителей на точность временной синхронизации в сетевых доменах и рассмотреть зависимость параметров качества синхронизации от количества ее потребителей в домене.

**Методы:** зависимость качества синхронизации шкалы времени от количества ее потребителей в домене оценивается по результатам имитационного моделирования процессов временной синхронизации в среде AnyLogic.

**Результат.** В работе показано, что точность синхронизации напрямую влияет на качественные параметры передачи данных, а увеличение нагрузки в виде количества запросов изначально ведет к заметному росту длительности цикла синхронизации. При этом дальнейшее увеличение нагрузки ведет к изменению характера функции распределения времени обработки, вплоть до проявления характера равномерного распределения на всем диапазоне значений.

**Новизна:** разработанные имитационные модели процессов сведения шкал времени позволяют оптимизировать соотношение количества потребителей синхронизации в домене с ее качественными параметрами.

**Практическая значимость:** описанные модели позволяют оценивать предельное количество потребителей в домене, при котором сохраняется заданное качество синхронизации, а также позволяют оценить время, которое необходимо на корректировку локальных часов рассматриваемого устройства, а изучение влияния количества потребителей на точность синхронизации позволяет разрабатывать более надежные и эффективные сетевые решения, адаптированные к современным требованиям к производительности и надежности.

**Ключевые слова:** синхронизация времени, TSN, PTP, gPTP, IIoT, шкала времени

**Ссылка для цитирования:** Сторожук Н.Л. Оценка предельного количества потребителей в домене TSN при заданных параметрах качества синхронизации // Труды учебных заведений связи. 2026. Т. 12. № 3. С. 72–80. DOI:10.31854/1813-324X-2026-12-3-72-80. EDN:EMQKGN

Original research  
<https://doi.org/10.31854/1813-324X-2026-12-3-72-80>  
EDN:EMQKGH

# Estimation of the Maximum Consumers Number in a TSN Domain with Synchronization Quality Parameters Set

✉ Nikolay L. Storozhuk, storozhuk.nl@sut.ru

The Bonch-Bruevich Saint Petersburg State University of Telecommunications,  
St. Petersburg, 193232, Russian Federation

## Annotation

**Relevance.** The Industrial Internet of Things, which has been rapidly developing in recent years, is one of the technologies whose operation requires real-time data exchange. This challenge can be optimally addressed by using time-sensitive networking technology, which is an extension of classic Ethernet designed to ensure higher-quality traffic transmission where network delays are unacceptable. Transmission quality is determined by the accuracy of network devices time synchronization, which in turn depends on their number in domain. Therefore, studying the impact of the number of consumers on synchronization accuracy is a pressing scientific and practical challenge.

**Objective.** To examine the dependence of synchronization quality parameters on the number of its consumers in a domain. **Methods.** The dependence of time scale synchronization quality on the number of its consumers in a domain is assessed using the results of time synchronization processes simulation modeling within AnyLogic environment.

**Result.** The study demonstrates that an increase in the number of requests initially increases the synchronization cycle duration significantly. And its further increase in the load leads to the processing time distribution function nature change, till the entire range of values become distributed uniformly.

**Novelty.** The simulation models of timescale convergence processes developed makes able to optimize the ratio between the numbers of synchronization consumers in a domain to its quality parameters.

**Practical Relevance.** The models described make able to estimate the maximum number of consumers in a domain that maintains a given synchronization quality, as well as the time required to adjust the local device clock.

**Keywords:** time synchronization, TSN, PTP, gPTP, IIoT, timescale

**For citation:** Storozhuk N.L. Estimation of the Maximum Consumers Number in a TSN Domain with Synchronization Quality Parameters Set. *Proceedings of Telecommunication Universities*. 2026;12(3):72–80. (in Russ.) DOI:10.31854/1813-324X-2026-12-3-72-80. EDN:EMQKGH

## Введение

Согласно национальной стратегии цифрового развития искусственного интеллекта (ИИ), к 2030 г. в нашей стране доля отраслей с высоким уровнем внедрения ИИ вырастет до 95 %<sup>1</sup>. В России, по данным опроса Ассоциации менеджеров, 39 % крупных предприятий уже используют ИИ-инструменты, а 25 % опрошенных планируют их внедрить в ближайшем будущем<sup>2</sup>. Это означает, что в течение нескольких лет объем рынка платформ промышленного интернета вещей (IIoT, аббр. от англ.

Industrial Internet of Things) будет стремительно расти. IIoT является многоуровневой системой, включающей в себя установленные на узлах и агрегатах промышленных объектов датчики, контроллеры, исполнительные механизмы, а также средства передачи собираемых данных и команд управления. А для функционирования этой системы необходима телекоммуникационная сеть, обеспечивающая гарантированную и своевременную доставку передаваемой информации. Одна из основных проблем эксплуатации современных сетей –

<sup>1</sup> Указ Президента РФ от 10 октября 2019 г. № 490 «О развитии искусственного интеллекта в Российской Федерации».

