



Гибридный метод синхронизации времени в распределенных системах

Т. М. Татарникова^а, доктор техн. наук, профессор, orcid.org/0000-0002-6419-0072, tm-tatarn@yandex.ru

Е. Д. Архипцев^б, аспирант, orcid.org/0009-0006-3001-953X

^аСанкт-Петербургский государственный университет аэрокосмического приборостроения, Б. Морская ул., 67, Санкт-Петербург, 190000, РФ

^бСанкт-Петербургский государственный электротехнический университет «ЛЭТИ», Профессора Попова ул., 5, Санкт-Петербург, 197376, РФ

Введение: решение задачи согласованности данных и процессов в условиях децентрализованной асинхронной среды является актуальной для распределенных систем, таких как глобальные облачные платформы, интернет вещей, блокчейн-инфраструктура. Существующие протоколы синхронизации основаны на предположении о симметричности временной задержки на передаче и приеме, что не соответствует действительности в условиях меняющейся сетевой нагрузки, а стохастическая природа сетевых шумов игнорируется. Указанные недостатки протоколов приводят к некорректной синхронизации времени. **Цель:** разработать гибридный метод синхронизации времени, основанный на фильтре Калмана для сглаживания сетевых шумов и логических часах для адаптации к меняющейся сетевой нагрузке. **Результаты:** показано, что существующие протоколы синхронизации распределенных систем становятся ненадежными: на синхронизацию времени влияют сетевые задержки, джиттер и аппаратные погрешности. Приведены ограничения современных подходов синхронизации времени: в работе протокола NTP синхронизации клиента и сервера не учитывается асимметрия каналов; в работе протокола RTP синхронизации локальных сетей фиксированные интервалы синхронизации приводят к шуму и накоплению ошибок; гибридизация протоколов NTP и RTP не обеспечивает динамическую адаптацию к изменяющимся условиям. Предложена гибридная модель синхронизации времени на базе комбинации фильтра Калмана и логических часов. Фильтр Калмана позволяет эффективно подавлять сетевой джиттер и компенсировать дрейф физических часов, а логические часы позволяют быстрее адаптироваться к меняющейся сетевой нагрузке. Результаты натурального эксперимента демонстрируют снижение средней задержки более чем в три раза. **Практическая значимость:** внедрение метода на базе комбинации фильтра Калмана и логических часов может стать экономичной альтернативой специализированным протоколам синхронизации в средах с нестабильной нагрузкой.

Ключевые слова — синхронизация времени, распределенные системы, физические часы, логические часы, источник точного времени, меняющаяся сетевая нагрузка, фильтр Калмана.

Для цитирования: Татарникова Т. М., Архипцев Е. Д. Гибридный метод синхронизации времени в распределенных системах. *Информационно-управляющие системы*, 2025, № 4, с. 26–35. doi:10.31799/1684-8853-2025-4-26-35, EDN: XAGEEQ

For citation: Tatarnikova T. M., Arkhipcev E. D. Hybrid method of time synchronization in distributed systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 4, pp. 26–35 (In Russian). doi:10.31799/1684-8853-2025-4-26-35, EDN: XAGEEQ

Введение

В распределенных системах, таких как глобальные облачные платформы, интернет вещей (IoT), блокчейн-инфраструктура, остается актуальной проблема согласованности процессов в условиях децентрализованной асинхронной среды [1–3].

Одной из основных задач в этом направлении является синхронизация времени, и чем выше сетевая нагрузка, тем сложнее поддерживать решение данной задачи. В теории методы синхронизации для распределенных систем должны одновременно обеспечивать следующие характеристики: метрическую точность, каузальную согласованность и устойчивость к динамическим сетевым условиям [4, 5].

Метрическая точность — это точность измерения физического времени на всех узлах распределенной системы.

Каузальная согласованность — это модель согласованности, которая отражает потенциальные причинно-следственные связи между операциями и гарантирует, что все процессы соблюдают их в общем порядке [6]. Например, в мессенджерах каузальная согласованность позволяет избежать неверного отображения сообщений и изменений, т. е. случаев, когда ответ на сообщение может появиться раньше оригинального, если узлы не синхронизированы по времени.

Устойчивость к динамическим сетевым условиям — это способность сети возвращаться в свое синхронное состояние после того, как она столкнулась с помехами.

Однако на практике, подобно теореме CAP (Consistency — согласованность, Availability — доступность, Partition tolerance — устойчивость к фрагментации), устанавливающей, что в лю-

бой реализации распределенных не транзакционных баз данных возможно обеспечить не более двух из трех свойств [7], для реализации синхронизации в распределенных системах разработчикам протоколов также необходимо определять приоритеты и находить баланс между требуемыми характеристиками.

Синхронизация в распределенных системах основана на таких протоколах, как NTP (Network Time Protocol) – синхронизация клиента и сервера и PTP (Precision Time Protocol) – синхронизация в локальных сетях [8]. С появлением распределенных систем обработки больших данных на синхронизацию времени стали влиять сетевые задержки, джиттер и аппаратные погрешности [9, 10]. Соответственно, применение этих протоколов стало ненадежным в сценариях с жесткими требованиями к синхронизации времени.

Попытки гибридизации подходов к синхронизации времени по протоколам NTP и PTP обсуждались в работах [11, 12], по результатам которых можно сделать вывод о потенциале комбинирования физических и логических меток времени, например механизме HLC (Hybrid Logical Clocks), сочетающем физическое время с логическими счетчиками. Однако предлагаемые решения не обеспечивают динамическую адаптацию к изменяющимся условиям, к тому же стохастическая природа сетевых шумов игнорируется [13].

