Internet Engineering Task Force (IETF)
Request for Comments: 8990
Category: Standards Track
ISSN: 2070-1721

C. Bormann
Universität Bremen TZI
B. Carpenter, Ed.
Univ. of Auckland
B. Liu, Ed.
Huawei Technologies Co., Ltd

May 2021

GeneRic Autonomic Signaling Protocol (GRASP)

Базовый протокол автономной сигнализации GRASP

Аннотация

В этом документе определён протокол GRASP, позволяющий автоматическим[1] узлам и автоматическим агентам служб (Autonomic Service Agent или ASA) динамически обнаруживать партнёров для синхронизации состояний и согласования параметров. GRASP зависит от внешней среды защиты, которая описана в отдельных документах. Технические задачи и параметры для конкретных вариантов применения также описываются в отдельных документах. В приложениях кратко рассмотрены требования к протоколу, а также другие протоколы со сравнимыми свойствами.

Статус документа

Документ содержит проект стандарта Internet (Standards Track).

Документ является результатом работы IETF[2] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[3]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc8990.

Оглавление

[1] В переводе используется принятая в русском языке терминология для автоматических и автоматизированных узлов (систем, элементов). Автоматическая система работает без привлечения человека или внешней системы управления, автоматизированная просто выполняет задание (сценарий), заданный человеком или внешней системой управления. Термин «самоуправляемый» в переводе используется как синоним термина «автоматический». *Прим. перев.*

[2] Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

[3] Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

Успех Internet сделал сети IP больше и сложнее. Крупные ISP и сети предприятий становятся все более сложными для управления людьми, эксплуатационные расходы быстро растут. Это ведёт к росту потребности в самоуправлении сетей. Общие аспекты автоматических сетей (Autonomic Network или AN) рассмотрены в [RFC7575] и [RFC7576].

Один из подходов заключается в децентрализации логики управления сетью путём её переноса в элементы сети. Эталонная модель сетей с самоуправлением (AN) на этой основе представлена в [RFC8993]. С этим документом следует ознакомиться для понимания совместной работы различных автоматических элементов. Для самоуправления устройства, в которых реализованы автоматические агенты служб (Autonomic Service Agent или ASA, [RFC7575]), предъявляют свои требования к сигнализации. В частности, им нужно обнаруживать друг друга, синхронизировать состояния, а также согласовывать параметры и ресурсы напрямую. Для типов параметров и ресурсов нет никаких ограничений и они могут включать базовую информацию, нужную для адресации и маршрутизации, а также все остальное, что может настраиваться в традиционных сетях без самоуправления. Неделимый элемент обнаружения, синхронизации или согласования называется технической задачей, т. е. настраиваемым параметром или набором параметров (см. определения в параграфе 2.1).

Процесс согласования выполняется в несколько итераций и требует нескольких обменов сообщениями, образующих замкнутый цикл между согласующимися объектами. Фактически эти объекты являются агентами ASA, которые обычно, но не обязательно, размещаются в разных устройствах. Синхронизацию состояний при необходимости можно считать согласованием без итераций. Согласование и синхронизация логически следуют за обнаружением. Более подробно требования описаны в Приложении B. В параграфе 2.3 рассмотрена модель поведения для протокола, предназначенного для обнаружения, синхронизации и согласования. Устройство протокола GRASP, описано в разделе 2 на основе этой модели поведения. Соответствующие возможности имеющихся протоколов рассмотрены в Приложении C.

Предложенный механизм обнаружения ориентирован на синхронизацию и согласование задач. Он основан на процессе обнаружения соседей по локальному каналу, но поддерживает перенаправление на партнёров в других каналах. Какая-либо определённая топология сети не предполагается. Когда устройство запускается без

подготовленной конфигурации, топология ему неизвестна. Протокол можно применять как в небольшой и/или плоской сети, такой как небольшая сеть дома или в офис, так и в крупной сети с профессиональным управлением. Поэтому механизм обнаружения должен быть способен разрешать устройствам начальную загрузку самостоятельно без каких-либо предварительных допущений о структуре сети.

Поскольку протокол GRASP может применяться в процессе принятия решений между распределенными устройствами или сетями, он должен работать в безопасной среде со строгой проверкой подлинности (аутентификацией).

В реальных системах не все устройства будут поддерживать GRASP, поэтому некоторые агенты ASA будут напрямую управлять группами неавтоматических узлов, а иные узлы без самоуправления будут управляться традиционными способами. Такие варианты смешанного использования в спецификации не рассматриваются.

2. Обзор протокола

2.1. Терминология

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

В документе используются термины, определённые в [RFC7575], а ниже даны определения ещё ряда терминов.

Discovery - обнаружение
: Процесс, в котором агент ASA находит партнёров в соответствии с заданной задачей обнаружения. Результаты обнаружения могут различаться при разных задачах. Обнаруженные партнёры могут позднее участвовать в согласовании или выступать источником при синхронизации данных.

Negotiation - согласование
: Итеративное взаимодействие двух ASA для согласования параметров, подходящих для задач обоих ASA.

State Synchronization - синхронизация состояний
: Процесс взаимодействия агентов ASA для получения текущего состояния параметров, хранящихся в других ASA. Это особый случай согласования, в котором информация передаётся, но агенты ASA не запрашивают у своих партнёров изменение настройки параметров. Все остальные определения применимы как для согласования, так и для синхронизации.