<sup>2</sup> «K2 НейроТех» назвал шесть ИИ-трендов в промышленности в 2025 году // Cnews. 2025. URL: [https://www.cnews.ru/news/line/2025-05-28\\_k2\\_nejroteh\\_nazval\\_shest](https://www.cnews.ru/news/line/2025-05-28_k2_nejroteh_nazval_shest) (дата обращения 01.06.2026)

конвергенция пакетов систем передачи данных и мультимедийного трафика. Как следствие – резкое возрастание объема передаваемой информации [1]. А это, в свою очередь, приводит к увеличению задержек передачи и вероятности потери отдельных сообщений. Поэтому для эффективного контроля качества передачи цифрового сигнала в первую очередь необходимо выбрать критерий оценки [2]. Обеспечить обмен данными в системах IIoT на основе сетей Ethernet способна технология построения сетей, чувствительных ко времени (TSN, аббр. от англ. Time-Sensitive Networking).

TSN расширяет технологию Ethernet, включая в себя следующий ряд механизмов, необходимых для работы в режиме реального времени [3]:

- синхронизацию времени, обеспечивающую все элементы сетевой инфраструктуры единым высокоточным временем;
- планирование в режиме реального времени, позволяющее минимизировать задержки кадров с высоким приоритетом;
- резервирование и конфигурирование канала передачи данных, обеспечивающие резервирование потоков данных и организацию очередей.

Синхронизация сетевого оборудования по частоте и времени является одновременно фундаментальной целью и ключевым фактором в выполнении других функций TSN [4]. Таким образом, чтобы достаточно разветвленная сеть Ethernet с большим количеством узлов и устройств могла эффективно функционировать, в том числе при обработке трафика, чувствительного к задержкам, необходимо обеспечить ее строгую временную синхронизацию.

### Архитектура коммутационного ядра Ethernet TSN

В общем случае система частотно временно обеспечения сети связи с коммутацией пакетов состоит из технических средств системы синхронизации шкал времени, а также средств связи, являющихся потребителями сигналов синхронизации [5]. Для сквозной передачи с жесткими временными рамками все устройства сети должны иметь общую систему отсчета времени и, следовательно, синхронизировать свои часы. Это справедливо не только для конечных устройств коммуникационных потоков, таких как промышленный контроллер и производственный робот, но и для сетевых компонентов, например, коммутаторов Ethernet. Только благодаря синхронизации часов всех сетевых устройств они могут работать согласованно и выполнять требуемые операции точно в нужный момент. Коммутационное ядро L2 Ethernet TSN играет ключевую роль в этих процессах. Оно обеспечивает передачу данных по всем портам. Каждый порт имеет необходимое число исходящих очередей, которые контролируются многоуровневым планировщиком.

Ядро построено на основе архитектуры общей буферной памяти, способной одновременно переключаться со скоростью проводного соединения на всех портах без блокировки<sup>3</sup>. Пакеты хранятся в общей буферной памяти в виде ячеек фиксированного размера как показано на рисунке 1.

Коммутационное ядро Ethernet TSN обеспечивает выполнение определенных функций<sup>3</sup>.

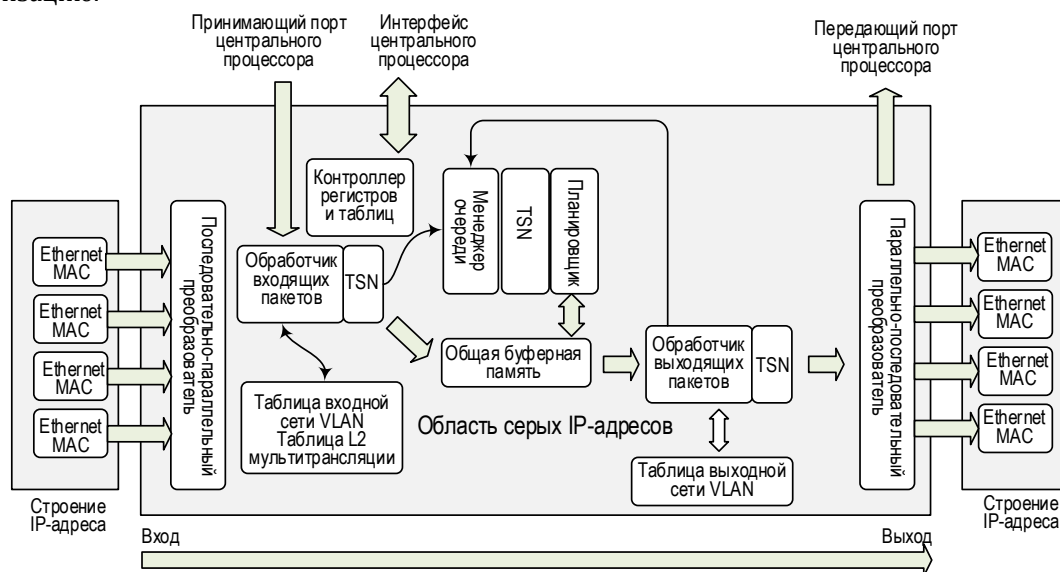


Рис. 1. Ядро коммутатора Ethernet TSN

Fig. 1. TSN Ethernet Switch Core

<sup>3</sup> L2/L3 Ethernet TSN Switch Core an Example TSN Design. User Guide. 2024. URL: [https://www.packetarc.com/datasheet\\_switch\\_tsn.pdf](https://www.packetarc.com/datasheet_switch_tsn.pdf) (дата обращения 01.06.2026)