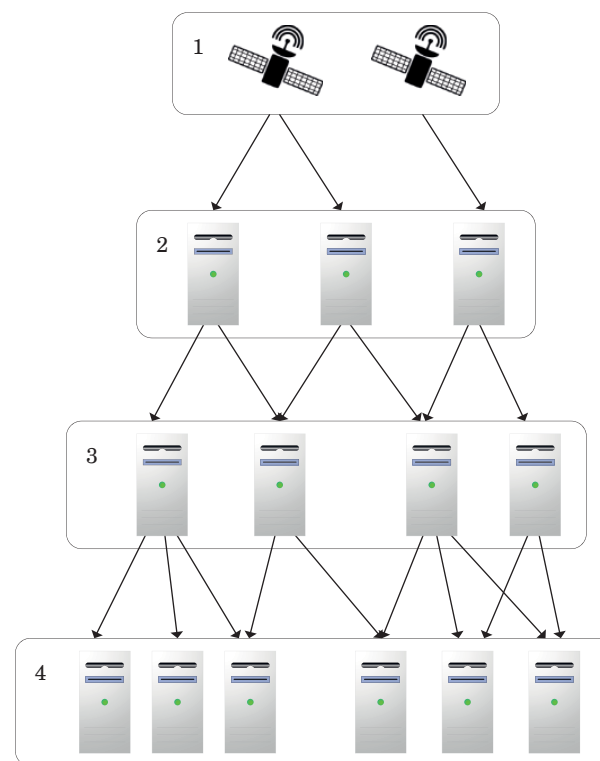
В статье предлагается гибридный метод синхронизации времени – фильтр Калмана в интеграции с логическими часами.

Известно, что фильтр Калмана успешно применяется в навигационных системах для оценки состояния через зашумленные измерения. Поэтому его интеграция с логическими часами может стать основой для создания адаптивного механизма, который не только корректирует физическое время, учитывая статистические аномалии, но и сохраняет причинно-следственную целостность событий.

Анализ ограничений современных подходов синхронизации времени

Network Time Protocol – наиболее распространенный протокол синхронизации времени, основанный на иерархии серверов точного времени Stratum (рис. 1).

Чем выше уровень сервера Stratum, тем он ближе к эталонному источнику и тем выше его метрическая точность. Серверы верхнего уровня (Stratum 1) получают время напрямую от атомных часов или GPS, используя аппаратное соединение через спутник, телефонные модемы



■ Рис. 1. Иерархия Stratum серверов

■ Fig. 1. Hierarchy of Stratum servers

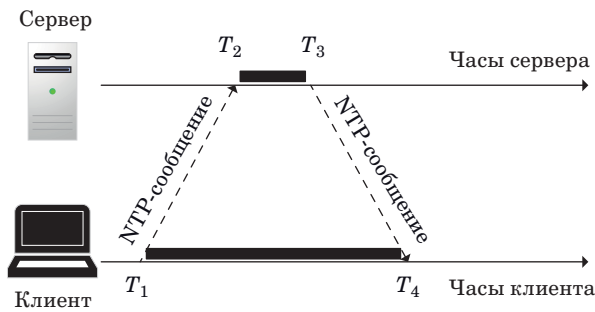
или радио. Для них характерны задержки до 1 мс при использовании локального источника – встроенных GPS-приемников и до 10 мс при использовании удаленного источника – радио или интернета. Доступ к Stratum 1 является ограниченным и предоставляется доверенным клиентам или в рамках исследовательских или представительских сетей, поскольку они имеют ограниченную пропускную способность. Серверы нижних уровней (Stratum 2, Stratum 3 и т. д.) в основном являются публичными и используют сетевое соединение. Серверы уровня Stratum 2 опрашивают несколько серверов уровня Stratum 1 для повышения точности и имеют погрешность до 50 мс [14].

В работе протокола NTP выделяются четыре основные временные точки (рис. 2): T_1 – время отправки запроса от клиента к серверу (по часам клиента); T_2 – время получения запроса сервером (по часам сервера); T_3 – время отправки ответа сервером (по часам сервера); T_4 – время получения ответа клиентом (по часам клиента).

Задержка *Delay* и смещение *Offset* вычисляются следующим образом:

$$Delay = (T_4 - T_1) - (T_3 - T_2); \quad (1)$$

$$Offset = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}, \quad (2)$$



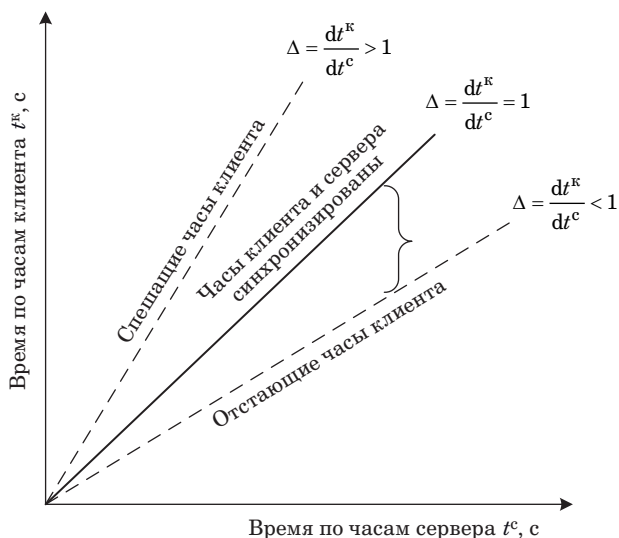
■ **Рис. 2.** Синхронизация времени между клиентом и сервером

■ **Fig. 2.** Time synchronization between client and server

где продолжительность интервалов $(T_2 - T_1)$ и $(T_4 - T_3)$ включает время распространения NTP-сообщения и разницу δ в показаниях часов клиента и сервера. На рис. 3 демонстрируются соотношения времени по часам клиента и часам сервера: рассинхронизация и дрейф частоты.

Для корректировки собственных часов клиенту необходимо добавить вычисленное значение *Offset* к текущему времени: $NewTime = CurrentTime + Offset$.

Работа протокола NTP основана на предположении о том, что сетевые задержки симметричны — имеют нормальное распределение. Если путь «Клиент → Сервер» и «Сервер → Клиент» имеют разную задержку (асимметрия каналов), то протокол NTP определит некорректное значение смещения. На практике так и получается. Например, в средах с беспроводным доступом Wi-Fi или сотовой связи задержки группируются вокруг нескольких типичных значений из-за



■ **Рис. 3.** Соотношение времени по часам клиента и сервера

■ **Fig. 3.** Time ratio between client and server clocks

нестабильности среды. Это связано с тем, что пакеты сериализуются, ожидают в очередях роутеров, тем самым создаются тяжелые хвосты в распределении сетевой задержки. Таким образом, работа протокола NTP в сетях подобного типа систематически приводит к ошибочной оценке *Offset* [15].

