

ПРОФИЛЬ PTP ДЛЯ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

## Ноябрь 2022 г.

Резюме

Данный документ определяет профиль PTP для поддержки чувствительных ко времени приложений в среде центра обработки данных. Документ разработан в рамках сообщества Open Compute Project (OCP) Timing Appliances Project [1]. Профиль PTP основан на стандарте IEEE Std 1588 TM-2019 [2]. При необходимости в профиле также используются ссылки и информация из других профилей PTP или других отраслевых спецификаций. Документ содержит набор требований к реализации, развертыванию и эксплуатации устройств синхронизации в ЦОД. Устройство синхронизации — это элемент, поддерживающий PTP, например коммутатор/маршрутизатор, сервер времени, сетевая карта, программный модуль, плата синхронизации, устройство мониторинга и т.д.

# Оглавление

1. [Введение 5](#_bookmark0)
2. [Терминология 5](#_bookmark1)
3. [Определение профиля PTP 5](#_bookmark2)
4. [Эталонная модель 6](#_bookmark3)
   1. [Модель 1 - Прозрачная модель часов 6](#_bookmark4)
5. [Модель 1 - Дополнительные требования 8](#_bookmark5)
6. [Профиль PTP 8](#_bookmark6)
   1. [Идентификатор профиля 8](#_bookmark7)
   2. [Типы часов 8](#_bookmark8)
   3. [Типы сообщений 9](#_bookmark9)
   4. [Требуемые, разрешенные или запрещенные транспортные механизмы 9](#_bookmark10)
   5. [Идентификация часов 9](#_bookmark11)
   6. [Механизм измерения задержки на пути 10](#_bookmark12)
   7. [Класс обслуживания 10](#_bookmark13)
   8. [Безопасность PTP 10](#_bookmark14)
   9. [Изоляция профиля и доменное число 10](#_bookmark15)
   10. [Одноступенчатое и двухступенчатое управление 11](#_bookmark16)
   11. [Конечная ТС с двухступенчатым режимом работы 11](#_bookmark17)
   12. [Скорость передачи сообщений PTP 12](#_bookmark18)
   13. [Интервал между сообщениями PTP 12](#_bookmark19)
   14. [Лучший алгоритм и атрибуты часов 13](#_bookmark20)
   15. [Одноадресная связь 18](#_bookmark21)
       1. [Одноадресное обнаружение 18](#_bookmark22)
       2. [Переговоры по одноадресной рассылке 19](#_bookmark23)
       3. [Сценарий "активный - резервный 22](#_bookmark24)
       4. [Активно-активный сценарий 22](#_bookmark25)
   16. [Сообщения управления PTP 23](#_bookmark26)
   17. [Пределы сети и бюджет ошибки для модели 1 23](#_bookmark27)
7. [Ссылки 25](#_bookmark28)
8. [Версия документа 26](#_bookmark29)
9. [Вкладчики 26](#_bookmark30)
10. [Лицензия 27](#_bookmark31)
11. [Приложение I. Примеры конфигурации Active-Standby 28](#_bookmark32)
12. [Приложение II. Примеры конфигурации Active-Active 31](#_bookmark33)
    1. [Введение 31](#_bookmark34)
    2. [Иллюстрация 31](#_bookmark35)

# 1 Введение

Время является ключевым элементом для достижения максимальной эффективности в распределенной системе. Производительность распределенной системы частично зависит от уровня синхронизации между ее элементами. Ряд отраслей, таких как телекоммуникационная, энергетическая, промышленная, автомобильная, профессиональная аудио- и видео, осознали необходимость высокоточного и надежного распределения и синхронизации времени в пакетных сетях. Хотя сценарии использования для каждой из этих отраслей различны, все они имеют одну общую черту - синхронизацию времени. Каждый сценарий использования определяет набор требований и конфигураций, которые указываются в «профиле PTP». В данном документе определен профиль PTP для удовлетворения потребностей приложений, установленных в ЦОД, чувствительных к времени, сетевой инфраструктуры ЦОД и использования синхронизированных часов [3]. Профиль определяет набор функций и значений атрибутов PTP, применимых к экземпляру PTP, который работает в одном устройстве (например, коммутаторе, маршрутизаторе, сервере) и в пределах только одного домена PTP. Кроме того, в данной спецификации рассматриваются дополнительные требования и случаи использования, которые не входят в определение профиля PTP.

# 2 Терминология

Комитет IEEE 1588 работает над поправкой, рекомендующей альтернативную терминологию, более широкую, чем та, которая используется в IEEE Std 1588 в настоящее время. Поправка еще не одобрена Ассоциацией стандартов IEEE и по состоянию на октябрь 2022 года еще не опубликована.

В данном документе используется следующий перевод терминов, используемых в стандарте IEEE1588:

|  |  |
| --- | --- |
| Условия IEEE Std 1588-2019 | Термины профиля OCP DC PTP |
| Мастер | TimeTransmitter (TT) |
| Slave | TimeReceiver (TR) |
| Grandmaster | Grandmaster (GM) или сервер открытого времени  (OTS) |

# 3 Определение профиля PTP

Профиль PTP — это документ или часть документа, определяющая набор функций и значений атрибутов PTP, применимых к экземпляру PTP, и написанная организацией в соответствии со спецификацией стандарта IEEE Std IEEE1588- 2019. Профиль позволяет организациям определять выбор значений атрибутов и необязательных функций PTP для удовлетворения требований конкретного приложения. В данном документе определен профиль PTP, применимый к центру обработки данных.

Профиль PTP — это набор обязательных и запрещенных опций, а также диапазонов и значений по умолчанию конфигурируемых атрибутов. Примерами являются:

* + - Опция измерения задержки на пути (задержка запрос-ответ или задержка на уровне пира)
    - Диапазон и значения по умолчанию всех настраиваемых атрибутов и членов набора данных
    - Типы экземпляров PTP
    - Опции required, permitted, prohibited
    - Характеристики неопределенности
    - Транспортные механизмы required, permitted, prohibited
    - Если необходимо, значение интервала наблюдения τ, используемого для измерения отклонений PTP.

# 4 Эталонная Модель

Модель, на которую ссылаются в данном разделе, обозначается как Модель 1. Модель состоит из трех уровней. Уровень опорного времени состоит в основном из источника опорного времени (например, GNSS) и функциональности PTP Open Time Server (GM) [4]. Уровень сетевой структуры состоит из набора сетевых элементов, поддерживающих часы PTP, такие как прозрачные часы (TC) или граничные часы (BC). Серверный уровень состоит из группы конечных узлов, которые поддерживают часы PTP, например обычные часы (OC), и на которых обычно размещаются приложения, чувствительные ко времени.

В модели 1 уровень сетевой структуры состоит из цепочки ТС.

## Модель 1 - Прозрачные часы Модель

Высокоуровневыми характеристиками модели 1, представленными на рис. 1, являются:

### GM (или открытый сервер времени) имеет один сетевой физический порт и всегда распределяет время в направлении уровня сетевого полотна и серверного уровня. GM, определенный в данном профиле PTP (и термин GM, используемый в данном документе), представляет собой OC с единственным портом PTP в соответствии с п. 9.2.2.2 стандарта IEEE Std 1588-2019.

* + - TC может иметь несколько сетевых физических портов (например, 16, 48). TC может иметь несколько портов PTP.
    - OC имеет один сетевой физический порт и всегда получает время от сетевого уровня и GM. OC, определенный в данном профиле PTP (и термин OC, используемый в данном документе), является timeReceiver-only2 OC в соответствии с п. 9.2.2.1 стандарта IEEE Std 1588- 2019.
    - Все сетевые элементы, обеспечивающие передачу сообщений PTP между GM и OC,

являются PTP-совместимыми (т.е. TC-capable).

* + - На каждом порту PTP имеется аппаратная временная метка (максимально приближенная к носителям).
    - В нормальном режиме работы OC имеет связь более чем с 1 GM.
    - Существует ряд ГМ, которые находятся в активном или резервном состоянии.
    - OC взаимодействует с GM на основе протокола одноадресного обнаружения и одноадресного согласования.
    - Связь между часами PTP осуществляется преимущественно на основе протокола IPv6.

1 См. раздел 2 - Терминология

2 См. раздел 2 - Терминология

* + - Алгоритм обнаружения и выбора часов PTP не опирается на многоадресную или широковещательную связь.

### Вычисление <meanPathDelay> основано на механизме сквозной задержки.

* + - Количество ТК между ГМ и ОЦ постоянно. Например, если число ТК = 5, то всего будет 7

часов (т.е. включая 1 GM и 1 OC) с 6 связями, соединяющими эти часы.

* + - Направление прямого (GM - OC) и обратного (OC - GM) маршрутов может быть не конгруэнтным. То есть пакеты PTP в прямом направлении могут пересекать другой набор TC, чем пакеты PTP в обратном направлении. Однако ожидается, что количество ТК в обоих направлениях в неконгруэнтном сценарии всегда будет одинаковым, и влияние на асимметрию задержки будет незначительным.
    - Предполагается, что асимметрия задержки из-за волоконно-оптических линий связи пренебрежимо мала, но, скорее всего, не равна нулю, по сравнению с требованиями к временной погрешности.