*Функция 1.* Поддержка номинальной скорости передачи данных проводного соединения на всех портах и на всех размерах кадров Ethernet.

*Функция 2.* Хранение и пересылка состояния разделяемой памяти.

*Функция 3.* Поддержка больших пакетов размером до 32746 байт.

*Функция 4.* Поддержка сети, чувствительной ко времени.

*Функция 5.* Поддержка операций управления очередями, в т. ч. обеспечивается:

- отключение планирования пакетов на порт;
- отключение постановки новых пакетов в очередь к порту;
- разрешение опустошения порта без отправки пакетов;
- разрешение проверки занятости порта.

*Функция 6.* Зеркальное отображение входов и выходов.

*Функция 7.* Поддержка анализатора портов удаленного коммутатора.

*Функция 8.* Автоматическое обновление адресов L2.

*Функция 9.* Поддержка связующего дерева, проверки входящего и исходящего трафика.

*Функция 10.* Поддержка таблиц трансляции исходящих виртуальных локальных сетей (VLAN, *аббр. от англ.* Virtual Local Area Network), обеспечивающих уникальное преобразование идентификатора виртуальной локальной сети (VID, *аббр. от англ.* Virtual Local Area Network Identifier) для каждого исходящего порта.

*Функция 11.* Поддержка списка управления доступом (ACL, *аббр. от англ.* Access Control List), который определяет, кто может получать доступ к ресурсам и разрешенные действия, результатом которых может быть:

- назначение нескольких действий каждому результату, а все результаты могут быть получены параллельно;
- отбрасывание пакета;
- отправка пакета на определенный порт;
- обновление счетчика;
- отправка пакета в определенную очередь на выходном порту;
- присвоение «цвета» пакету, который используется измерителем / маркером / ограничителем, что позволяет определять действия с пакетами разного цвета;
- принудительное использование пакетом определенного VID при поиске таблицы VLAN;
- создание входного зеркала для пакета;
- запрет на получение пакета в таблице коммутации.

*Функция 12.* Поддержка настраиваемого механизма классификации и ACL.

*Функция 13.* Поддержка настраиваемого сопоставления исходящей очереди из битов, указывающих тип обслуживания или приоритет кадра.

*Функция 13.* Присутствие дефицитного-взвешенного циклического планировщика.

*Функция 14.* Контроль многоадресных и широко-вещательных штормов с помощью отдельных специализированных сегментов маркеров.

Если сформулировать кратко, то коммутационное ядро Ethernet TSN обеспечивает выполнение функций синхронизации времени, планирования трафика, резервирования и контроля ошибок.

### Диаграмма состояний блока синхронизации

На зависящем от среды передачи – медиазависимом (MD, *аббр. от англ.* Media Dependent) – уровне порты получают с медианезависимого (MI, *аббр. от англ.* Media Independent) уровня данные, соответствующие видам сетевой технологии, к которой непосредственно подключен порт. Для описания процесса функционирования блока синхронизации коммутационного ядра TSN следует рассматривать MI уровень, описывающий взаимосвязи элементов процесса, локальные и глобальные переменные, которые им управляют. Устройство с реализованным в нем функционалом PTP, которое называют «экземпляром PTP», реализует функции выбора лучших часов и синхронизации. Данные функции включают параметры, специфические для конкретного порта, и параметры, которые относятся к экземпляру PTP в целом. Функции распределены между определенным множеством логических структур, которые совместно определяют поведение этого экземпляра. Так как указанные функции описываются конкретными конечными автоматами, то функционирование блока синхронизации в составе коммутатора TSN в целом можно описать таким же способом.

В стандарте IEEE 1588-2002 описаны следующие виды источников шкал времени [6]:

- гроссмейстерские часы (грандмастер – GM, *аббр. от англ.* Grandmaster) – основной источник точного времени;
- обычные часы (OC, *аббр. от англ.* Ordinary Clock) – устройство с одним портом, которое может быть ведущими часам или локальными ведомыми часами;
- ведущие часы (*от англ.* Master) – выступают в качестве источников точного времени для конечных устройств;
- ведомые часы (*от англ.* Slave) – конечные устройства, на которых необходимо осуществить синхронизацию времени по протоколу PTP;
- граничные часы (BC, *аббр. от англ.* Boundary Clock) – сетевое оборудование, которое выступает в качестве ведомого устройства для GM и источником точного времени для конечных устройств.