В протоколе PTP, известного как стандарт IEEE 1588, более высокая метрическая точность, чем в протоколе NTP, достигается за счет обязательного использования специализированного оборудования, способного на физическом уровне взаимодействия PHY (Physical Layer Protocol) предоставлять метки времени в передаваемых пакетах времени непосредственно перед моментом передачи их в канал связи. Это позволяет минимизировать джиттер, вносимый операционными системами и стеками протоколов, а также обеспечивает автоматическое восстановление при отказах.

В распределенной системе, синхронизированной по протоколу PTP, различают следующие типы узлов: гротмейстерские часы (Grandmaster) — основной источник, по которому синхронизируется время в системе; ведущие часы — источники точного времени для конечных узлов; ведомые часы — конечные узлы, на которых необходимо осуществить синхронизацию времени по протоколу PTP.

Выбор Grandmaster осуществляется динамически на основе специального механизма Best Master Clock Algorithm (BMCA). При потере узлом Grandmaster связи с физическими часами его место автоматически занимает следующий узел, претендующий на роль Grandmaster, который становится источником точного времени для конечных узлов. С этой целью все ведущие часы системы постоянно находятся в режиме прослушивания сообщений, отправленных на широковещательный адрес протокола PTP.

Основными недостатками протокола PTP являются предположение о симметричности задержек в направлениях «Ведущий → Ведомый» и «Ведомый → Ведущий» и фиксированный интервал синхронизации. Правда, в последней версии стандарта в протокол введены компенсации асимметричных задержек [16, 17] через калибровку смещений, но для этого необходимо использовать сторонние протоколы, оборудование и т. д. Фиксированные интервалы синхронизации в сценариях с резким изменением сетевой нагрузки, например flash-трафик в дата-центре, приводят к «раскачиванию» часов: частые коррекции вводят высокочастотный шум, а редкие вызывают накопление ошибок. Также немаловажным является отсутствие встроенной криптографической защиты в протоколе PTP, что дает возможность злоумышленнику искусствен-

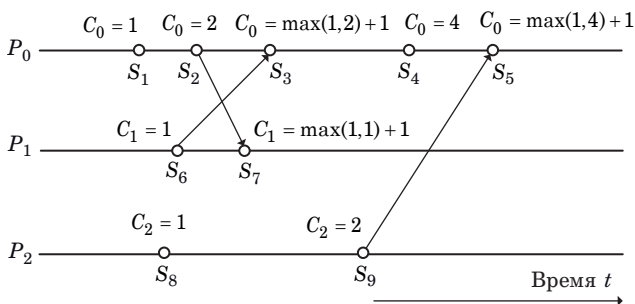
но увеличивать время доставки пакетов или фальсифицировать сообщения для назначения ложного Grandmaster. Тем не менее наличие механизма компенсации задержек является основным преимуществом протокола РТР, что делает его незаменимым в промышленных системах управления, телекоммуникациях и высокочастотном трейдинге, где погрешность синхронизации свыше 1 мс неприемлема [18].

Механизм HLC синхронизации времени в распределенных системах сочетает физическое время с логическими часами Лэмпорта и векторными часами. Механизм обеспечивает не только метрическую точность, но и частично каузальную согласованность.

Логические часы Лэмпорта представляют собой счетчик определения порядка событий. Работа счетчика описывается следующей схемой (рис. 4):

- монотонное увеличение счетчика C при выполнении внутренних событий S каждого процесса P ;
- значение счетчика i -го процесса передается j -му процессу в пакете времени;
- при получении j -м процессом пакета времени значение счетчика j -го процесса устанавливается как максимальное из текущего значения собственного счетчика и полученного значения счетчика i -го процесса и увеличивается на 1.

Порядок изменения событий подчиняется правилу: если событие S_1 произошло до S_2 , то временная метка S_1 меньше S_2 : $TimeStamp(S_1) < TimeStamp(S_2)$. Однако обратное утверждение неверно, поскольку для событий без детерминистской связи нельзя выстроить временную зависимость. Например, если событие S_1 в процессе P_i отправляет сообщение, а в процессе P_j событие S_2 получает это сообщение, то можно сказать, что событие S_1 произошло раньше события S_2 . Предположим, что до того, как процесс P_j получил сообщение о событии S_2 , в P_j произошло событие S_3 , т. е. S_3 произошло раньше S_2 , а было отправлено сообщение процессу P_i , а затем процесс P_i получил сообщение



■ Рис. 4. Часы Лэмпорта
■ Fig. 4. Lamport clock

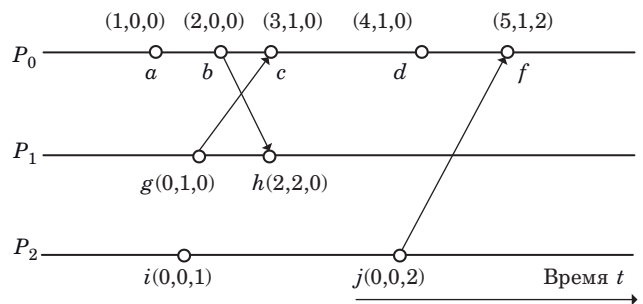
о событии S_4 после события S_1 , т. е. событие S_3 , как и событие S_1 , произошло раньше события S_4 . Основываясь на алгоритме Лэмпорта, невозможно определить какие-либо связи между событиями S_1 и S_3 . Все, что известно, это $S_1 \rightarrow S_2$, и $S_3 \rightarrow S_4$, и $S_1 \rightarrow S_4$, и $S_3 \rightarrow S_2$, но непонятно, $S_1 \rightarrow S_3$ или $S_3 \rightarrow S_1$. Таким образом, логические часы Лэмпорта способны обеспечить только частичный порядок, т. е. они указывают на причинно-следственную связь между некоторыми, но не всеми событиями. Это может привести к неопределенности при определении точного порядка событий.

Векторные часы — это расширение возможностей логических часов Лэмпорта. Каждый процесс P_i , $i = \overline{1, N}$ в системе поддерживает вектор логических часов $(1, \dots, i, \dots, N)$, где i -й элемент вектора представляет состояние часов i -го активного процесса. Приведенный на рис. 5 пример иллюстрирует механизм векторных часов для трех процессов.