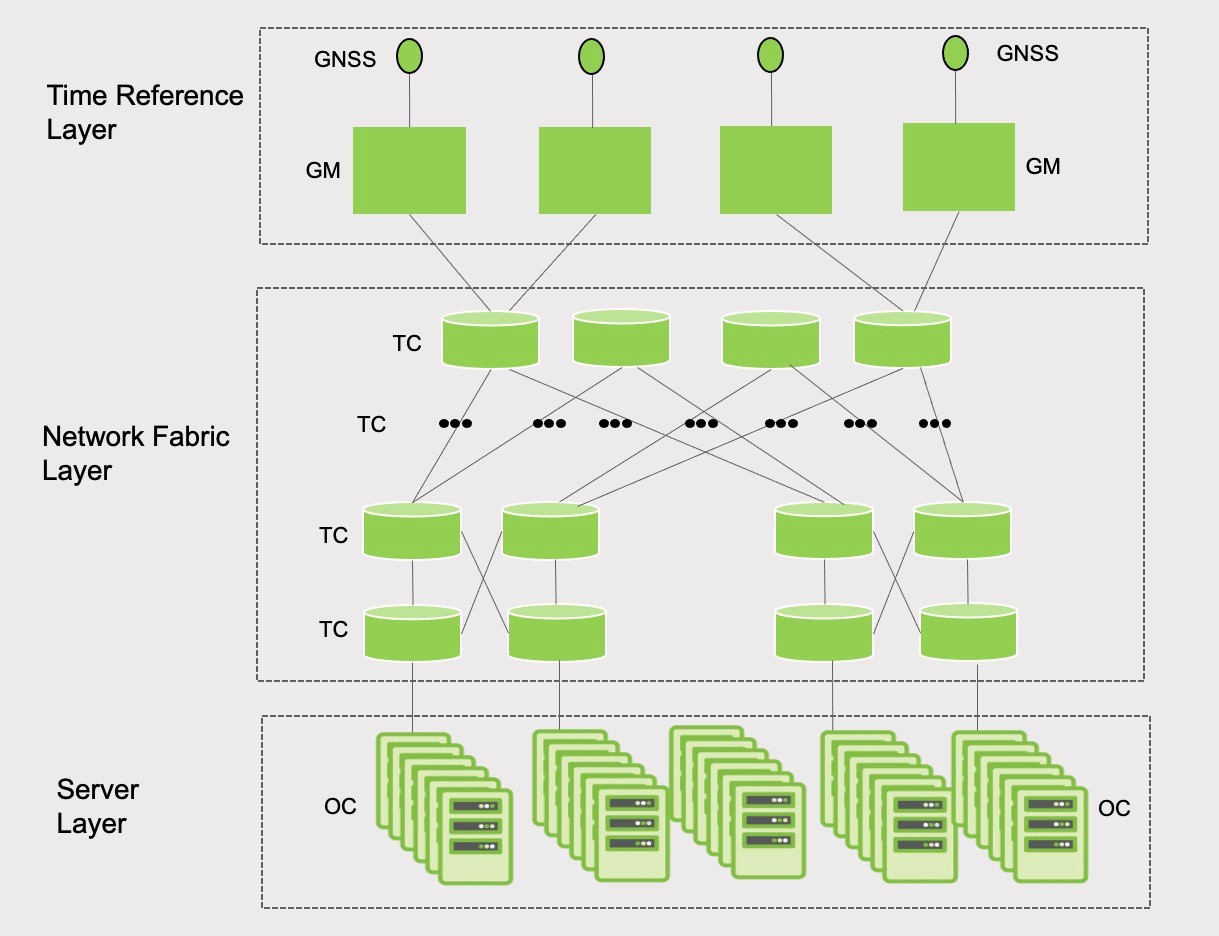


Рисунок 1. Модель 1 - цепочка прозрачных часов

# 5 Модель 1 - Дополнительные требования

* Приложениям более высокого уровня требуется прослеживаемость UTC. Протокол PTP передает временную шкалу PTP (т.е. TAI) плюс всю информацию для получения временной шкалы UTC из временной шкалы TAI. Преобразование шкалы времени должно выполняться приложением.
* Максимальная временная ошибка между любыми двумя OC должна быть в пределах ±5 микросекунд, т.е. |TOC,j– TOC,k| ≤ 5 μs для *k*≠*j.*
* Максимальная временная погрешность между ГМ и любыми ОС должна находиться в пределах ±2,5 микросекунды, т.е. |TGM – TOC| ≤ 2.5 μs.
* Максимальная временная ошибка между любыми двумя ГМ должна находиться в пределах ±100 наносекунд, т.е. |TGM,j – TGM,k | ≤ 100 ns для *k*≠*j .*
* Максимальная временная ошибка, генерируемая ТС, должна находиться в пределах ±100 наносекунд, т.е. |TTC,j | ≤ 200 ns
* В нормальных условиях эксплуатации каждый OC имеет возможность подключения к нескольким GM. При выходе из строя одного из GM OC должен иметь возможность подключения, по крайней мере, к другому GM.
* GM в сети находятся в активном или резервном состоянии.

# PTP Профиль

Профиль PTP основан на стандарте IEEE Std 1588-2019.

## Профиль Идентификатор

Приведенная ниже информация идентифицирует профиль. Профиль определяется OCP.

profileName: Профиль PTP для центра обработки данных (DC-PTP Profile 1) profileNumber: 1

primaryVersion: 1

revisionNumber: 1

profileIdentifier: 7A-4D-2F-01-01-00 organizationName: Open Compute Project (OCP)

sourceIdentification: Данный профиль определен OCP и может быть загружен с сайта

[https://www.opencompute.org.](http://www.opencompute.org/)

## Часы

Профиль позволяет использовать следующие часы. Полные определения приведены в п. 3 стандарта IEEE Std 1588-2019.

|  |  |
| --- | --- |
| GM | PTP-часы, являющиеся источником времени для всех часов в домене |
| ТК | PTP-часы, которые измеряют время прохождения сообщения о событии PTP через PTP-часы и предоставляют эту информацию PTP-часам, принимающим это сообщение о событии PTP. PTP-часы в данном профиле поддерживают механизм запроса задержки - ответа (т.е. сквозные прозрачные часы). |
| OC | PTP-часы, которые имеют один PTP-порт в своем домене и поддерживают шкалу времени, используемую в домене. |

Некоторые дополнительные требования, относящиеся к GM и выходящие за рамки профиля PTP,

определены в проекте OCP-TAP Open Time Server [4].

## Сообщение Типы

Профиль позволяет передавать следующие сообщения:

* + 1. Announce
    2. Sync
    3. Follow\_Up
    4. Delay\_Req
    5. Delay\_Resp
    6. Signaling
    7. Management

## 

## Транспортные механизмы required, permitted, or prohibited

Должен поддерживаться транспортный механизм UDP over IPv6 согласно приложению D стандарта IEEE Std 1588-2019.

Должен поддерживаться транспортный механизм UDP over IPv4 согласно приложению C.

Контрольная сумма UDP должна вычисляться при повторной передаче сообщения PTP (см. 3.1.65

стандарта IEEE Std 1588-2019) по TC.

Должны поддерживаться номера портов назначения UDP согласно приложению C.2 стандарта IEEE Std 1588- 2019. Номер UDP-порта источника одноадресного PTP-сообщения может быть любым эфемерным номером порта и должен сохраняться в течение всего времени существования PTP-соединения, установленного с использованием механизма одноадресного согласования.

## Часы идентичность

Идентификатор часов должен представлять собой EUI-64, как указано в п. 7.5.2.2 стандарта IEEE Std 1588-2019. EUI-64 должен быть глобально уникальным. Если EUI-64 формируется из существующего EUI-48, то это должно быть сделано путем добавления двух октетов после последних шести октетов EUI-48 таким образом, чтобы 64 бита clockIdentity не совпадали с битами любого EUI-64, который был ранее назначен или может быть назначен в будущем уполномоченным лицом MA-L, MA-M или MA-S, из которого был назначен EUI-48. Это означает, что либо организация, формирующая EUI-64, владеет MA- L, MA-M или MA-S, на основе которых был сформирован EUI-48, либо владелец этих MA-L, MA-M или MA-S передал организации, формирующей EUI-64, исключительное право на формируемый clockIdentity.

Примечание: при использовании MAC-адреса идентификатор часов формируется путем добавления двух октетов после последних шести октетов MAC-адреса. В стандарте IEEE Std 1588-2008 идентификатор часов формировался путем добавления двух октетов 'FFFE' между 3-м и 4-м октетами MAC-адреса, однако это соответствие было отменено IEEE.

## Механизм измерения задержки

Механизм измерения задержки на пути должен быть механизмом задержки запрос-ответ. Значение члена набора данных portDS.delayMechanism должно быть E2E.

## Класс Сервис

Сообщения о событиях PTP должны устанавливать в поле DSCP поля IPv6 Traffic Class наивысший возможный класс обслуживания. Это позволит минимизировать задержки и колебания задержек при прохождении пакетов PTP через набор прозрачных часов. В модели 1 GM и OC должны устанавливать значение класса трафика.

## PTP Безопасность

Безопасность PTP не учитывается, так как сеть будет представлять собой единый доверенный домен, управляемый одной организацией.

## Изоляция профиля и домен Число

Все экземпляры PTP должны взаимодействовать с использованием одного доменного номера, причем значение

domainNumber должно быть равно 0.

sdoId — это новый параметр в стандарте IEEE Std 1588-2019. Признанная организация по стандартизации, отраслевая торговая ассоциация, нормативная или правительственная организация или другая организация, как описано в п. 20.3.2 стандарта IEEE Std 1588-2019, может получить sdoId от регистрационного органа (RA) IEEE. sdoId используется для обеспечения изоляции профиля PTP от любых других профилей PTP, работающих в той же сети и разработанных другими организациями.

Организация может получить только один sdoId. Если организация разрабатывает несколько профилей PTP и требует их изоляции, то изоляция осуществляется с помощью domainNumber. Если организация не получила sdoId, то профиль PTP будет использовать sdoId 0x000.

Этот профиль PTP не требует указания sdoId, поскольку предполагается, что он будет единственным профилем в сети центра обработки данных. Если это предположение неверно, то другой профиль, работающий в сети, будет конфликтовать с этим профилем, если sdoId и domainNumber другого профиля равны 0.

Примечание - sdoId обратно совместим с IEEE Std 1588-2008. Первый полубайт sdoId, т.е. majorSdoId, соответствует полю transportSpecific стандарта IEEE Std 1588-2008. Последние 8 бит sdoId, т.е. minorSdoId, были зарезервированы в IEEE Std 1588-2008 и заданы как 0x00.

## Одноступенчатая и двухступенчатая работа

ГМ, определенный в этом профиле, должен поддерживать одноступенчатую или двухступенчатую работу на передаче или может поддерживать обе операции на передаче.

TC, определенный в данном профиле, должен поддерживать одношаговую работу на передачу (т.е. на выход) на всех своих портах или двухшаговую работу на передачу на всех своих портах. TC, определенный в данном профиле, должен поддерживать одноэтапную работу на всех своих портах (см. 6.11).

Все часы PTP должны поддерживать как одношаговый, так и двухшаговый режим работы при приеме (т.е. на входе).

Порт PTP может передавать сообщение Sync как одношаговое или двухшаговое. Если передача сообщения Sync одношаговая, флаг twoStepFlag общего заголовка PTP устанавливается в FALSE, в противном случае - в TRUE. Для сообщений PTP, отличных от Sync, флаг twoStepFlag всегда должен быть установлен в FALSE. Все порты PTP должны быть способны принимать и обрабатывать одношаговые и двухшаговые сообщения Sync.

Примечание: одномоментная работа уменьшает количество сообщений PTP, передаваемых портом PTP. Это может быть применимо при рассмотрении возможности масштабирования одноадресной связи, которую может обслуживать ГМ. Одноэтапная работа может облегчить выполнение требований к передаче сообщений Sync, указанных в п. 9.5.9 стандарта IEEE Std 1588-2019.