Функция временной синхронизации экземпляра РТР описывается с использованием конечных автоматов, их локальных переменных, взаимосвязей, глобальных переменных и структур, которые обеспечивают данные взаимосвязи. Для построения модели процесса используется мультиагентный подход в среде имитационного моделирования AnyLogic. Рассматриваемая имитационная модель отражает принципы функционирования экземпляра РТР при обработке информации о синхронизации на МI уровне. Логика выполнения модели описывается последовательностью процессов, которая приводится на рисунке 2.

Изображенные на рисунке блоки представляют собой конечные автоматы, которые реализуют различные функции и передают друг другу резуль-

таты их выполнения. Экземпляр РТР включает в себя объект *LocalClock*, который представляет собой автономные локальные часы, обеспечивающие для него общее время. *PortSyncSync* – это механизм синхронизации, позволяющий синхронизировать данные между портами на МI уровне.

Относительно этого объекта отсчитываются все предоставляемые временные метки, также он предоставляет значение *currentTime* (текущее время), которое используется в конечных автоматах для установки различных таймеров. Так как объект *LocalClock* не является конечным автоматом и сам по себе не реализует каких-либо функций, он и его связь с другими блоками экземпляра РТР на рисунке 2 показаны пунктиром. В таблице 1 приведено назначение каждого из этих объектов [7].

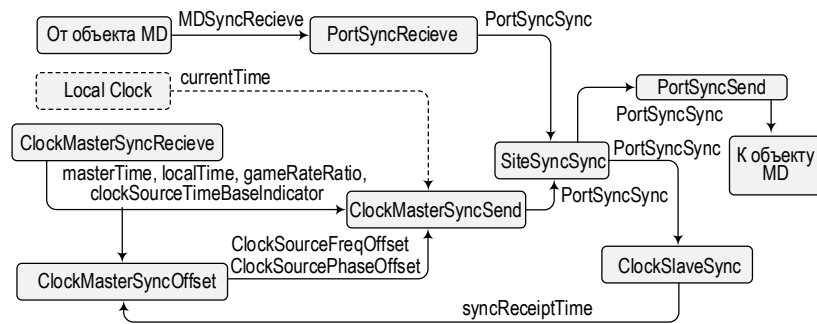


Рис. 2. Адаптированная диаграмма состояний блока синхронизации

Fig. 2. Adapted State Diagram of the Synchronization Block

ТАБЛИЦА 1. Назначение функциональных блоков модели (по одному на каждый экземпляр РТР)

TABLE 1. Purpose of the functional blocks of the model

| Функциональный блок           | Назначение блока  |
|-------------------------------|---|
| <i>ClockMasterSyncReceive</i> | Принимает функции <i>ClockSourceTime.invoke</i> от объекта <i>ClockSource</i> и уведомления о тактах объекта <i>LocalClock</i> ; обновляет <i>masterTime</i> и предоставляет <i>masterTime</i> конечным автоматам <i>ClockMasterSyncOffset</i> и <i>ClockMasterSyncSend</i>   |
| <i>ClockMasterSyncOffset</i>  | Принимает <i>syncReceiptTime</i> от объекта <i>ClockSlave</i> и <i>masterTime</i> от конечного автомата <i>ClockMasterSyncReceive</i> ; вычисляет сдвиги фазы и частоты между <i>masterTime</i> и <i>syncReceiptTime</i> (если экземпляр РТР не уровня GM) и передает сдвиги частоты и фазы конечному автомату <i>ClockMasterSyncSend</i>   |
| <i>ClockMasterSyncSend</i>    | Принимает <i>masterTime</i> от конечного автомата <i>ClockMasterSyncReceive</i> ; принимает сдвиги фазы и частоты между <i>masterTime</i> и <i>syncReceiptTime</i> от конечного автомата <i>ClockMasterSyncOffset</i> и передает <i>masterTime</i> , сдвиги фазы и частоты объекту <i>SiteSync</i> , используя структуру <i>PortSyncSync</i>  |
| <i>PortSyncSyncReceive</i>    | Принимает информацию о временной синхронизации от объекта MD порта РТР; вычисляет накопленное значение <i>rateRatio</i> ; вычисляет <i>syncReceiptTimeoutTime</i> и отправляет информацию объекту <i>SiteSync</i>   |
| <i>SiteSync</i>               | Принимает информацию о временной синхронизации, накопленном значении <i>rateRatio</i> и значении <i>syncReceiptTimeoutTime</i> от объекта <i>PortSync</i> текущего ведомого порта или от объекта <i>ClockMaster</i> и отправляет эту информацию объектам <i>PortSync</i> всех портов и объекту <i>ClockSlave</i>  |
| <i>PortSyncSyncSend</i>       | Принимает информацию о временной синхронизации от объекта <i>SiteSync</i> ; запрашивает у объекта MD порта РТР отправку связанного с временной синхронизацией сообщения о событии; принимает <i>syncEventEgressTimestamp</i> для этого сообщения о событии от объекта MD; использует последнюю информацию о временной синхронизации, полученную от объекта <i>SiteSync</i> , и временную метку, чтобы просчитать информацию о временной синхронизации, которая будет отправлена объектом MD в общем сообщении или в последующем сообщении о событии, и отправляет эту последнюю информацию объекту MD |
| <i>ClockSlaveSync</i>         | Принимает информацию о временной синхронизации от объекта <i>SiteSync</i> ; вычисляет <i>clockSlaveTime</i> и <i>syncReceiptTime</i> ; устанавливает <i>syncReceiptLocalTime</i> , <i>gmTimeBaseIndicator</i> , <i>lastGmPhaseChange</i> и <i>lastGmFreqChange</i> ; отправляет <i>clockSlaveTime</i> объекту <i>ClockMaster</i> и предоставляет информацию объекту <i>ClockTarget</i> , позволяя этому объекту определить, имел ли место разрыв фазы или частоты   |