Работа векторных часов описывается следующей схемой:

- все часы всегда установлены на ноль;
- каждый раз, когда в i -м процессе происходит внутреннее событие, значение логических часов i -го процесса в векторе увеличивается на 1;
- каждый раз, когда i -й процесс отправляет сообщение j -му процессу, значение логических часов i -го процесса в векторе увеличивается на 1;
- каждый раз, когда j -й процесс получает сообщение, он увеличивает значение логических часов i -го процесса в векторе на 1;
- каждый элемент вектора часов обновляется путем взятия максимального значения как из собственного вектора, так и из вектора в полученном сообщении для каждого элемента.

Таким образом, сравнивая векторные часы, система может определить, произошло ли событие на одном узле до, после или одновременно с событием на другом узле, что позволяет эффективно разрешать конфликты и обеспечивать каузальную согласованность [19, 20].



■ Рис. 5. Векторные часы
■ Fig. 5. Vector clock

Основными ограничениями векторных часов, которые влияют на их применимость в современных распределенных системах, являются [21]:

– требование хранения N часов в системе с N узлами, что приводит к высоким затратам на коммуникацию;

– сложная реализация векторных часов в сценариях, когда узлы часто присоединяются и покидают систему или при частых сбоях сети;

– увеличение объема передаваемых сообщений из-за необходимости периодического обмена векторными часами между процессами, что становится проблемой для систем с ограниченной пропускной способностью или приложений, чувствительных к задержкам.

При реализации механизма HLC каждый узел в системе поддерживает гибридную метку времени, состоящую из двух компонент:

– физического времени (pt) – локального времени узла, синхронизированного с NTP или другим протоколом;

– логического счетчика (lc) – счетчика, который необходим для регистрации локальных событий или для разрешения конфликтов.

При получении нового события на узле генерируется новая метка по следующему правилу:

$$\begin{aligned} pt_{new} &= \max(pt_{old}, pt_{local}); \\ lc_{new} &= lc_{old} + 1 \text{ при } pt_{local} > pt_{old}, \end{aligned} \quad (3)$$

где pt_{local} – текущее физическое время узла, полученное из пакета NTP; (pt_{old}, lc_{old}) – текущая метка HLC до обновления.

Перед отправкой сообщения генерируется новая метка HLC, как при локальном событии. На узел отправляется сообщение с меткой (pt_{new}, lc_{new}). При получении сообщения на узле вычисляется новая метка:

$$\begin{aligned} pt_{new} &= \max(pt_{current}, pt_{msg}, pt_{local}); \\ lc_{new} &= lc_{current} + 1, lc_{msg} + 1 \text{ при } pt_{msg} = pt_{new}, \end{aligned} \quad (4)$$

где ($pt_{current}, lc_{current}$) – текущая метка HLC на принимающем узле; (pt_{msg}, lc_{msg}) – физическая метка, извлекаемая из полученного сообщения msg .

Таким образом, механизм HLC обеспечивает синхронизацию с физическим временем и каузальную согласованность. Но поскольку механизм HLC основан на физическом времени, которое может быть неточным из-за сбоя NTP или аппаратных ошибок, то метрическая точность не гарантируется.

Как показывает анализ, ни один из существующих протоколов – NTP, RTP, HLC – не решает эту задачу синхронизации в полной мере. Однако комбинация их сильных сторон с адаптивными алгоритмами позволяет создавать гибридные модели, способные компенсировать индивидуаль-

ные недостатки. Одно из таких решений – комбинация фильтра Калмана и логических часов.

Комбинация фильтра Калмана и логических часов

Для повышения точности синхронизации времени применим фильтр Калмана – адаптивный алгоритм, позволяющий минимизировать шум измерений и предсказывать истинное значение временных меток. Данный метод основан на статистической обработке поступающих измерений и динамическом обновлении оценок с учетом априорной модели изменений времени.

Применение фильтра Калмана позволяет:

– сглаживать вариации задержек передачи данных в NTP, минимизируя ошибки синхронизации;

– корректировать временные метки RTP при резких изменениях мастер-часов;

– обеспечивать устойчивость HLC к неточностям в измерениях физического времени.

Фильтр Калмана опирается на две ключевые модели.

Модель процесса прогнозирования состояния

$$\mathbf{x}_t = \mathbf{F} \times \mathbf{x}_{t-1} + \mathbf{B} \times u_{t-1} + \mathbf{w}_t, \quad (5)$$

где $\mathbf{x}_t$ – вектор состояния (смещение часов *Offset* и дрейф частоты *Drift*) в текущий момент времени t ; $\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ – матрица перехода (модель эволюции состояния), Δt – интервал дискретизации; $\mathbf{B} = \begin{bmatrix} \Delta t \\ 1 \end{bmatrix}$ – матрица-столбец применения

управляющего воздействия; u_{t-1} – управляющее воздействие, по сути значение *Offset* – коррекция от NTP в предыдущий момент времени ($t - 1$); $\mathbf{w}_t$ – вектор шума процесса (белый гауссов шум) в текущий момент времени t .

Модель наблюдения, которая связывает состояние с измерениями:

$$\mathbf{z}_t = \mathbf{H} \times \mathbf{x}_t + \mathbf{v}_t, \quad (6)$$

где $\mathbf{z}_t$ – измерение (разница между показаниями часов клиента и сервера); $\mathbf{H}$ – матрица наблюдений; $\mathbf{x}_t$ – вектор состояния в текущий момент времени t ; $\mathbf{v}_t$ – шум измерений (джиттер задержки *Delay*).

Шум процесса представляется ковариационной матрицей

$$\mathbf{Q}_t = \begin{bmatrix} \mathbf{w}_t \times \mathbf{w}_t^T \end{bmatrix} = \begin{bmatrix} \sigma_{Offset}^2 & 0 \\ 0 & \sigma_{Drift}^2 \end{bmatrix}, \quad (7)$$

где σ_{Offset}^2 , σ_{Drift}^2 — дисперсии шумов смещения и дрейфа частоты.

Шум измерений представляется ковариационной матрицей

$$\mathbf{R}_t = [\mathbf{v}_t \times \mathbf{v}_t^T]. \quad (8)$$

Фильтр Калмана реализует итеративный процесс, состоящий из двух основных этапов: прогноза — предсказания состояния системы (5) на основе модели наблюдения (6) и коррекции — уточнения предсказания с использованием новых измерений. Эти этапы образуют замкнутый цикл, позволяющий адаптивно балансировать между доверием к модели и поступающим данным.