Примечание: IEEE Std 1588-2019 допускает одноступенчатую и двухступенчатую работу на основе PTP- портов. Однако IEEE Std 1588-2019 не описывает эту возможность. Данный профиль требует, чтобы все порты PTP на основе каждого такта были одинаковыми.

## End-to-End TC с двухэтапной работой

Данный раздел относится к сценарию, в котором может использоваться двухступенчатый режим работы ТС.

Если сквозной TC использует двухэтапную работу, то каждое сообщение Delay\_Req и соответствующее ему Delay\_Resp должны проходить через один и тот же сквозной TC. Это связано с тем, что сквозной TC проставляет временные метки сообщения Delay\_Req на входе и выходе и вычисляет время пребывания сообщения Delay\_Req. Однако в двухэтапном случае TC обновляет время пребывания соответствующего сообщения Delay\_Resp. Подробно это описано в

10.2.2.2.2 и 10.2.2.2.3 стандарта IEEE Std 1588-2019. В первом подпункте описывается одноэтапный случай и указывается, что "<время пребывания> сообщения Delay\_Req должно быть добавлено в поле коррекции сообщения Delay\_Req исходящим PTP-портом ТС перед повторной передачей сообщения Delay\_Req". В данном случае TC изменяет именно сообщение Delay\_Req, а не сообщение Delay\_Resp. Однако последний подпункт описывает двухэтапный случай и указывает, что "<время пребывания> должно быть добавлено в поле коррекции сообщения Delay\_Resp, связанного с сообщением Delay\_Req, до передачи сообщения Delay\_Resp на порт PTP egress, который является портом PTP ingress для сообщения Delay\_Req".

Если все TC являются двухшаговыми, то для выполнения требований подпункта Delay\_Req и Delay\_Resp должны проходить через один и тот же набор прозрачных часов (каналов и сетевых элементов) между GM и OC. Это свойство может не всегда выполняться при использовании, например, распыления пакетов, балансировки нагрузки и методов многопутевой передачи с равными затратами. Это особенно актуально для сред центров обработки данных и является причиной, по которой рекомендуется использовать ТС с одношаговым режимом работы, как указано в разделе 6.10.

Если все ТК одношаговые, то Delay\_Req и Delay\_Resp не обязательно должны пересекать один и тот же набор ТК (каналов и элементов сети) между GM и OC.

## Сообщение PTP скорость

Таблица 1 определяет диапазон скоростей передачи сообщений Announce, Sync, Delay\_Req и Delay\_Resp. GM должен поддерживать весь диапазон. OC должен поддерживать весь диапазон, но может поддерживать его подмножество. Скорость передачи сообщений, выбранная OC, зависит от ожидаемой производительности. TC не зависит от скорости передачи сообщений PTP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сообщение | Верхний конец диапазона logMessageInterval | Средняя скорость, соответствующа я верхней границе диапазона (pps) | Нижняя граница диапазона logMessageInterval | Средняя скорость, соответствующа я нижней границе диапазона (pps) | Значение по умолчанию logMessageInterval |
| Announce | +4 | 0,0625 (1 на 16 с) | -3 | 8 | 4 |
| Sync | +3 | 0,125 (1 на 8 с) | -7 | 128 | 0 |
| Delay\_Req & Delay\_Resp | 0 | 1 | -7 | 128 | 0 |

Таблица 1. Диапазон значений logMessageInterval для порта PTP

## Интервал между сообщениями PTP

Требования к фактическим межсообщениям для одноадресных сообщений Announce, Sync, Delay\_Req и

Delay\_Resp определены в 16.1 стандарта IEEE Std 1588-2019. Требования предъявляются к:

* (a) среднее арифметическое значение последовательного межсообщения, рассчитанное по соответствующему числу последовательных интервалов
* (b) распределение межпосылочных интервалов

Для сообщений Announce и Sync среднее арифметическое значение межсообщений должно находиться в пределах ±30% от предоставленного межсообщения. Для сообщений Delay\_Req и Delay\_Resp среднее арифметическое значение межсообщений Delay\_Req должно составлять не менее 90% от предоставленного межсообщения для сообщений Delay\_Resp. Цель этого требования - обеспечить получение портом GM сообщений Delay\_Req со скоростью, которую он может выдержать.

Если средний межсообщенийный интервал сообщений Delay\_Req составляет менее 90% от предоставленного межсообщенийного интервала для сообщений Delay\_Resp, порт-донор может игнорировать любые сообщения Delay\_Req, превышающие предоставленный интервал.

При распределении межсообщений не менее 90% межсообщений должны находиться в пределах

±30% от предоставленного среднего межсообщения. Это требование относится к протоколам Announce, Sync и

Delay\_Req.

Рассмотрим *N* последовательных межсообщений *Δti* , *i* = 1, 2, ..., *N*, где *Δti* = (*ti* - *ti-1)*. Среднее арифметическое значение межсообщений*, Δtav, равно*

Например, если порт-грантор выдает сообщения Sync или Announce с logMessageInterval, равным 0, то средний интервал между сообщениями равен 1 с. Это означает, что (а) среднее значение длительности подходящего числа последовательных межсообщений должно находиться в диапазоне от 0,7 до 1,3 с. (б) 90% фактических межсообщений должны иметь длительность от 0.Кроме того, если GM-порт выдает сообщения Delay\_Resp с logMessageInterval, равным 0, то (a) среднее значение длительности подходящего количества последовательных межсообщений Delay\_Req должно быть больше или равно 0,9 с, и (б) 90% фактических межсообщений Delay\_Req должны иметь длительность от 0,7 до 1,3 с.

В принципе, средняя скорость синхронизации и средняя скорость Delay\_Req/Delay\_Resp не обязательно должны быть одинаковыми. Если фактическая задержка на пути передачи данных PTP изменяется достаточно медленно (после обработки OC любого корректирующего поля), то нечастые измерения задержки по сравнению со средним интервалом Sync могут дать приемлемые характеристики. В этом случае средняя скорость Delay\_Req/Delay\_Resp может быть выбрана меньшей, чем средняя скорость Sync. Выбор скорости синхронизации зависит от реализации фильтра OC и от того, какой шум генерирует генератор на OC. Если генератор имеет большую генерацию шума, то скорость синхронизации, скорее всего, будет больше. В этом случае OC будет чаще использовать новую информацию о синхронизации для коррекции временной ошибки.

## Лучший алгоритм часов и атрибуты часов

В данном профиле используется альтернативный BMCA (A-BMCA), описанный в этом подпункте. Требования к A-BMCA описаны в п. 9.3.1 [2]. A-BMCA отличается от BMCA по умолчанию, описанного в [2], использованием нового члена localGmPriority в unicastDiscoveryPortDS, который описан ниже.

Атрибуты часов для ГМ и УО приведены в табл. 2. Атрибуты clockClass, clockAccuracy, offsetScaledLogVariance задаются в defaultDS и представляют собой свойства локальных часов, которые являются либо внутренним генератором, либо внешним источником времени, обеспечивающим время для ГМ вне PTP или интегрированным с ГМ. Для класса часов в техническом описании ГМ должно быть указано максимальное время, необходимое для перехода от класса часов 7 к классу часов 52.

Атрибут priority1 не используется и должен быть установлен в значение 128. Он не используется в данном профиле PTP, поскольку имеет более высокий приоритет в A-BMCA, чем все остальные атрибуты. Неправильная конфигурация может привести к тому, что OC выберет неправильный GM.

Атрибут priority2 может быть сконфигурирован для того, чтобы заставить потенциальные ГМ быть активными или резервными ГМ для определенных групп OC, а также для реализации полного или частичного резервирования на основе активных и резервных ГМ, как показано в разделе 6.15.3 и Приложении I. Если приоритет2 не используется, т.е. значение по умолчанию равно 128 для всех потенциальных ГМ, то выбор фактического ГМ со стороны A-BMCA осуществляется на основе localGmPriority и clockIdentity (при условии равенства остальных атрибутов).

Атрибут timeReceiverOnly равен TRUE для OC и FALSE для GM. Атрибут timeTransmitterOnly является

FALSE для OC и TRUE для GM.

Атрибут ptpTimescale всегда равен TRUE, поскольку данный профиль PTP использует шкалу времени PTP. Остальные атрибуты timePropertiesDS имеют значения в GM в зависимости от того, прослеживаются ли эти значения к первичной ссылке или, в случае timeSource, в зависимости от фактического источника времени для часов.

Атрибут synchronizationUncertain является необязательным. Он передается в виде флага в сообщении Announce. Он является новым в IEEE Std 1588-2019 и может не поддерживаться, если узлы PTP основаны на предыдущей версии протокола. Если он не используется, то его значение равно FALSE. Если он используется в OC, то его значение устанавливается в TRUE, если:

* Флаг synchronizationUncertain в сообщении Announce, полученном от ГМ, установлен в TRUE, или
* Состояние порта PTP OC - UNCALIBRATED

В противном случае атрибут synchronizationUncertain для OC устанавливается в FALSE. Если атрибут synchronizationUncertain используется в GM, то он устанавливается в TRUE, если время или частота GM, или и то, и другое, не отслеживаются по первичному эталону, в противном случае он устанавливается в FALSE.

A-BMCA отличается от стандартного BMCA из [2] использованием нового члена localGmPriority в unicastDiscoveryPortDS, т.е. в таблице одноадресных передатчиков времени (UMT). Этот член представляет собой массив с типом данных UInteger8, т.е. они могут иметь значения в диапазоне [0, 255], по умолчанию значение равно 128. A-BMCA работает точно так же, как и BMCA по умолчанию, но отличается от алгоритма сравнения наборов данных BMCA по умолчанию тем, что сравнение localGmPriority производится после сравнения значений приоритета2 и до сравнения значений GM Identity.

При сравнении двух наборов данных A и B локальные приоритеты GmPriorities для A и B получаются из UMT. При получении сообщения Announce потенциальный GM идентифицируется по IP-адресу источника (т.е. по адресу порта) IP-пакета, содержащего сообщение Announce. Соответствующий localGmPriority для этого потенциального ГМ получается из UMT путем нахождения индекса массива этого IP-адреса в массиве UMT portAddress. Приоритетом localGmPriority является элемент массива localGmPriority с таким же индексом.