Как видно из рисунка 2, блоки передают друг другу данные либо в виде одной или нескольких глобальных переменных, либо в виде структур данных, состав которых определен стандартом IEEE 802.1AS. В таблице 2 приведены отдельные глобальные переменные, которые не входят в состав вышеназванных структур данных, но которыми также обмениваются автоматы, приведенные на рисунке 2 [7].

ТАБЛИЦА 2. Назначение глобальных переменных

TABLE 2. Assigning Global Variables

| Наименование переменной             | Назначение переменной  |
|-------------------------------------|--|
| <i>clockSourceFreqOffset</i>        | Сдвиг частоты объекта <i>ClockSource</i> относительно текущей частоты GM   |
| <i>clockSourcePhaseOffset</i>       | Разность между временем, показываемым объектом <i>ClockSource</i> , и синхронизированным временем  |
| <i>clockSourceTimeBaseIndicator</i> | Глобальная переменная, значение которой объект <i>ClockMaster</i> устанавливает равным параметру <i>timeBaseIndicator</i> функции <i>ClockSourceTime.invoke</i> интерфейса приложения. Параметр <i>timeBaseIndicator</i> для <i>ClockSourceTime.invoke</i> задается объектом <i>ClockSource</i> и изменяется этим объектом всякий раз, когда изменяется его временная база |
| <i>currentTime</i>                  | Текущее значение времени относительно часов объекта <i>LocalClock</i> (т. е. текущее локальное время)  |
| <i>gmRateRatio</i>                  | Измеренное отношение частоты объекта <i>ClockSource</i> к частоте объекта <i>LocalClock</i>  |
| <i>localTime</i>                    | Значение <i>currentTime</i> в момент, когда последняя функция <i>ClockSourceTime.invoke</i> была получена от объекта <i>ClockSource</i> или когда объект <i>LocalClock</i> в последний раз обновил свое время  |
| <i>masterTime</i>                   | Время, поддерживаемое объектом <i>ClockMaster</i> , на основе информации, полученной от объектов <i>ClockSource</i> и <i>LocalClock</i>  |
| <i>syncReceiptTime</i>              | Синхронизированное время, вычисляемое объектом <i>ClockSlave</i> в момент получения информации о временной синхронизации, содержащейся в структуре <i>PortSyncSync</i>   |

### Моделирование работы локального источника шкалы времени

Для унификации процессов обмена информацией между различными блоками в рамках модели вышеописанные структуры данных и переменные представляются в виде обобщенных агентов, которые перемещаются между функциональными блоками и обрабатываются ими. Адаптированная диаграмма состояний блока синхронизации, представленная на рисунке 2, содержит набор функциональных блоков, которые взаимодействуют между собой в зависимости от роли устройства и типа получаемой информации. Каждый блок соответствует конечному автомату, который реализует заданную функцию и передает результат на последующие блоки, с которыми он имеет взаимосвязь. Данная взаимосвязь обеспечивается с помощью агентов, описываемых параметрами и обрабатываемых блоками.

Приведенная диаграмма соответствует описанию процесса функционирования блока синхронизации в общем, с учетом возможности выбора сценария функционирования. Необходимо понимать, что в зависимости от роли устройства в сети взаимосвязи между процессами перестраиваются, при этом различаются этапы обработки информации. В сети TSN выделяются конечный и пересылающий экземпляры RTP, которые можно отнести к соответствующему типу часов.

Конечный экземпляр RTP может рассматриваться как локальные ОС, а пересылающий – как ВС. Предлагаемая модель описывает процесс функционирования пересылающего экземпляра RTP, который соответствует ВС.