Для коррекции предсказания применяются следующие выражения:

$$\mathbf{K}_t = \mathbf{P}_t \times \mathbf{H}^T \times (\mathbf{H} \times \mathbf{P}_t \times \mathbf{H}^T + \mathbf{R}_t)^{-1}; \quad (9)$$

$$\mathbf{x}_t = \mathbf{x}_{\bar{t}} + \mathbf{K}_t \times (\mathbf{z}_t - \mathbf{H} \times \mathbf{x}_{\bar{t}}); \quad (10)$$

$$\mathbf{P}_t = [\mathbf{P}_t - \mathbf{K}_t \times \mathbf{H}^T \times \mathbf{P}_t \times \mathbf{H}^T + \mathbf{K}_t \times \mathbf{R}_t \times \mathbf{K}_t^T], \quad (11)$$

где $\mathbf{K}_t$ — коэффициент Калмана (коэффициент усиления фильтра); $\mathbf{P}_t$ — матрица ковариации предсказываемой ошибки оценивания; $\mathbf{x}_{\bar{t}}$ — вектор предсказываемой оценки состояния; $\mathbf{P}_t$ — матрица ковариации ошибки оценивания.

Алгоритм работы фильтра Калмана.

Шаг 1. Инициализация фильтра Калмана: $\mathbf{x}_0$ — начального значения вектора предсказываемой оценки состояния и $\mathbf{P}_0$ — матрицы ковариации предсказываемой ошибки оценивания.

Шаг 2. Измерение $\mathbf{z}_t$.

Шаг 3. Вычисление коэффициента Калмана по формуле (9).

Шаг 4. Коррекция вектора состояний по формуле (10).

Шаг 5. Коррекция матрицы ковариации ошибки оценивания по формуле (11).

Шаг 6. Предсказание следующего шага итерации:

$$\mathbf{x}_{\bar{t}} = \mathbf{F} \times \mathbf{x}_{t-1} + \mathbf{B} \times \mathbf{u}_{t-1};$$

$$\mathbf{P}_{\bar{t}} = \mathbf{F} \times \mathbf{P}_t \times \mathbf{F}^T + \mathbf{Q}.$$

Шаг 7. Переход на шаг 2 с повторением процедуры фильтрации Калмана для каждого нового измерения.

Результаты натурального эксперимента

Для эксперимента было создано четыре виртуальные вычислительные машины (ВВМ) с соответствующими методами синхронизации

времени: по протоколу NTP, по протоколу RTP, HLC с часами Лэмпорта и HLC с векторными часами. На каждом сервере генерировалась нагрузка в виде постоянных HTTP-запросов и применялся гибридный метод интеграции фильтра Калмана с логическими часами. Параметры фильтра Калмана $\mathbf{Q}$ подбирались экспериментальным путем:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,01 \end{bmatrix}. \quad (12)$$

Замеры *Offset* на ВВМ#1 выполнены с синхронизацией времени по протоколу NTP, так называемый сырой *Offset*, и с синхронизацией гибридным методом интеграции фильтра Калмана с логическими часами (рис. 6, а). Гистограмма изменения джиттера задержки *Delay* на том же самом интервале замеров представлена на рис. 6, б.

Продолжительность эксперимента составила 6 ч, но на графиках приведены замеры на 100 с, отражающие наиболее общую картину результатов эксперимента.

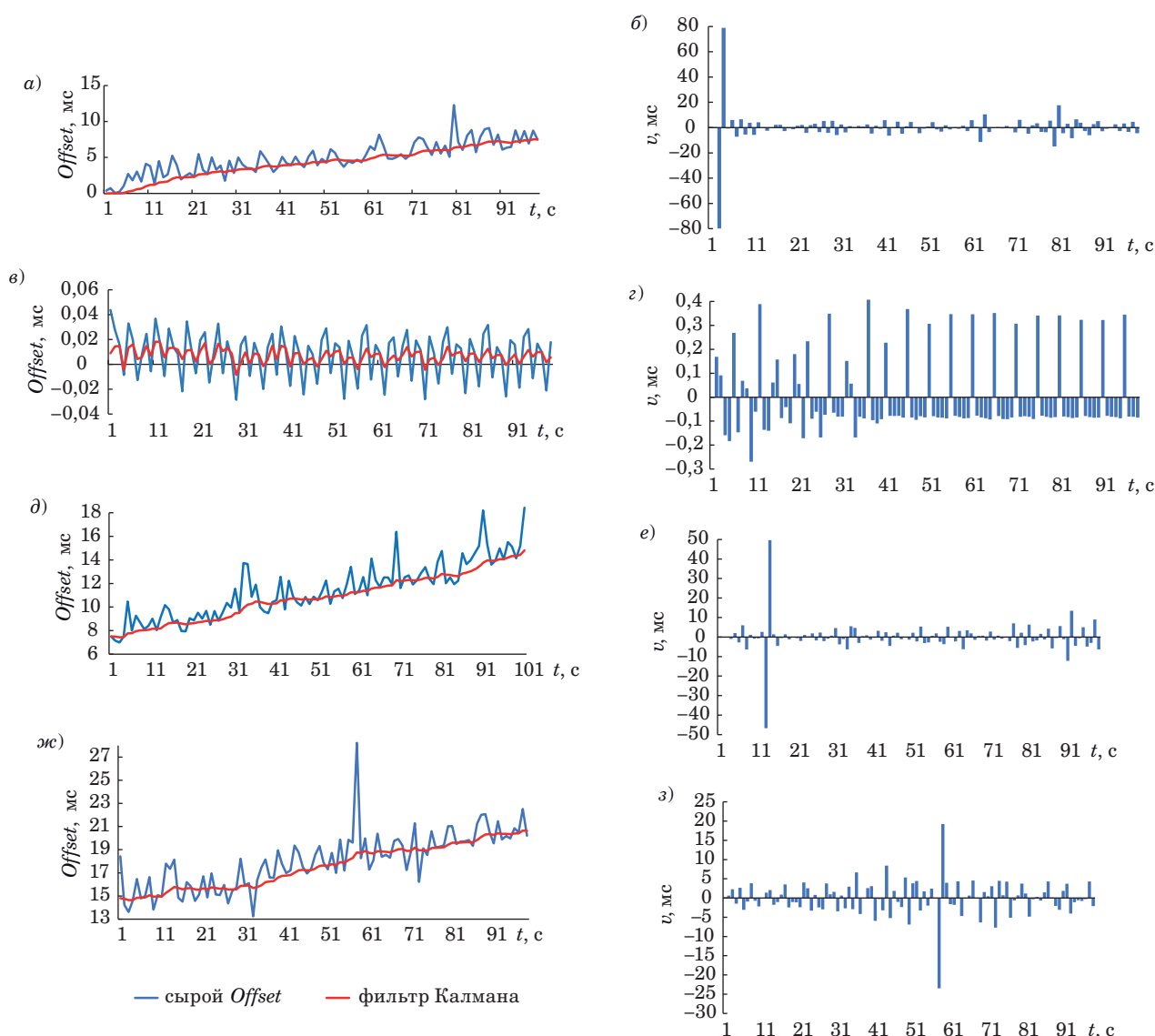
Без использования фильтра Калмана среднеквадратическое отклонение джиттера составило 12,23 мс, с фильтром Калмана оно стремится к нулю — по этой причине приводить гистограмму изменения джиттера после фильтра Калмана не имеет смысла. Таким образом, при использовании гибридного метода интеграции фильтра Калмана с логическими часами удалось снизить среднеквадратическое отклонение джиттера на 74 %, а максимальный *Offset* с 12,3 до 7,5 мс.