Каждый OC запрашивает сообщения Announce у ВСЕХ потенциальных GM в UMT. Алгоритм сравнения наборов данных A-BMCA сравнивает локальные значенияGmPriority после сравнения значений приоритета2 и перед сравнением идентификаторов ГМ (tiebreaker GM Identity) двух сравниваемых потенциальных ГМ. Если атрибуты priority1, GM class, GM accuracy, Gm offsetScaledLogVariance и priority2 для A и B одинаковы, то решение о том, какой из GM A и GM B лучше, будет приниматься на основе localGmPriority. Значения localGmPriority для ГМ могут быть разными в разных OC, поскольку каждый OC имеет свой собственный UMT, члены которого могут отличаться от UMT других OC. Это означает, что один OC может предпочесть GM A GM B, а другой OC может предпочесть GM B GM A. Эта возможность позволяет сконфигурировать сеть, использующую данный PTP-профиль, для случая активного- активного GM. Примеры приведены в Приложении II.

Если все потенциальные ГМ UMT имеют одинаковое значение localGmPriority, то A-BMCA работает точно так же, как и BMCA по умолчанию из [2].

Члены набора данных, перечисленные в табл. 2, не относятся к ТС. ТС не участвуют в A-BMCA.

|  |  |  |  |
| --- | --- | --- | --- |
| **Набор данных** | **Участник** | **Значение** | |
| **GM** | **OC** |
| defaultDS | clockClass | 1. (прослеживается до первичного эталонного источника времени) 2. (в удержании и в рамках спецификаций удержания)   52 (в удержании, но вне спецификаций удержания, или в свободном движении) | 255 |
| defaultDS | clockAccuracy | 0x22 (250 нс) | 0xFE (неизвестно) |
| defaultDS | offsetScaledLogVariance | 0x4E5D (PTPVAR = 1,144 × 10-15 с2, или  TDEV = 30 нс) | 0xFFFF (максимально возможное значение, обозначающее неизвестность) |
| defaultDS | приоритет1 | 128 (в данном профиле не используется) | 128 (в данном профиле не используется) |
| defaultDS | приоритет2 | Настраивается в диапазоне [0, 255]. Значение по умолчанию - 128. Для конфигурации "активный - резервный". | Настраивается в диапазоне [0,255].  Значение по умолчанию - 128 |
| defaultDS | timeReceiverOnly | FALSE | TRUE |
| portDS | timeTransmitterOnly | TRUE | FALSE |
| timePropertiesDS | currentUtcOffset | Если известно, то значение, прослеживаемое по первичному эталону, который обеспечивает UTC. В противном случае - значение на момент разработки узла | currentUtcOffset Ebest, после выполнения BMCA |
| timePropertiesDS | currentUtcOffsetValid | TRUE, если значения currentUtcOffset, leap59 и leap61 основаны на значениях, полученных из первичной ссылки, обеспечивающей UTC; в противном случае устанавливается значение FALSE | CurrentUtcOffsetValid для Ebest, после выполнения BMCA |

|  |  |  |  |
| --- | --- | --- | --- |
| timePropertiesDS | скачок59 | Если известно, то значение, прослеживаемое к первичному эталону; в противном случае устанавливается значение FALSE | скачок59 Ebest, после запуска BMCA |
| timePropertiesDS | скачок61 | Если известно, то значение, прослеживаемое к первичному эталону; в противном случае устанавливается значение FALSE | Leap61 из Ebest, после пробега BMCA |
| timePropertiesDS | timeTraceable | FALSE (не используется в данном профиле) | FALSE (не используется в данном профиле) |
| timePropertiesDS | frequencyTraceable | FALSE (не используется в данном профиле) | FALSE (не используется в данном профиле) |
| timePropertiesDS | timeSource | Если известно, то на соответствующее значение из таблицы 6/IEEE Std 1588-2019. В противном случае устанавливается значение INTERNAL\_OSCILLATOR | timeSource of Ebest, после запуска BMCA |
| timePropertiesDS | ptpTimescale | TRUE | TRUE |
| currentDS | синхронизацияНеопреде ленность | FALSE (по умолчанию) | FALSE (по умолчанию) |
| unicastDiscoveryPortDS (одноадресная таблица TimeTransmitter) | localGmPriority | 128 (в данном профиле не используется) | Настраивается в диапазоне [0,255].  Значение по умолчанию - 128 |

Таблица 2. Члены и значения набора данных

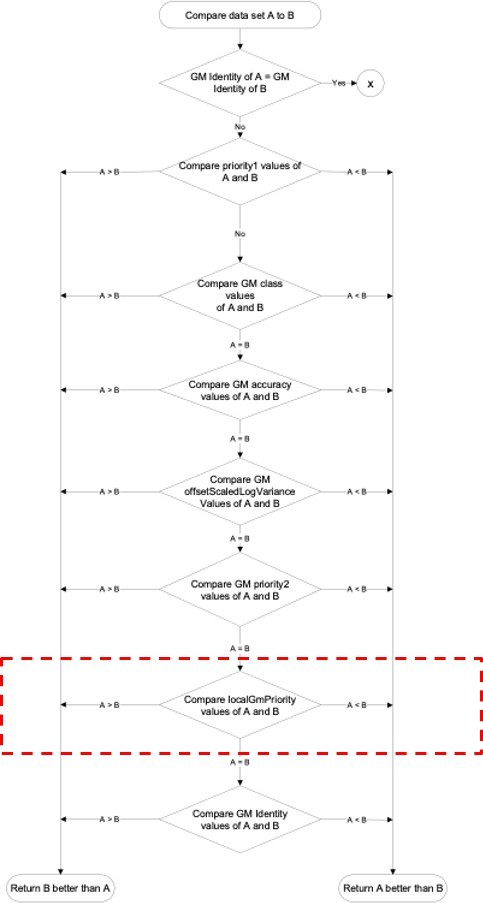


Рисунок 2. Алгоритм сравнения наборов данных для A-BMCA (красным пунктиром выделено отличие от стандартного алгоритма BMCA на рис. 34 из [2])

Примечание: BMCA, определенный в предыдущем документе (т.е. primaryVersion: 1 revisionNumber: 0), был основан на BMCA по умолчанию. В данном документе (т.е. primaryVersion: 1 revisionNumber: 1) BMCA основан на A-BMCA. A-BMCA и BMCA по умолчанию могут сосуществовать в сети в том смысле, что протокол будет функционировать в соответствии со стандартом IEEE Std 1588. Набор OC может быть синхронизирован GM в соответствии с BMCA по умолчанию, а другой набор OC может быть синхронизирован GM в соответствии с a-BMCA.

## Одноадресная связь

Связь PTP в этом профиле основана на одноадресной рассылке. Большинство профилей PTP в отрасли основаны на многоадресной рассылке, за исключением двух телекоммуникационных профилей ITU-T, которые основаны на одноадресной рассылке [5, 6].

Должны поддерживаться как одноадресное обнаружение (п. 17.4 стандарта IEEE Std 1588-2019), так и одноадресное согласование (п. 16.1 стандарта IEEE Std 1588- 2019). В модели 1 каждый OC сначала использует одноадресное обнаружение для определения потенциальных ГМ, а затем использует одноадресное согласование для запроса сообщений Announce от потенциальных ГМ. Затем OC вызывает A-BMCA для определения того, кто из потенциальных GM становится фактическим GM, т.е. активным GM. Наконец, OC использует одноадресные переговоры для запроса сообщений Sync и Delay\_Resp от активного ГМ и использует Sync,

Delay\_Resp и Delay\_Req для синхронизации с выбранным GM. Другие потенциальные GM доступны в качестве резервных, если активный GM вышел из строя, или могут быть выбраны в качестве активного GM для других OC.

A-BMCA может использоваться двумя способами. В одном из вариантов все элементы массива localGmPriority устанавливаются в одно и то же значение, и A-BMCA работает так же, как и BMCA по умолчанию из [2]. При таком подходе полное или частичное резервирование достигается за счет того, что часть ГМ активна, а часть в резерве. Это описано в п. 6.15.3, а примеры приведены в Приложении I. При втором способе элементы массива localGmPriority устанавливаются в разные значения. При таком подходе все ГМ активны, и при отказе одного ГМ синхронизируемые им УО переключаются на другие ГМ. Это описано в разделе 6.15.4, а примеры приведены в Приложении II.

Функция одноадресного согласования постоянно включена. Член unicastNegotiationPortDS.enable (unicastNegotiationPortDS) должен быть TRUE для каждого порта PTP (для каждого порта PTP существует unicastNegotiationPortDS). Этот член набора данных применим к GM и OC и не применим к TC.

Флаг unicastFlag всех сообщений PTP должен быть установлен в TRUE.

## 6.15.1 Unicast Discovery

Одноадресное обнаружение указано в п. 17.4 стандарта IEEE Std 1588-2019.

В модели 1 данного профиля PTP в каждом OC конфигурируется таблица потенциальных GM. Эта таблица иногда называется Unicast Table (UMT) и определяется в наборе данных unicastDiscoveryPortDS в п.

* + 1. стандарта IEEE Std 1588-2019. Этот набор данных содержит следующие члены:
       1. maxTableSize: максимальное количество потенциальных ГМ, которое может находиться в таблице
       2. logQueryInterval: логарифм по основанию 2 среднего интервала времени в секундах между последовательными запросами, которые OC делает к потенциальному GM для получения сообщений Announce (если запрос не был удовлетворен), по умолчанию значение равно 0,
       3. actualTableSize: количество потенциальных ГМ, находящихся в данный момент в таблице; и
       4. portAddress: массив, содержащий протокольные адреса, т.е. IPv6-адреса потенциальных ГМ.
       5. localGmPriority: массив, содержащий локальные приоритеты потенциальных ГМ. Тип данных - UInteger8. Этот массив является новым членом unicastDiscoveryPortDS, используемым в A- BMCA, описанном в п. 6.14.

Каждый OC использует одноадресное согласование для запроса сообщений Announce от каждого потенциального GM, содержащегося в unicastDiscoveryPortDS. Если потенциальный GM не удовлетворяет запрос, OC повторяет попытку через интервал времени, соответствующий logQueryInterval. Полученные сообщения Announce вызывают событие принятия решения о состоянии, которое приводит к вызову A- BMCA. В результате один из потенциальных ГМ становится активным ГМ для OC. Если этот GM не справляется, OC перестает получать сообщения Announce и истекает время accounceReceiptTimeout. Это приведет к вызову A-BMCA. В результате A-BMCA один из других GM (т.е. лучший из оставшихся потенциальных GM) станет активным GM. Если в unicastDiscoveryPortDS нет ни одного GM или ни один из GM в unicastDiscoveryPortDS не передает OC сообщения Announce, OC переходит в режим free-run или holdover.

После выбора GM OC использует одноадресное согласование для запроса сообщений Sync и Delay\_Resp от GM. После получения сообщений Sync OC получает сообщения Sync от GM. Получив сообщения Delay\_Resp, OC посылает сообщения Delay\_Req в GM и получает сообщение Delay\_Resp в ответ на каждое сообщение Delay\_Req.