Задача ВС заключается в том, что они синхронизируют ОС от GM, получая информацию на ведомый порт, и после подстройки собственных часов отправляют сообщения синхронизации с остальных портов, которые функционируют как ведущие. Подстройка ОС предполагает, что, определяя смещение относительно GM, подсистема подстройки часов осуществляет постепенное изменение частоты, обеспечивая выравнивание шкалы путем уменьшения разности шкал на заданную величину шага изменения  $T_{изм}^4$ . Также может быть использован механизм половинного деления, при котором величина сдвига за цикл  $T_{шаг}$  уменьшается вдвое до достижения заданной точности. Следовательно, математически процесс выравнивания шкалы времени (за один цикл подстройки) можно описать блок-схемой, которая приводится на рисунке 3а (для пошагового изменения) и блок-схемой на рисунке 3б (для метода половинного деления).

<sup>4</sup> Как работает протокол синхронизации PTPv2 // Elec.ru. 2020. URL: <https://www.elec.ru/publications/tsifrovye-tehnologii-svjaz-izmerenija/6015> (дата обращения 01.06.2026)

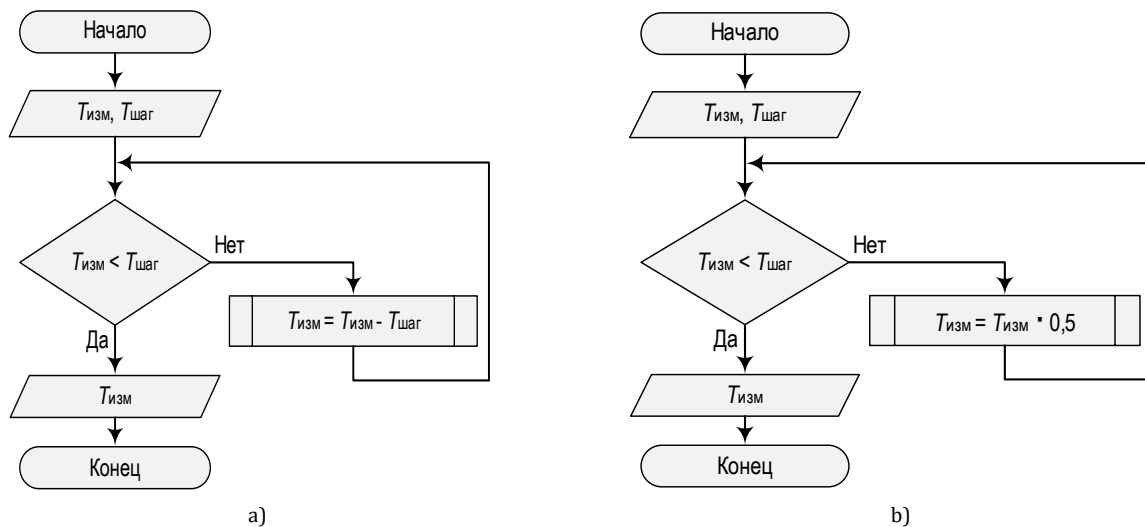


Рис. 3. Блок-схема подстройки локальных часов: а) при пошаговом изменении сдвига; б) при использовании механизма половинного деления

Fig. 3. Block Diagram of Local Clock Adjustment: a) with Step-by-Step Change of Shift; b) with Use of Half Division Mechanism

Здесь полагаем, что локальным часам известно начальное смещение относительно часов GM –  $T_{изм}$ . Процесс подстройки состоит в постепенном уменьшении величины этого смещения на заданный шаг  $T_{шаг}$  (см. рисунок 3а) или делении временного промежутка на 2 (см. рисунок 3б). Величина начального сдвига определяет количество итераций, которые требуются для одного цикла подстройки. Когда величина смещения не превышает шага изменения, локальные часы синхронизированы.

#### Условия имитационного моделирования

Рассмотрим процесс функционирования имитационной модели, учитывая описанную логику процесса. На вход модели поступает агент, несущий информацию о том, какой сдвиг шкалы времени соответствует локальным часам относительно действующего в сети GM. Каждый блок, который участвует в процессе, обрабатывает агента в соответствии с заданными правилами.

Имитационная модель работы локального источника шкалы времени в коммутаторе сети TSN строится в соответствии с разделением на функции приложений, MI и MD уровни.

Анализ функционирования блока синхронизации производился при различных параметрах модели:

1) начальный сдвиг локальной шкалы времени в диапазоне:

- от 50 до 150 нс,
- от 250 до 350 нс,
- от 450 до 550 нс,
- от 950 до 1050 нс,
- от 2450 до 2550 нс;

2) методы подстройки:

- равномерный шаг,

- механизм половинного деления;
- при разном количестве потребителей, а следовательно, и при различной загрузке элемента SiteSyncSync, поступающими запросами, где величина вызванной задержки распределена по нормальному закону, описываемому параметрами среднееквадратическое отклонение (СКО) и математическое ожидание (МО):

$$\text{СКО } \sigma = 5, \text{ МО } \mu = 25,$$

$$\text{СКО } \sigma = 5, \text{ МО } \mu = 75,$$

$$\text{СКО } \sigma = 50, \text{ МО } \mu = 75.$$

#### Моделирование синхронизации при изменении начального сдвига

Имитационное моделирование проводится следующим образом: для каждого диапазона значений начального сдвига, где он сгенерирован как случайная величина, получено пять реализаций моделирования, которые показали определенный результат.