С синхронизацией времени по протоколу RTP и с синхронизацией гибридным методом интеграции фильтра Калмана с логическими часами проведены замеры *Offset* на ВВМ#2 (рис. 6, в). Гистограмма изменения джиттера на том же самом интервале замеров представлена на рис. 6, г.

Без использования фильтра Калмана среднеквадратическое отклонение джиттера по протоколу RTP составило 0,17 мс, с фильтром Калмана оно стремится к нулю. При использовании гибридного метода интеграции фильтра Калмана с логическими часами удалось снизить среднеквадратическое отклонение джиттера на 58 %, а максимальный *Offset* — с 0,044 до 0,018 мс.

Результаты замеров *Offset* на ВВМ#3 с синхронизацией времени по механизму HLC с часами Лэмпорта и с синхронизацией гибридным методом интеграции фильтра Калмана с логическими часами показаны на рис. 6, д. Гистограмма изменения джиттера на том же самом интервале замеров представлена на рис. 6, е.

Без использования фильтра Калмана среднеквадратическое отклонение джиттера составило 7,79 мс, с фильтром Калмана оно стремится к нулю.



■ **Рис. 6.** Графики замеров *Offset* (слева) и гистограмма изменения джиттера (справа) на BVM#1 (а, б); BVM#2 (в, з); BVM#3 (д, е); BVM#4 (ж, з)

■ **Fig. 6.** *Offset* measurement graphs (left) and jitter change histogram (right) on virtual machine #1 (а, б); virtual machine #2 (в, з); virtual machine #3 (д, е); virtual machine #4 (ж, з)

лю. При использовании гибридного метода интеграции фильтра Калмана с логическими часами удалось снизить среднеквадратическое отклонение джиттера на 52 %, а максимальный *Offset* — с 18,43 до 14,8 мс.

Графики замеров *Offset* на BVM#4 с синхронизацией времени по механизму NTP с векторными часами и с синхронизацией гибридным методом интеграции фильтра Калмана с логическими часами приведены на рис. 6, ж. Гистограмма изменения джиттера на том же самом интервале замеров представлена на рис. 6, з.

Без использования фильтра Калмана среднеквадратическое отклонение джиттера составило 4,36 мс, с фильтром Калмана оно стремится к ну-

лю. При использовании гибридного метода интеграции фильтра Калмана с логическими часами удалось снизить среднеквадратическое отклонение джиттера на 47 %, а максимальный *Offset* — с 28,23 до 20,68 мс.

Заключение

Интеграция фильтра Калмана в классические протоколы синхронизации времени, такие как NTP или PTP, обеспечивает существенное повышение точности синхронизации часов клиента и сервера в распределенных системах за счет принципиально иного подхода к обработке

неопределенностей, присущих сетевым взаимодействиям. Ключевое преимущество заключается в способности алгоритма Калмана декомпонировать системные шумы и полезный сигнал на основе стохастической модели, что невозможно в рамках детерминированных методов коррекции, используемых стандартными протоколами.

Таким образом, гибридный метод, сочетающий применение фильтра Калмана с логическими часами, преодолевает фундаментальное ограничение протоколов синхронизации — зависимость точности от стационарности сетевой среды. Это возможно благодаря наличию адаптивной стохастической обратной связи — коррекции состояния системы на основе модели наблюдения.

Эксперимент подтверждает, что фильтр Калмана — эффективный инструмент для борьбы с джиттером в распределенных системах, особенно в NTP-сетях. При помощи него удалось снизить среднеквадратическое отклонение джиттера и *Offset* для всех рассмотренных протоколов и механизмов синхронизации времени.

Внедрение метода в облачные инфраструктуры может стать решением для задач, где важна балансировка между точностью и каузальной согласованностью.

Дальнейшие исследования целесообразно направить на создание самообучающихся алгоритмов для динамических сред.