## Unicast Negotiation

OC запрашивает сообщения Announce и затем выбирает лучшего потенциального GM с помощью A- BMCA. Затем OC запрашивает у этого GM сообщения Sync и Delay\_Resp. После того как OC получает сообщения Sync, GM посылает OC сообщения Sync (и Follow\_Up, если связь двухэтапная). После предоставления OC сообщений Delay\_Resp OC отправляет GM сообщения Delay\_Req, а GM отвечает Delay\_Resp. Запрос сообщений Announce, Sync и Delay\_Resp осуществляется с использованием функции одноадресного согласования IEEE Std 1588- 2019. Функция одноадресных переговоров выполняется с помощью следующих четырех TLV:

* + - * REQUEST\_UNICAST\_TRANSMISSION
      * GRANT\_UNICAST\_TRANSMISSION
      * CANCEL\_UNICAST\_TRANSMISSION
      * ПОДТВЕРЖДЕНИЕ\_ОТМЕНЫ\_КОММУНИКАЦИОННОЙ\_ПЕРЕДАЧИ Каждый TLV включается в сигнальное сообщение.

Отправка, прием и обработка TLV одноадресных переговоров OC и GM должны соответствовать требованиям раздела 16.1 стандарта IEEE Std 1588-2019 и его подразделов. Следующий текст данного раздела представляет собой краткое описание процесса одноадресного согласования.

TC не участвуют в процессе одноадресных переговоров. Однако они пересылают одноадресные сигнальные сообщения, содержащие TLV одноадресных переговоров, которыми обмениваются OCs и GMs.

Процесс одноадресного согласования проиллюстрирован на рис. 2, 3, 4 для запросов сообщений Announce, Sync и Delay\_Resp соответственно. OC запрашивает одноадресные сообщения Announce, Sync или Delay\_Resp у GM, посылая ему TLV REQUEST\_UNICAST\_TRANSMISSION. Этот TLV содержит поле messageType, которое указывает тип сообщения (т.е. Announce, Sync, Delay\_Resp), поле logInterMessagePeriod, которое представляет собой логарифм по основанию два желаемого среднего интервала в секундах между последовательными сообщениями этого типа, и поле durationField, которое представляет собой количество секунд, в течение которых GM должен продолжать передачу этих сообщений. В ответ GM выдает TLV GRANT\_UNICAST\_TRANSMISSION для удовлетворения или отклонения запроса. Этот TLV содержит поле messageType, указывающее на предоставляемое сообщение, поле logInterMessagePeriod, представляющее собой логарифм по основанию два предоставленного среднего интервала в секундах между последовательными сообщениями данного типа, поле durationField, представляющее собой предоставленное количество секунд, в течение которых GM будет продолжать передачу этих сообщений, и флаг R (Renewal Invited), который равен TRUE, если GM считает, что грант может быть возобновлен, если OC запросит новый грант после истечения срока действия текущего гранта, и FALSE в противном случае. Нулевое значение поля durationField означает, что в предоставлении гранта было отказано. Предоставленные значения logInterMessagePeriod и durationField не обязательно должны совпадать с запрошенными значениями logInterMessagePeriod и durationField соответственно.

УО может запросить для LogInterMessagePeriod любое значение из диапазона, указанного в табл. 1. GM

может предоставлять разным OC различные скорости передачи сообщений.

Длительность гранта начинается с момента передачи GRANT\_UNICAST\_TRANSMISSION\_TLV и заканчивается по истечении интервала времени, равного значению поля durationField. Как правило, OC запрашивает продление гранта, передавая новый TLV REQUEST\_UNICAST\_TRANSMISSION до истечения срока действия гранта (т.е. до окончания его продолжительности), чтобы обслуживание было непрерывным.

После того как ГМ предоставил OC сообщения Announce или Sync, ГМ посылает OC сообщения Announce или Sync. После того как ГМ разрешил сообщения Delay\_Resp, OC отправляет ГМ сообщения Delay\_Req, а ГМ отвечает на них Delay\_Resp. ГМ отвечает на все сообщения Delay\_Req, поступившие до истечения срока разрешения. Однако GM может ответить на сообщение Delay\_Req, т.е. отправить соответствующее сообщение Delay\_Resp, после истечения срока действия гранта, если сообщение Delay\_Req поступило до истечения срока действия гранта.

OC может отменить грант, отправив в GM TLV CANCEL\_UNICAST\_TRANSMISSION. Этот TLV содержит поле messageType, указывающее на тип сообщения, грант которого отменяется, и флаг R (maintainRequest), установленный в FALSE. В ответ GM посылает OC TLV ACKNOWLEDGE\_ CANCEL\_UNICAST\_TRANSMISSION.

Если ГМ не может продолжать предоставлять предоставленные сообщения до истечения поля durationField, он может сообщить об этом OC, отправив ему TLV CANCEL\_UNICAST\_TRANSMISSION с флагом G (maintainGrant), установленным в FALSE. В ответ OC посылает ГМ TLV ACKNOWLEDGE\_ CANCEL\_UNICAST\_TRANSMISSION. GM должен (т.е. это рекомендуется, но не требуется) продолжать посылать сообщения до тех пор, пока не получит TLV ACKNOWLEDGE\_ CANCEL\_UNICAST\_TRANSMISSION или не пошлет OC определенное реализацией количество TLV CANCEL\_UNICAST\_TRANSMISSION.

Сигнальное сообщение может содержать более одного TLV.

В этом профиле PTP все запросы выполняются OC, а все гранты - GM. OC не может предоставлять услуги, а GM не может запрашивать услуги.

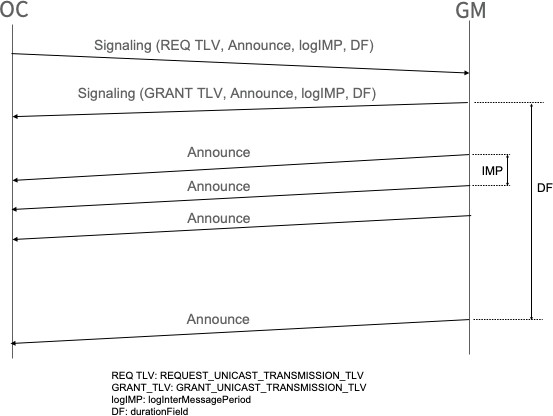


Рисунок 3. Одноадресное согласование для сообщений Announce

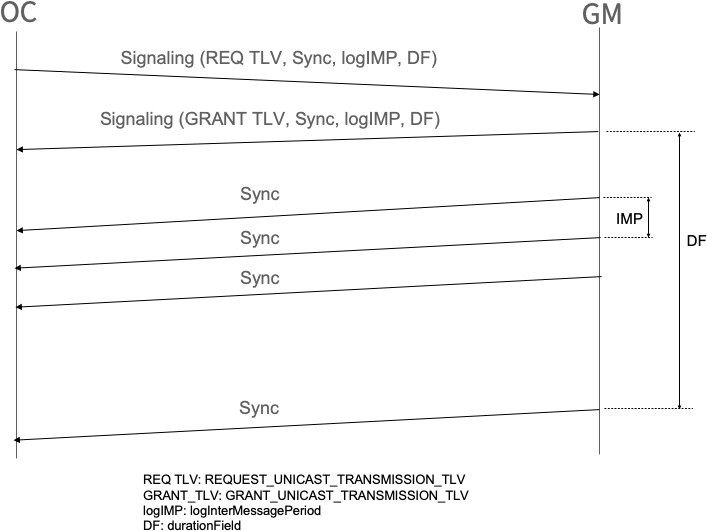


Рисунок 4. Одноадресное согласование для сообщений синхронизации

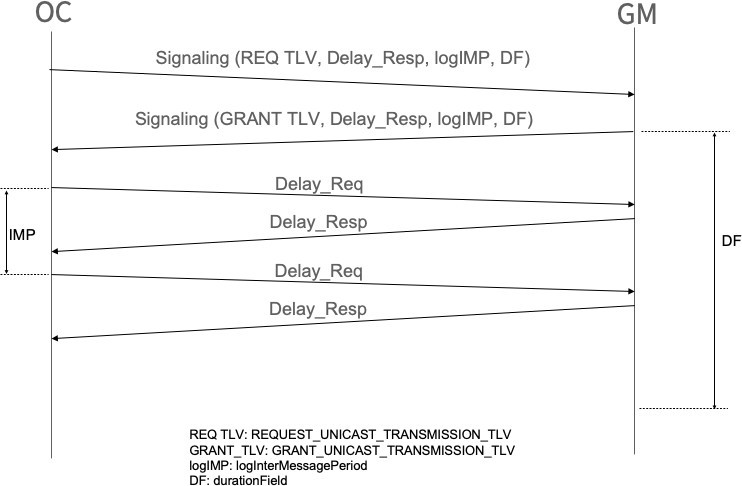


Рисунок 5. Одноадресное согласование для сообщений Delay\_Resp

## Сценарий Active-Standby

A-BMCA может использоваться для обеспечения полного и частичного резервирования ГМ при нормальной работе и в сценариях отказа, когда часть ГМ является активной, а часть - резервной. В этом случае все локальные приоритеты устанавливаются на одно и то же значение, и A-BMCA работает точно так же, как и BMCA по умолчанию [2]. Поскольку все локальные приоритеты имеют одно и то же значение, атрибуты каждого потенциального ГМ одинаковы для всех OC. Это означает, что если один потенциальный ГМ признан алгоритмом сравнения наборов данных лучше другого потенциального ГМ в одной УО, то он будет признан лучшим и во всех УО. В результате резервный ГМ либо не используется, либо используется лишь частично, когда основной ГМ (т.е. лучший по своим характеристикам) выходит из строя.

Примеры сценария "активный-резервный" приведены в Приложении I.

## Сценарий Active-Active

В примерах 6.15.3 резервные ГМ находятся в режиме ожидания, т.е. не используются во время нормальной работы при отсутствии сбоев. Резервный ГМ используется только при возникновении сбоя. Примеры, приведенные в Приложении I, показали, что для полного резервирования, т.е. способности переносить отказ всех ГМ, требуется равное количество ГМ.

резервные ГМ. В некоторых приложениях наличие большого числа ГМ, используемых только при сбоях, может показаться слишком дорогим. К сожалению, как указано в п. 6.15.3, атрибуты потенциального ГМ одинаковы для всех УО, т.е. невозможно, чтобы разные УО имели различные предпочтения среди ГМ. Это объясняется тем, что алгоритм сравнения наборов данных IEEE Std 1588 (см. п.