Среднее значение времени выполнения цикла синхронизации для экземпляра RTP (в установленном режиме) составляет примерно 155 нс. Значение начального сдвига не оказывает существенного влияния на среднее значение времени реализации цикла и его дисперсию, а СКО времени реализации цикла для установленного режима находится в пределах 60 нс.

Наличие начального сдвига между GM и ведомыми часами ведет к появлению длительного периода начальной подстройки, который в среднем в 8–10 раз превышает данное значение начального сдвига.

При устоявшемся режиме наблюдается наличие двух устойчивых мод в составе распределения времени синхронизации, что может быть результатом ухода часов, кратного шагу подстройки. Двухмодальность проявляется не во всех экспериментах. Несмотря на наличие больших значений начального сдвига, алгоритм конечного автомата экземпляра РТР приводит к устойчивому режиму удержания синхронизации.

Внутри одного эксперимента пять самостоятельных реализаций подстройки локальных часов отражают корректность и адекватность сформированной модели, выбранных законов распределения случайных величин, что следует из независимого характера полученных результатов для каждой из реализаций.

### Моделирование синхронизации при различных методах подстройки

Между локальными часами и GM в разные периоды присутствует расхождение шкал времени. В начальный момент времени это расхождение может быть довольно большим, в процессе синхронизации оно становится менее значительным. Но для любого варианта экземпляра РТР реализует процедуру синхронизации в итеративном режиме, где на каждом шаге уменьшается расхождение шкал, проверяется полученный результат и выдается новое значение для коррекции шкалы локальных часов.

Данная циклическая процедура может характеризоваться как скоростью сходимости (длительностью цикла реализации), так и значением точности, в том числе реализацией монотонности процедуры коррекции часов. Не удалось найти информации в явном виде о выбранных методах и алгоритмах реализации процедуры сведения шкал при перечисленных выше противоречивых требованиях ни в отечественных, ни в зарубежных публикациях. Также вероятно, что при различных начальных условиях возможна различная оптимальность применения различных алгоритмов коррекции шкал времени.

Были смоделированы два алгоритма: с равномерной подстройкой и основанный на методе половинного деления.

В результате проведенного моделирования выяснилось, что для режима первичной корректировки разные методы подстройки существенно различаются. Например, для случая начальной расстройки 2450–2550 нс СКО: для алгоритма с равномерным шагом составляет 2575 нс, для метода половинного деления и 100 нс, а в установившемся режиме СКО: для алгоритма равномерной подстройки – 150 нс, а для алгоритма с половинным делением 145 нс.

Для установившегося режима среднее время выполнения цикла подстройки составляет около 155 нс и для механизма равномерной подстройки, и для метода половинного деления. Следовательно, алгоритм влияет на длительность первичной подстройки, не вызывая значительных отклонений в установившемся режиме. Величина СКО длительности цикла при этом равна примерно 60 нс.

Сравнивая различные методы подстройки, можно сделать вывод, что метод половинного деления показал более высокую эффективность. Так, при малых значениях начального сдвига выигрыш составляет 10–20 %, при больших значениях расстройки выигрыш в десятки раз превышает первый метод.

### Моделирование синхронизации при изменении количества запросов

Так как контроль выполнения цикла подстройки выполняет блок *SiteSyncSync*, то на него поступают входящие запросы. Увеличение количества запросов увеличивает время их обработки. Следовательно, изменение нагрузки отразится на длительности задержки в работе блока *SiteSyncSync*.

Моделирование показало, что изменение параметров рассматриваемого блока меняет поведение модели. Так при небольших значениях СКО и МО величины задержки, вызванной поступающими запросами, наблюдается выброс значений времени вхождения в синхронизм в районе 700–1300 нс, который обусловлен первичной подстройкой. При этом в установившемся режиме распределение времени вхождения в синхронизм носит многомодальный характер со значением МО около 155 нс и СКО 55 нс.

Дальнейшее увеличение МО приводит к тому, что модальность сохраняется, но положение мод смещается в сторону увеличения времени выполнения цикла подстройки. Выброс, характерный для времени первичного цикла подстройки, смещается за 1000 нс, что говорит об увеличении времени для достижения синхронизации.

При больших значениях СКО ( $\sigma = 50$ ) положение выброса незначительно смещается в сторону увеличения, но характер распределения времени в установившемся режиме изменяется: исчезает многомодальность. Это обусловлено увеличением общей дисперсии, которая зависит от параметров процесса *SiteSyncSync*.