Литература

1. Balakrishnan K., Dhanalakshmi R., Bam Bahadur Sinha, Gopalakrishnan R. Clock synchronization in industrial Internet of Things and potential works in precision time protocol: Review, challenges and future directions. *International Journal of Cognitive Computing in Engineering*, 2023, vol. 4, pp. 205–219. <https://doi.org/10.1016/j.ijcce.2023.06.001>
2. Vilaplana J., Solsona F., Teixidó I., Mateo Fornes J. A queuing theory model for cloud computing. *The Journal of Supercomputing*, 2014, vol. 69, pp. 492–507. doi:10.1007/s11227-014-1177-y
3. Özsu M. T., Valduriez P. *Principles of Distributed Database Systems*. Springer Cham, 2020. 676 p.
4. Поляков В. Б., Аксёнов Н. В., Сергеев М. В. Средства обеспечения временной синхронизации данных и процессов в системах интернет вещей на уровне устройств и их классификация. *Альманах современной метрологии*, 2019, № 4 (20), с. 40–53. EDN: PFXPMY
5. Tatarnikova T. M., Arkhiptsev E. M. Designing fault-tolerant systems with micro-service architecture. *XXVII International Conference on Soft Computing and Measurements (SCM)*, Saint-Petersburg, Russian Federation, 2024, pp. 348–351. doi:10.1109/SCM62608.2024.10554143
6. Elbushra M. M., Lindström J. Causal consistent databases. *Open Journal of Databases*, 2015, vol. 2, no. 1, pp. 17–35.
7. Татарникова Т. М., Архипцев Е. Д., Кармановский Н. С. Определение размера кластера и числа реплик высоконагруженных информационных систем. *Известия высших учебных заведений. Приборостроение*, 2023, т. 66, № 8, с. 646–651. doi:10.17586/0021-3454-2023-66-8-646-651
8. Рыжков А. В., Колтунов М. Н., Насонов А. Ю., Шварц М. Л. Проблемы сличения шкал времени в пакетных сетях электросвязи. *Т-Сотт: Телекоммуникации и транспорт*, 2017, т. 11, № 11, с. 10–17.
9. Фунг В. К., Богатырев В. А. Задержки и надежность обслуживания запросов в виртуальном компьютерном кластере. *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*, 2024, № 68, с. 48–58. doi:10.17223/19988605/68/5
10. Воронин В. В., Землянухина А. В., Репецкий А. А. Уязвимость NTP-протокола: анализ происшествий и стратегия безопасности. *Вестник ТОГУ*, 2025, № 1(76), с. 49–58. <https://doi.org/10.38161/1996-3440-2025-1-49-58>
11. Mallela C., Tholu K., Bordogna M. Timing models for PTP in Ethernet networks. *2017 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, 2017, pp. 1–6.
12. Kulkarni S. S., Demirbas M., Madeppa D., Avva B., Leone M. Logical physical clocks. *Principles of Distributed Systems: 18th International Conference*, 2014, pp. 17–32.
13. Zhou K., Wang X., Cui S. Time synchronization in Industrial Internet of Things: A survey. *IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2024, vol. 7, pp. 506–510.
14. Rinaldi S., Giustina D. D., Ferrari P., Flammini A., Sisinni E. Time synchronization over heterogeneous network for smart grid application: Design and characterization of a real case. *Ad Hoc Networks*, 2016, vol. 50, pp. 41–57. <https://doi.org/10.1016/j.adhoc.2016.04.001>
15. Губанов Н. Г., Рагузин А. С. Анализ и моделирование протоколов синхронизации в информационной системе региональной электросетевой компании. *Вестник Самарского государственного технического университета. Серия: Технические науки*, 2016, № 1(49), с. 7–14.
16. Exel R. Mitigation of asymmetric link delays in IEEE 1588 clock synchronization systems. *IEEE Communications Letters*, 2014, vol. 18, no. 3, pp. 507–510.

17. Батенков К. А., Цицин Е. А. Подстройка параметров времени для сетевых устройств с помощью протокола синхронизации РТР (Precision Time Protocol). *Инфокоммуникационные технологии: актуальные вопросы цифровой экономики: сборник научных трудов I Международной научно-практической конференции*, Екатеринбург, 17–18 февраля 2021 г. Екатеринбург, УрТИСИ, 2021, с. 90–95.
18. Мазуренко Д. К. Сетевые решения построения пакетной сети распределения сигналов единого точного времени. *Т-Сотт: Телекоммуникации и транспорт*, 2015, т. 9, № 4, с. 67–71.
19. Finkenzeller A., Butowski O., Regnath E., Hamad M., Steinhorst S. PTPsec: Securing the Preci-

sion Time Protocol against time delay attacks using cyclic path asymmetry analysis. *IEEE INFOCOM 2024 – IEEE Conference on Computer Communications*, 2024, pp. 461–470.

20. Bogatyrev V. A., Derkach A. N., Aleksankov S. M. The model of reliability of dulated real-time computers for cyber-physical systems. *Studies in Systems, Decision and Control*, 2020, vol. 260, pp. 11–21. doi:10.1007/978-3-030-32648-7_2, EDN: ZXAHKJ
21. Tatarnikova T. M., Arkhiptsev E. D. Determine the number of distributed big data storage replicas. *XXVI International Conference on Soft Computing and Measurements (SCM)*, 2023, pp. 223–226. doi:10.1109/SCM58628.2023.10159087

UDC 004.032.26

doi:10.31799/1684-8853-2025-4-26-35

EDN: XAGEEQ

Hybrid method of time synchronization in distributed systems

T. M. Tatarnikova^a, Dr. Sc., Tech., Professor, orcid.org/0000-0002-6419-0072, tm-tatarn@yandex.ru

E. D. Arkhiptsev^b, Post-Graduate Student, orcid.org/0009-0006-3001-953X

^aSaint-Petersburg State University of Aerospace Instrumentation, 67, B. Morskaya St., 190000, Saint-Petersburg, Russian Federation

^bSaint-Petersburg Electrotechnical University «LETI», 5, Prof. Popov St., 197376, Saint-Petersburg, Russian Federation

Introduction: The solution to the problem of data and process consistency in a decentralized asynchronous environment is relevant for distributed systems such as global cloud platforms, the Internet of Things, and blockchain infrastructure. Existing synchronization protocols are based on the assumption of symmetrical time delays during transmission and reception, which is not true in the conditions of changing network load in addition, the stochastic nature of network noise is ignored. These shortcomings of the protocols lead to incorrect time synchronization. **Purpose:** To develop a hybrid time synchronization method based on the Kalman filter to smooth out network noise and on a logical clock for the system to adapt to the changing network load. **Results:** We demonstrate that the existing distributed systems synchronization protocols are becoming unreliable as time synchronization is affected by network delays, jitter, and hardware errors. We highlight the limitations of modern time synchronization: the NTP protocol for synchronizing Clients and Servers does not consider channel asymmetry; fixed synchronization intervals in the PTP protocol for synchronizing local networks lead to noise and error accumulation; hybridization of the NTP and PTP protocols does not provide dynamic adaptation to changing conditions. We propose a hybrid time synchronization model based on the combination of the Kalman filter and a logical clock. The Kalman filter makes it possible to effectively suppress network jitter and to compensate for physical clock drift, while logical clocks enable a faster adaptation to changing network loads. The results of a full-scale experiment demonstrate a more than threefold reduction in average delay. **Practical relevance:** The implementation of the method based on the combination of the Kalman filter and a logical clock can become a cost-effective alternative to specialized synchronization protocols in environments with unstable loads.