9.3.4 в [2]) сравнивает присущие ГМ атрибуты. Это означает, что если ГМ, предназначенный для нормальной работы, предпочитается одним УО ГМ, предназначенному для нормальной работы в режиме ожидания, то и все остальные УО будут иметь такие же предпочтения. Для преодоления этого ограничения в алгоритме сравнения наборов данных A- BMCA, описанном в 6.14, используется новый атрибут - localGmPriority, который позволяет различным УО иметь различные предпочтения среди потенциальных ГМ. Подробности приведены в разделе 6.14, а примеры использования А-БМКА в сценариях "активный-активный" - в Приложении II.

## Управление PTP сообщений

Профиль использует механизм управления PTP и управляющие сообщения PTP (TLVs), определенные в п. 15 стандарта IEEE Std 1588-2019. Сообщения управления используются узлом управления PTP для конфигурирования и/или мониторинга экземпляров PTP.

В этой версии профиля должны поддерживаться следующие TLV управления:

* DEFAULT\_DATA\_SET (managementId 2000)
* CURRENT\_DATA\_SET (managementId 2001)
* PARENT\_DATA\_SET (managementId 2002)

Должен поддерживаться следующий дополнительный TLV (Примечание: Этот TLV является специфическим для реализации TLV и поддерживается реализацией linuxptp. TLV содержит набор счетчиков сообщений PTP, которые могут быть использованы для мониторинга):

* PORT\_STATS\_NP (managementId C005)

Для расчета границ временной ошибки могут быть определены дополнительные TLV управления PTP.

Этот вопрос подлежит дальнейшему изучению.

## Пределы сети и бюджет ошибок для модели 1

Данный раздел является начальным анализом. Сетевое ограничение, полученное в разделе 5, имеет вид:

* Максимальная абсолютная временная погрешность любого OC относительно TAI должна быть max|TEOC| ≤ 2,5 μs
* Разница во временной точности между любыми двумя OC должна быть в пределах ±5 микросекунд, т.е. |TOC,*j* - TOC,k| ≤ 5 μs для *k* ≠ *j*.

Вклад в max|TEOC| вносят следующие эффекты:

* + 1. Гранулярность временных меток. Это связано с тактовой частотой, используемой для формирования временной метки.
    2. Генерация временных меток. Это связано с тем, что генерация временной метки происходит не в том месте, где она снимается, т.е. в опорной плоскости (см. 7.3.4.2 стандарта IEEE Std 1588-2019).
    3. Сочетание времени пребывания и точности свободного хода ТК. В данном профиле предполагается, что ТК работают в свободном режиме. Они не синтонизируются ни на физическом уровне, ни через PTP.
    4. Максимальное количество ТК между ГМ и ОК.
    5. Генерация шумов, обусловленных характеристиками OC-генератора.
    6. Характеристики фильтра PLL.
    7. Точность ГМ. Это максимальная временная погрешность ГМ относительно TAI при отслеживании. Данный анализ относится к спецификации ITU-T G.8272 PRTC-A [7].
    8. Постоянная временная ошибка. Это связано с асимметрией каналов и узлов после любой компенсации
    9. Разрешение на ошибку времени, создаваемое приложением или внутри него (т.е. любая дополнительная ошибка между уровнем PTP и приложением/сервером).
    10. Эффект переходного процесса при потере активного ГМ и переключении на резервный ГМ.
    11. Эффект длительного удержания ГМ (например, помехи ГНСС, солнечная активность) или ОС, если резервный ГМ недоступен.

В табл. 3 приведены исходные предположения для приведенных выше эффектов. Значения в таблице, за исключением гранулярности временных меток, максимального времени пребывания, количества ТС и характеристик фильтра конечных точек, относятся к абсолютной величине.

|  |  |
| --- | --- |
| **Эффект** | **Значение** |
| Гранулярность временной метки | 8 нс |
| Формирование временных меток | 8 нс |
| Максимальное время пребывания в ТК | 0,1 мс |
| Точность свободного хода генератора ТС | 100 стр. |
| Максимальное количество ТС | 5 |
| Генерация шума при работе OC | 100 нс (TBD) |
| Характеристики фильтра конечных точек OC | TBD |
| Точность ГМ относительно TAI при прослеживаемости | 100 нс |
| Погрешность постоянного времени | 200 нс |
| Допущение временной погрешности при подаче заявки | 200 нс (TBD) |

|  |  |
| --- | --- |
| Эффект переходного процесса при потере связи УО с активным ГМ и переключении на резервный ГМ | 1400 нс (TBD - см. ниже) |
| Влияние длительного удержания ГМ с тактовым классом 7 на ОС при отсутствии резервного ГМ с тактовым классом 6 | 1400 нс за время T, указанное производителем (TBD  см. ниже) |

Таблица 3. Бюджет максимальной абсолютной временной погрешности

Максимальная погрешность, вносимая ТС за счет точности свободного пробега и времени пребывания, составляет . Кроме того, ТС вносит погрешности в 8 нс за счет гранулярности временной метки и 8 нс за счет генерации временной метки. Эти ошибки будут добавлены как на входе, так и на выходе, что в сумме составит 32 нс. Таким образом, общая ошибка, вносимая TC при прохождении маршрута от входа до выхода, составляет 42 нс.

Погрешности, связанные с гранулярностью временных меток и их генерацией, также вносятся на выходе GM

и входе OC. Эти ошибки добавят 16 нс, итого 32 нс.

Вышеперечисленные ошибки вносят вклад в общую ошибку времени на OC (для конечного приложения). Во-первых, они накапливаются при прохождении сообщения Sync по сети от GM к OC и вносят вклад в ошибку восстановленного времени на OC. Во-вторых, они также накапливаются при прохождении сообщений Sync и Delay\_Req по сети от GM к OC и от OC к GM и вносят вклад в ошибку средней задержки пути на OC. Суммарная ошибка, накапливающаяся при прохождении по сети сообщений Sync или Delay\_Req, при условии наличия 5 ТК в тракте, составляет 5(42 нс) + 32 нс = 242 нс. Таким образом, общая ошибка в синхронизированном времени равна сумме ошибки для Sync и ошибки в измеренной задержке тракта, т.е. 242 нс (ошибка в Sync) + 242 нс (ошибка в meanPathdelay) = 484 нс (ошибка в смещении времени между OC и GM). Наконец, необходимо добавить 100 нс на генерацию шума OC, чтобы получить 584 нс.

Погрешность, вносимая ГМ на основе PRTC-A, составляет 100 нс.

Общее допустимое значение постоянной временной ошибки, обусловленной асимметрией каналов и узлов, основано на стандарте G.8271.1. Стандарт G.8271.1 допускает 800 нс для сети, состоящей из 20 хопов с каналами связи, которые, скорее всего, гораздо длиннее, чем ожидается в среде ЦОД (т.е. длина волокна между узлами в ЦОД составляет метры или десятки метров). Учитывая, что cTE линейно аддитивен и что количество часов состоит из 5 TC, 1 OC и 1GM, общий cTE составляет примерно ¼ от распределения, приведенного в G.8271.1. Таким образом, ошибка постоянной времени составляет 200 нс.

Суммарная ошибка на входе приложения составляет 584 нс + 100 нс + 200 нс + 200 нс = 1100 нс. Это вполне соответствует max|TEos|≤2.5 μs

Если OC теряет связь с сетью и переходит в режим удержания или GM теряет связь с источником времени (например, GNSS) и переходит в режим удержания с clockClass = 7, можно считать, что приложение уже накопило ошибку в 1100 нс относительно TAI. В худшем случае приложение может дрейфовать еще 1400 нс, прежде чем превысит требование 2,5 мкс. Это означает, что требование по удержанию для OC или GM можно принять равным 1400 нс за период времени T. Этот период T должен быть указан в техническом описании OC или GM. Кроме того, если OC переключается с одного активного ГМ на другой активный ГМ, то любой переходный процесс при таком переключении должен быть в пределах 1400 нс.

# 7 Ссылки

1. Предложение по инкубации проекта OCP Timing Appliances

Project (TAP), июль 2020 г.,

<https://www.opencompute.org/wiki/Time_Appliances_Project>

1. IEEE Std 1588-2019, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, June 2020
2. OCP Contribution, Practical Uses of Synchronized Clocks, Sept 2020, [https://www.opencompute.org/wiki/Time\_Appliances\_Project](http://www.opencompute.org/wiki/Time_Appliances_Project)
3. OCP Open Time Server, [https://www.opencompute.org/wiki/Time\_Appliances\_Project](http://www.opencompute.org/wiki/Time_Appliances_Project)
4. ITU-T G.8275.2, Телекоммуникационный профиль протокола точного времени для фазовой/временной синхронизации с частичной поддержкой синхронизации из сети
5. ITU-T G.8265.1, Телекоммуникационный профиль протокола точного времени для частотной синхронизации
6. ITU-T G.8272, Временные характеристики первичных эталонных часов времени

# 8 Документ Версия

|  |  |
| --- | --- |
| **Версия** | **Комментарии** |
| 0.1 | Первоначальный документ по профилю DC PTP передан в OCP |
| 0.2 | Преобразование v0.1 в шаблон документа OCP. Устранение замечаний, полученных от различных авторов |
| 0.3 | Добавлена таблица ревизий. Обновлен раздел о лицензиях OCP. |
| 0.4 | Добавлен IEEE CID и обновлен profileIdentifier на основе полученного CID (идентификатора компании). Добавлены сообщения управления PTP. Добавлен уточняющий текст по IPv6/IPv4/UDP. |
| 1.0 | Первоначальная версия передана в инкубационный комитет OCP и в комитет по рецензированию. Размещена в вики OCP- TAP и на github. |
| 2.0 | Незначительные редакторские правки. Определение A-BMCA с атрибутом localGmPriority. Обновления UMT на основе определения A-BMCA. Приложение с примерами Active-Standby и Active-Active. |

# 9 Вкладчики

### EDITOR(S):

### Николай Красногорский, Qantum

### Вячеслав Миронов, Qantum

# 

# 10 Лицензия

OCP поощряет участников делиться своими предложениями, спецификациями и разработками с сообществом. Это делается в целях обеспечения открытости и поощрения постоянной и открытой обратной связи. Важно помнить, что, предоставляя отзывы на любые подобные документы в письменной или устной форме, участник или его организация предоставляет OCP и ее членам безотзывное право использовать эти отзывы в любых целях без каких-либо дополнительных обязательств.

Подтверждается, что любая такая документация и любые вспомогательные материалы, которые предоставляются OCP в связи с данным документом, включая, без ограничений, любые технические документы, статьи, фотографии, исследования, диаграммы, контактную информацию (вместе - "Материалы"), предоставляются на условиях международной лицензии Creative Commons Attribution- ShareAlike 4.0, находящейся здесь: https://creativecommons.org/licenses/by-sa/4.0/, или любой более поздней версии, и, не ограничивая вышесказанное, OCP может предоставлять Материалы на таких условиях.