Следовательно, увеличение нагрузки в виде количества запросов изначально ведет к заметному росту дисперсии длительности цикла синхронизации. При этом дальнейшее увеличение нагрузки ведет к изменению характера функции распределения времени обработки вплоть до проявления характера равномерного распределения на всем диапазоне значений.

Результат моделирования по данному параметру показывает, что блок *SiteSyncSync* является ключевым и его производительность в наибольшей мере влияет эффективность экземпляра РТР.

### Заключение

Полученная модель функционирования модуля синхронизации позволяет оценить время, которое необходимо на корректировку локальных часов рассматриваемого устройства. Так как моделирование выполнялось для граничных часов, которые в терминологии TSN соответствуют пересылаю-

щему экземпляру РТР, часть блоков не задействована. При необходимости (рассмотрении сценариев работы коммутатора) модель может быть откорректирована. За основу построения взята диаграмма состояний. Модель экземпляра РТР и процессов его функционирования является ключевой для реализации моделей верхнего уровня, а именно взаимодействия различных видов экземпляров РТР и реализации сетевых режимов их работы. Таким образом, подобное моделирование может позволить оценивать предельное количество потребителей в домене TSN, при котором сохраняется заданное качество синхронизации.

### Список источников

1. Бугров Д.А., Сторожук Н.Л. Применение эволюционного метода для оптимизации магистральной сети связи // Электросвязь. 2007. № 5. С. 30–33. EDN:IADPFX
2. Власов И.И., Новиков Э.В., Птичников М.М., Сторожук Н.Л. Цифровые сети связи. Кабельные и волоконно-оптические линии. М.: ФАЗИС, 2008. 500 с.
3. Берёзкин А.А., Паршин А.А., Парфенов Д.Д., Киричек Р.В. Анализ стандартов сетей, синхронизируемых по времени, для управления роботизированными системами в режиме реального времени // Электросвязь. 2023. № 6. С. 20–31. DOI:10.34832/ELSV.2023.43.6.003. EDN:LWDQXI
4. Сторожук М.Н., Киричек Р.В. Обзор систем мониторинга для протокола TSN // Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А.С. Попова, посвященной Дню радио. 2022. № 1. С. 131–132. EDN:IHIAET
5. ГОСТ Р 71148-2023. Требования по построению систем синхронизации сетей связи: сетей связи с коммутацией каналов, сетей связи с коммутацией пакетов. М.: Российский институт стандартизации, 2024.
6. IEEE Std 1588™-2002. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE, 2002. DOI:10.1109/IEEESTD.2002.94144
7. IEEE Std 802.1AS™-2020. IEEE Standard for Local and Metropolitan Area Networks. Timing and Synchronization for Time-Sensitive Applications. IEEE, 2020. DOI:10.1109/IEEESTD.2020.9121845

### References


1. Bugrov D.A., Storozhuk N.L. Applying the Evolution Method to Optimize a Main Communication Network. *Electrosvyaz*. 2007;5:30–33. (in Russ.) EDN:IADPFX
2. Vlasov I.I., Novikov E.V., Ptichnikov M.M., Storozhuk N.L. *Digital Communication Networks. Cable and Fiber-Optic Lines*. Moscow: FAZIS Publ.; 2008. 500 p. (in Russ.)
3. Berezkin A.A., Parshin A.A., Parfenov D.D., Kirichek R.V. Analysis of Time-Synchronized Network Standards for Real-Time Control of Robotic Systems. *Electrosvyaz*. 2023;6:20–31. (in Russ.) DOI:10.34832/ELSV.2023.43.6.003. EDN:LWDQXI
4. Storozhuk M.N., Kirichek R.V. Overview of Monitoring Systems for the TSN Protocol. *Proceedings of the Scientific and Technical Conference of the St. Petersburg Scientific and Technical Association of the Russian Electronic Networks named after A.S. Popov, dedicated to Radio Day*. 2022;1:131–132. (in Russ.) EDN:IHIAET
5. GOST P 71148-2023. *Requirements for the construction of synchronization systems of communication networks: circuit-switched communication networks, packet-switched communication networks*. Moscow: Russian Institute of Standardization Publ.; 2024. (in Russ.)
6. IEEE Std 1588™-2002. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE; 2002. DOI:10.1109/IEEESTD.2002.94144
7. IEEE Std 802.1AS™-2020. *IEEE Standard for Local and Metropolitan Area Networks. Timing and Synchronization for Time-Sensitive Applications*. IEEE; 2020. DOI:10.1109/IEEESTD.2020.9121845

Статья поступила в редакцию 30.03.2026; одобрена после рецензирования 13.04.2026; принята к публикации 21.04.2026

The article was submitted 30.03.2026; approved after reviewing 13.04.2026; accepted for publication 21.04.2026

## Информация об авторе:

**СТОРОЖУК**  
**Николай Леонидович**

кандидат технических наук, доцент кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича  
 <https://orcid.org/0009-0005-7463-502X>

Автор сообщает об отсутствии конфликтов интересов.

The author declares no conflicts of interests.