Keywords — time synchronization, distributed systems, physical clock, logical clock, precise time source, changing load, Kalman filter.

For citation: Tatarnikova T. M., Arkhiptsev E. D. Hybrid method of time synchronization in distributed systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 4, pp. 26–35 (In Russian). doi:10.31799/1684-8853-2025-4-26-35, EDN: XAGEEQ

References

1. Balakrishnan K., Dhanalakshmi R., Bam Bahadur Sinha, Gopalakrishnan R. Clock synchronization in industrial Internet of Things and potential works in precision time protocol: Review, challenges and future directions. *International Journal of Cognitive Computing in Engineering*, 2023, vol. 4, pp. 205–219. https://doi.org/10.1016/j.ijcce.2023.06.001
2. Vilaplana J., Solsona F., Teixido I., Mateo Fornes J. A queuing theory model for cloud computing. *The Journal of Supercomputing*, 2014, vol. 69, pp. 492–507. doi:10.1007/s11227-014-1177-y
3. Özsu M. T., Valduriez P. *Principles of Distributed Database Systems*. Springer Cham, 2020. 676 p.
4. Polyakov V. B., Aksyonov N. V., Sergeev M. V. The means of providing time synchronization of data and processes in the Internet of Things at the level of devices and their classification. *Al'manac of Modern Metrology*, 2019, no. 4 (20), pp. 40–53 (In Russian). EDN: PFXPMY
5. Tatarnikova T. M., Arkhiptsev E. M. Designing fault-tolerant systems with micro-service architecture. *XXVII International Conference on Soft Computing and Measurements (SCM)*, Saint-Petersburg, Russian Federation, 2024, pp. 348–351 (In Russian). doi:10.1109/SCM62608.2024.10554143
6. Elbushra M. M., Lindström J. Causal consistent databases. *Open Journal of Databases*, 2015, vol. 2, no. 1, pp. 17–35.
7. Tatarnikova T. M., Arkhiptsev E. D., Karmanovskiy N. S. Determining the cluster size and the number of replicas of highly loaded information systems. *Journal of Instrument Engineering*, 2023, vol. 66, no. 8, pp. 646–651 (In Russian).
8. Ryzhkov A. V., Koltunov M. N., Nasonov A. Y., Schwartz M. L. Problems of comparing time scales in packet telecommunication networks. *T-Comm*, 2017, vol. 11, no. 11, pp. 10–17 (In Russian).
9. Phung V. Q., Bogatyrev V. A. Delays and reliability of request servicing in a virtual computer cluster. *Tomsk State*

- University Journal of Control and Computer Science*, 2024, vol. 68, pp. 48–58 (In Russian). doi:10.17223/19988605/68/5
10. Voronin V. V., Zemlyanukhina A. V., Repetskiy A. A. NTP protocol vulnerabilities: Incident analysis and security strategy. *Bulletin of Pacific National University*, 2025, no. 1(76), pp. 49–58 (In Russian). <https://doi.org/10.38161/1996-3440-2025-1-49-58>
 11. Mallela C., Tholu K., Bordogna M. Timing models for PTP in Ethernet networks. *2017 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, 2017, pp. 1–6.
 12. Kulkarni S. S., Demirbas M., Madeppa D., Avva B., Leone M. Logical physical clocks. *Principles of Distributed Systems: 18th International Conference*, 2014, pp. 17–32.
 13. Zhou K., Wang X., Cui S. Time synchronization in industrial Internet of Things: A survey. *IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2024, vol. 7, pp. 506–510.
 14. Rinaldi S., Giustina D. D., Ferrari P., Flammini A., Sisinni E. Time synchronization over heterogeneous network for smart grid application: Design and characterization of a real case. *Ad Hoc Networks*, 2016, vol. 50, pp. 41–57. <https://doi.org/10.1016/j.adhoc.2016.04.001/>
 15. Gubanov N. G., Raguzin A. S. Analysis and modeling of the existing protocols used in time synchronization of the regional power-grid information systems. *Vestnik Samarskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Tekhnicheskie nauki*, 2016, no. 1(49), pp. 7–14 (In Russian).
 16. Exel R. Mitigation of asymmetric link delays in IEEE 1588 clock synchronization systems. *IEEE Communications Letters*, 2014, vol. 18, no. 3, pp. 507–510.
 17. Batenkov K. A., Tsitsin E. A. Adjusting time settings for networked devices with ptp synchronization protocol. *Sbornik nauchnykh trudov I Mezhdunarodnoy nauchno-prakticheskoy konferentsii "Infokommunikatsionnyye tekhnologii: aktual'nyye voprosy tsifrovoy ekonomiki"* [Proc. I Intern. Scient. and Pract. Conf. "Infocommunication technologies: current issues of the digital economy"]. Ekaterinburg, 2021, pp. 90–95 (In Russian).
 18. Mazurenko D. K. Network reference model of the UTC distribution for the packet networks. *T-Comm*, 2015, vol. 9, no. 4, pp. 67–71 (In Russian).
 19. Finkenzeller A., Butowski O., Regnath E., Hamad M., Steinhorst S. PTPsec: Securing the Precision Time Protocol against time delay attacks using cyclic path asymmetry analysis. *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, 2024, pp. 461–470.
 20. Bogatyrev V. A., Derkach A. N., Aleksankov S. M. The model of reliability of dublated real-time computers for cyber-physical systems. *Studies in Systems, Decision and Control*, 2020, vol. 260, pp. 11–21. doi:10.1007/978-3-030-32648-7_2, EDN: ZXAHKJ
 21. Tatarnikova T. M., Arkhiptsev E. D. Determine the number of distributed big data storage replicas. *XXVI International Conference on Soft Computing and Measurements (SCM)*, 2023, pp. 223–226. doi:10.1109/SCM58628.2023.10159087