Все участники, вносящие вклад в настоящий документ, заявляют, что они обладают полномочиями на предоставление прав и лицензий, указанных в настоящем документе. Кроме того, они заявляют и гарантируют, что Материалы не нарушают и не будут нарушать авторские права или права на коммерческую тайну любой третьей стороны, включая, без ограничения, права на интеллектуальную собственность. Соавтор(ы) также заявляет(ют), что в той мере, в какой Материалы включают материалы, защищенные авторским правом или правом на коммерческую тайну, которые принадлежат или созданы любой третьей стороной, они получили разрешение на их использование в соответствии с вышеизложенным. Они предоставят OCP доказательства такого разрешения по запросу OCP. Настоящий документ и любые "Материалы" публикуются на вики-странице соответствующего проекта и являются общедоступными в соответствии с Уставом и политикой OCP в области ИС. С ними можно ознакомиться по адресу [http://www.opencompute.org/participate/legal-documents/.](http://www.opencompute.org/participate/legal-documents/) Если у вас возникли какие-либо вопросы, пожалуйста, свяжитесь с OCP.

# 11 Приложение I. Примеры конфигурации Active-Standby

В примерах данного приложения для каждого ГМ локальныйGmPriority установлен на одно и то же значение, т.е. 128. В результате localGmPriority не влияет на определение алгоритмом сравнения наборов данных, какой из двух ГМ лучше. A-BMCA работает точно так же, как и BMCA по умолчанию из [2].

На рис. 6 показан пример, состоящий из 1000 КО, разделенных на 2 группы, каждая из которых состоит из 500 КО. Имеется 4 потенциальных GM, обозначенных с 1 по 4 соответственно. GM 1 и GM 2 - потенциальные GM для группы OC 1, их IPv6-адреса вводятся в unicastDiscoveryPortDS каждого OC группы

1. GM 3 и GM 4 - потенциальные GM для группы OC 2, их IPv6-адреса вводятся в unicastDiscoveryPortDS каждого OC группы 2. Атрибуты GM заданы таким образом, что GM 1 лучше GM 2, что определяется A- BMCA, а GM 3 лучше GM 4, что определяется A-BMCA. Если предположить, что все ГМ имеют одинаковые clockClass, clockAccuracy и offsetScaledLogVariance, то это можно сделать, настроив атрибут priority2 таким образом, чтобы приоритет2 для ГМ 1 и ГМ 3 был меньше, чем приоритет2 для ГМ 2 и ГМ 4 соответственно. При этом предполагается, что приоритет1 установлен в одно и то же значение по умолчанию во всех GM, чтобы предотвратить случайное переопределение влияния clockClass, clockAccuracy и offsetScaledLogVariance. Это сделано в других профилях PTP, таких как ITU-T Rec. G.8275.2 [5], который также основан на одноадресном обнаружении и одноадресном согласовании. В качестве альтернативы, если clockClass, clockAccuracy, offsetScaledLogVariance и prioirity2 одинаковы у каждого потенциального GM, но clockIdentities GM 1 и GM 3 меньше clockIdentities GM 2 и GM 4, соответственно, то GM 1 и GM 3 также будут выбраны в качестве активных GM для групп 1 и 2, соответственно. Кроме того, в последнем случае, когда потенциальные ГМ имеют одинаковые тактовые атрибуты, может быть неважно, кто из них активный, а кто резервный. В результате A-BMCA GM 1 и GM 3 будут активными GM для групп OC 1 и 2, соответственно, а GM 2 и GM 4 - резервными GM для групп OC 1 и 2, соответственно. Использование clockIdentites является решающим фактором.

В примере 1 также показано, что резервный ГМ не используется, если активный ГМ соответствующей группы OC не вышел из строя. В этом примере можно допустить отказ обоих активных ГМ. Однако два резервных ГМ не используются, если нет отказов.

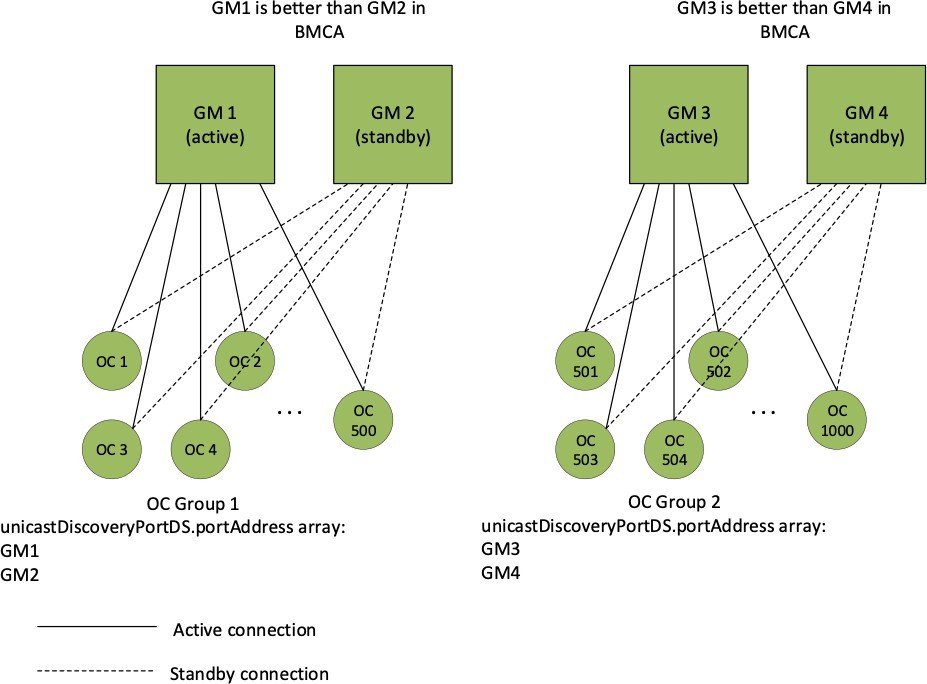


Рисунок 6. Пример 1 активно-резервных ГМ в двух группах по 500 OC в каждой

На рис. 7 показан пример, состоящий из 2 групп OC с 3 потенциальными ГМ, обозначенными с 1 по 3 соответственно. GM 1 и GM 3 являются потенциальными GM для группы OC 1 и вводятся в unicastDiscoveryPortDS каждого OC группы 1. GM 2 и GM 3 являются потенциальными GM для группы OC 2 и вводятся в unicastDiscoveryPortDS каждого OC группы 2. GM3 по существу является общим GM для двух групп OC. Атрибуты GM устанавливаются таким образом, чтобы GM 1 и GM 2 были лучше, чем GM 3, как это определено A-BMCA. Как и в приведенном выше примере, это можно сделать, настроив атрибуты priority2 таким образом, чтобы приоритет 2 для GM 1 был меньше приоритета 2 для GM 3, а приоритет2 для GM 2 был меньше приоритета2 для GM 3. Это также произойдет, если clockIdentity ГМ 1 и ГМ 2 меньше clockIdentity ГМ 3, а все остальные атрибуты ГМ 1, 2 и 3 одинаковы. В результате A-BMCA GM 1 и GM 2 будут активными GM для групп OC 1 и 2 соответственно. GM 3 будет резервным GM для обеих групп 1 и 2. Если GM 1 или GM 2 выйдет из строя, GM 3 станет GM той группы, GM которой вышел из строя. Если откажут оба GM 1 и GM 2, то либо GM 3 станет GM для обеих OC-групп 1 и 2, и, следовательно, он должен быть способен выдержать нагрузку обеих групп, либо допускается только один отказ (т.е. одного GM).

В примере 2 имеется только один резервный ГМ, и поэтому при отсутствии отказов не используется только один ГМ (в отличие от примера 1, где при отсутствии отказов не используются два ГМ). Однако либо один

резервный GM должен выдерживать большую нагрузку в случае отказа обоих активных GM, иначе в любой момент времени можно допустить только один активный отказ.

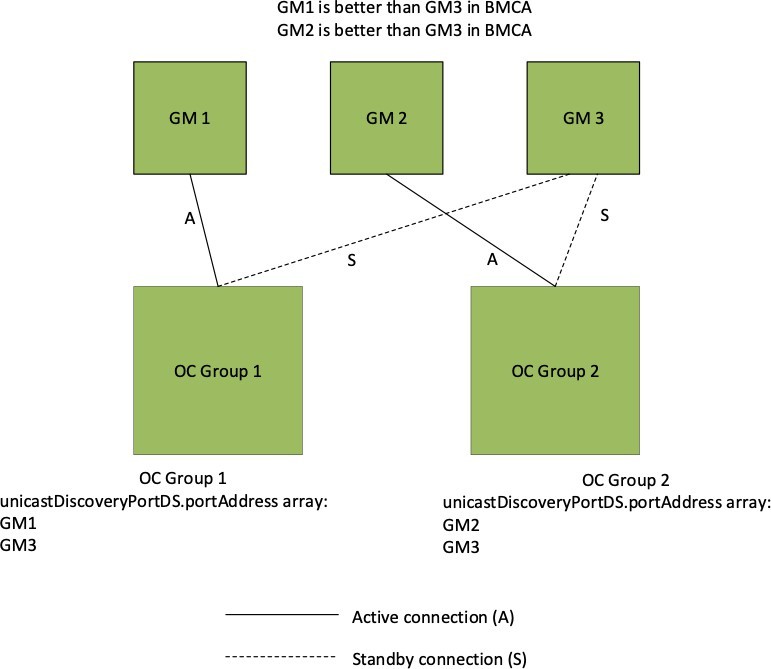


Рисунок 7. Пример 2 один активный ГМ для каждой группы и один резервный ГМ для обеих групп

Пример 1 иллюстрирует случай полного резервирования, когда на каждый активный ГМ приходится один резервный ГМ. Пример 2 иллюстрирует случай частичного резервирования, когда резервных ГМ меньше, чем активных. Чтобы сбалансироватьнагрузку между активными GM, OC должны быть распределены между нимикак можно более равномерно. Для балансировки нагрузки между резервными ГМ, а также для достижения максимальной устойчивости к сбоям резервные ГМ должны быть назначены на равное количество групп OC. Например, если имеется 60000 OC, 12 потенциальных активных GM и 4 потенциальных резервных GM, то OC должны быть разделены на группы по 5000 OC в каждой (т.е. 60000 OC/12 GM), и каждый из 12 потенциальных активных GM должен быть введен в unicastDiscoveryPortDS OC ровно одной группы. Каждый потенциальный резервный GM должен быть введен в unicastDiscoveryPortDS OCs ровно 3 групп (и ни одна группа не должна иметь два резервных GM, введенных в unicastDiscoveryPortDS любой из ее OCs). При таком подходе резервный GM служит резервным для 3 групп OC. В данном примере, если резервный GM может выдержать нагрузку до *N* групп (*N* ≤ 3), то можно допустить *N* активных отказов.

# 12 Приложение II. Примеры конфигурации Active-Active

## II.1 Введение

При использовании A-BMCA из п. 6.14 OC запрашивает сообщения Announce у потенциальных GM в UMT без учета их локальных значений localGmPriority (т.е. потенциально у всех потенциальных GM в UMT). OC может получать сообщения Announce от некоторых или всех ГМ. В любом случае OC использует полученные сообщения Announce в A-BMCA. Если все ГМ имеют одинаковые значения приоритета 1, тактового класса, тактовой точности, local ScaledLogVariance и приоритета 2, то лучший ГМ будет выбран на основе значений localGmPriority. Поскольку значения таблицы localGmPriority в каждом OC могут быть разными, это позволяет разным OC отдавать предпочтение разным GM в разном порядке приоритета, с целью достижения равномерного распределения PTP-соединений между множеством активных GM. Кроме того, если GM выходит из строя, то OC перестают получать сообщения Announce от этого GM. В этом случае OC, для которых этот ГМ имел наивысший локальный приоритет (localGmPriority), выберут в качестве лучшего ГМ ГМ со вторым по значению локальным приоритетом (localGmPriority).

В общем случае, когда имеется *N* ГМ, UMT каждого OC имеет *N* записей ГМ и *N* значений localGmPriority. В этом случае может произойти до *N* - 1 отказов, при этом синхронизация будет по-прежнему обеспечиваться OC (если все GM откажут, то синхронизация перестанет обеспечиваться GM). Можно показать, что равномерное распределение УО между ГМ после каждого из последовательных *N* - 1 отказов может быть получено, если *N* значений localGmPriority различны и если каждый УО имеет различную перестановку *N* различных значений localGmPriority. К сожалению, количество перестановок очень велико даже для умеренного числа ГМ. Для упрощения подхода ГМ и ОС можно разделить на группы, в каждой из которых будет примерно одинаковое количество ГМ и ОС. Это приводит к уменьшению числа ГМ в каждой группе и уменьшению числа перестановок значений параметра localGmPriority для этой группы. Недостатком является то, что при наличии одного или нескольких отказов в группе результирующее равномерное распределение OC между GMs происходит только в этой группе, а не по всем OCs. В результате некоторые ГМ будут задействованы в большей степени, чем другие.

## II.2 Иллюстрация

Данный подход проиллюстрирован здесь для случая кластера из 4 ГМ и 24 групп КО, что позволяет добиться равного количества КО, синхронизируемых каждым ГМ после 3 сбоев (т.е. общее число перестановок четырех объектов равно 24). На рис. II-1 показан кластер, состоящий из 4 ГМ (ГМ 1, ГМ 2, ГМ 3 и ГМ 4) и 24 групп УО (группы с 1 по 24). Количество УО в каждой из 24 групп не имеет значения и зависит от общего количества УО в кластере. Значения UMT localGmPriority для УО в каждой группе приведены в табл. 4. Все возможные перестановки четырех приоритетов ГМ содержатся в 24 записях табл. 4. Это возможно потому, что для 4 ГМ существует 4! = 24 перестановки.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер группы OC** | **Лучший генеральный директор**  **(GM с приоритетом 0)** | **2-й лучший ГМ (ГМ с приоритетом 1)** | **3-й лучший ГМ (ГМ с**  **приоритетом 2)** | **4-й лучший ГМ (ГМ с**  **приоритетом 3)** |
| 1 | 1 | 2 | 3 | 4 |
| 2 | 1 | 2 | 4 | 3 |
| 3 | 1 | 3 | 2 | 4 |
| 4 | 1 | 3 | 4 | 2 |
| 5 | 1 | 4 | 2 | 3 |
| 6 | 1 | 4 | 3 | 2 |
| 7 | 2 | 1 | 3 | 4 |
| 8 | 2 | 1 | 4 | 3 |
| 9 | 2 | 3 | 1 | 4 |
| 10 | 2 | 3 | 4 | 1 |
| 11 | 2 | 4 | 1 | 3 |
| 12 | 2 | 4 | 3 | 1 |
| 13 | 3 | 1 | 2 | 4 |
| 14 | 3 | 1 | 4 | 2 |
| 15 | 3 | 2 | 1 | 4 |
| 16 | 3 | 2 | 4 | 1 |
| 17 | 3 | 4 | 1 | 2 |
| 18 | 3 | 4 | 2 | 1 |
| 19 | 4 | 1 | 2 | 3 |
| 20 | 4 | 1 | 3 | 2 |
| 21 | 4 | 2 | 1 | 3 |
| 22 | 4 | 2 | 3 | 1 |
| 23 | 4 | 3 | 1 | 2 |
| 24 | 4 | 3 | 2 | 1 |

Таблица 4. UMT localGmPriority для OC в каждой группе, для случая 4 ГМ и 24 групп OC.

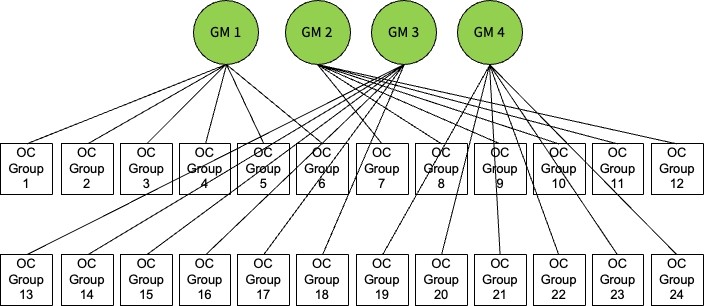


Рисунок 8. Группы OC с 1 по 24 синхронизируются по соответствующему наилучшему приоритету GM, исходя из приоритетов, приведенных в табл. 4.

На рис. 8 также показан ГМ с наилучшим приоритетом для каждой OC-группы. Поскольку в табл. 4 присутствуют все перестановки четырех приоритетов, каждый ГМ синхронизирует 6 OC-групп. В результате происходит равномерное распределение групп OC между ГМ. ГМ 1 синхронизирует группы OC с 1 по 6, ГМ 2 - группы OC с 7 по 12, ГМ 3 - группы OC.

OC Groups 13 - 18, а GM 4 синхронизирует OC Groups 19 - 24.

На рис. 9 показан случай, когда ГМ 1 вышел из строя. На основании табл. 4 группы OC с 1 по 6, которые ранее синхронизировались с помощью GM 1, переключаются на второй по качеству GM в своих UMT. Это GM 2 для групп OC 1 и 2, GM 3 для групп OC 3 и 4 и GM 4 для групп OC 5 и 6. После сбоя каждый оставшийся GM синхронизирует 8 OC-групп (вместо 6 первоначальных), при этом распределение OC- групп по GM остается равномерным, а PTP-соединения одинаково сбалансированы по оставшимся GM. GM 2 синхронизирует OC-группы 1, 2 и 7 до 12, GM 3 синхронизирует OC-группы 3, 4 и 13 до 18, а GM 4

синхронизирует OC-группы 5, 6 и 19 до 24.

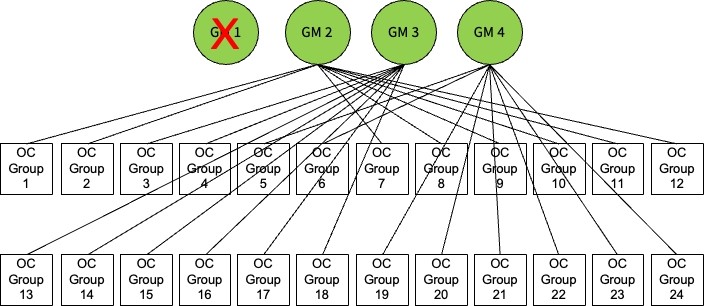


Рисунок 9. Группы OC 1 - 24, синхронизированные соответствующим ГМ с наилучшим приоритетом, исходя

из приоритетов, приведенных в табл. 4, для случая, когда ГМ 1

вышел из строя.

На рис. 10 показан случай, когда отказали ГМ 1 и ГМ 3. Если предположить, что первым отказал ГМ 1, то на основании табл. 4 группы OC 3, 13, 15 и 16, которые ранее синхронизировались ГМ 3 (см. рис. 9), переключаются на ГМ 2. OC

Группы 4, 14, 17 и 18, которые ранее были синхронизированы ГМ 3 (см. рис. 9), переходят на ГМ 4. После сбоя каждый оставшийся GM синхронизирует 12 групп OC, и распределение групп OC по GM остается равномерным. GM 2 синхронизирует OC-группы 1, 2, 3, 7-12, 13, 15 и 16, а GM 4 синхронизирует OC-группы

4, 5, 6, 14 и 17-24. Заметим, что такая же конечная конфигурация получается, если первым выходит из строя ГМ 3, а затем ГМ 1.

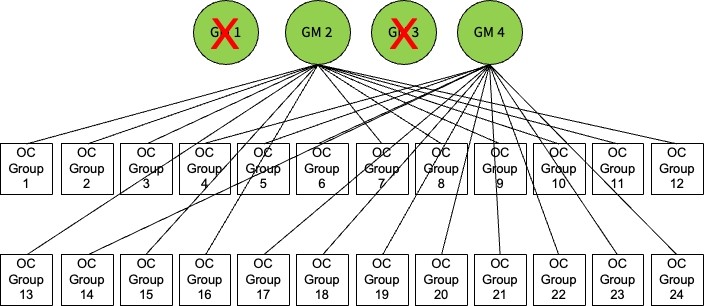


Рисунок 10. Группы OC 1 - 24, синхронизированные соответствующим ГМ с наилучшим приоритетом, исходя из приоритетов, приведенных в табл. 4, для случая, когда ГМ 1 и ГМ 2 вышли из строя.

Из табл. 4 видно, что при отказе 3 из 4 ГМ все группы OC переключаются на оставшийся ГМ, который не отказал. Наконец, видно, что причина равномерного распределения OC-групп по четырем ГМ после отказа заключается в том, что UMT 24 групп содержат все возможные перестановки приоритетов четырех ГМ. Для выполнения этого условия количество групп OC должно быть N! при наличии N ГМ. Разбиение ГМ на множество независимых кластеров необходимо для минимизации сложности реализации.