Technical Objective (Objective) - техническая задача (задача)
: Технической задачей является структура данных, основным содержимым которой является имя и значение. Значение состоит из одного параметра конфигурации или набора параметров того или иного типа. Формат задачи определён в параграфе 2.10.1. Задачи применяются в 3 контекстах: обнаружение, согласование и синхронизация. Обычно данная задача не связана сразу с контекстом согласования и синхронизации.
: Один агент ASA может поддерживать несколько независимых задач.
: Параметры в значении данной задачи применяются к соответствующей службе, функции или действию. В принципе они могут быть чем угодно, что сетевой узел может задать конкретным логическим, числовым или строковым значением или более сложной структурой данных. Предполагается, что каждый узел содержит один или несколько агентов ASA, которые сами могут управлять вспомогательными неавтоматическими узлами.
:: ***Discovery Objective - задача обнаружения***
:: Задача обнаружения. Значение может быть неопределённым.
:: ***Synchronization Objective - задача синхронизации***
:: Задача, для которой конкретное техническое содержимое должно синхронизироваться между двумя или несколькими ASA. Таким образом, каждый агент ASA будет поддерживать свою копию задачи.
:: ***Negotiation Objective - задача согласования***
:: Задача, техническое содержание которой должно быть определено в координации с другим ASA. Каждый агент ASA будет поддерживать свою копию задачи.
: Подробное обсуждение задач, включая их формат, приведено в параграфе 2.10.

Discovery Initiator - инициатор обнаружения
: Агент ASA, запускающий обнаружение передачей сообщения Discovery, указывающего конкретную задачу для обнаружения.

Discovery Responder - ответчик при обнаружении (обнаруживаемый)
: Партнёр, который содержит ASA, поддерживающий задачу для обнаружения, указанную инициатором, или кэширующий местоположение агентов ASA, поддерживающих эту задачу. Ответчик передаёт сообщение Discovery Response.

Synchronization Initiator - инициатор синхронизации
: Агент ASA, запускающий синхронизацию отправкой запроса, указывающего конкретную задачу для синхронизации.

Synchronization Responder - ответчик при синхронизации (опорное устройство)
: Агент ASA, отвечающий значением запрошенной задачи.

Negotiation Initiator - инициатор согласования
: Агент ASA, запускающий согласование отправкой запроса, указывающего конкретную задачу для согласования.

Negotiation Counterpart - вторая сторона согласования
: Партнёр, с которым инициатор согласует конкретную задачу.

GRASP Instance - экземпляр GRASP
: Экземпляр протокольной машины GRASP, вероятно включающий потоки или процессы, а также динамические структуры данных, такие как кэш обнаружения, и работающий в данной защищённой среде на одном устройстве.

GRASP Core - ядро GRASP
: Код и общие структуры данных экземпляра GRASP, взаимодействующие с отдельными ASA через подходящий интерфейс прикладных программ (Application Programming Interface или API).

Interface (GRASP Interface) - интерфейс GRASP
: Если явно не указано иное, это обозначает сетевой интерфейс (физический или виртуальный), используемый в настоящее время конкретным экземпляром GRASP. Устройство может иметь другие интерфейсы, не используемые GRASP, а также не участвующие в работе автоматической сети AN.

2.2. Высокоуровневая модель развёртывания

Реализация GRASP будет частью инфраструктуры автоматической сети (Autonomic Networking Infrastructure или ANI) в автоматическом узле, которая должна также обеспечивать подходящую защищённую среду. В соответствии с [RFC8993] этой среде **следует** быть автоматической плоскостью управления (Autonomic Control Plane или ACP) [RFC8994]. В результате все автоматические узлы в ACP могут доверять друг другу. Предполагается, что GRASP будет иметь доступ к ACP через типовой интерфейс программных сокетов и ACP сделает доступными лишь сетевые интерфейсы внутри сети AN. При отсутствии ACP применяются соображения, приведённые в параграфе 2.5.1.

Имеется также один или несколько автоматических агентов служб (ASA). В случае специализированного для одной задачи (single-purpose) устройства эти компоненты могут быть полностью интегрированы с GRASP и ACP. Предполагается, что более распространены будут многоцелевые устройства, содержащие несколько ASA, такие как маршрутизатор или крупный коммутатор. В таких случаях предполагается, что ACP, GRASP и агенты ASA будут реализованы как отдельные процессы, которые могут поддерживать асинхронные и одновременные операции, например, на основе многопоточности.

В некоторых случаях может разворачиваться ограниченная модель согласования на основе отношений частичного доверия, например, между двумя административными доменами. Агенты ASA в этом случае могут обмениваться ограниченной информацией и согласовывать некоторые конкретные конфигурации.

Протокол GRASP разработан специально для работы в одной области адресации. Его механизмы обнаружения и лавинной рассылки не поддерживают автоматические операции через какие-либо трансляторы адресов или прокси вышележащего уровня.

Требуется подходящий интерфейс API между GRASP и агентами ASA. В некоторых реализациях агенты ASA будут работать в пользовательском пространстве с библиотекой GRASP, предоставляющей API, а эта библиотека будет взаимодействовать с ядром GRASP через системные вызовы. Детали интерфейсов API выходят за рамки этого документа. Возможные модели развёртывания подробно описаны в [RFC8993].

Экземпляр GRASP должен знать сетевые интерфейсы, которые он будет использовать, а также соответствующие глобальные и локальные для канала адреса. При наличии ACP такая информация будет доступна из таблицы смежности, описанной в [RFC8993]. В иных случаях протокол GRASP должен сам определить эту информацию. Детали такого определения зависят от устройства и операционной системы. Далее в документе термины «интерфейс» и «интерфейс GRASP» относятся лишь к набору сетевых интерфейсов, используемых конкретным экземпляром GRASP.

Поскольку GRASP нужно работать с очень высокой надёжностью, особенно при начальной загрузке и во время отказов, важно, чтобы каждая реализация продолжала работать в таких условиях. Например, отказам при обнаружении или каким-либо сбоям в сокетах недопустимо приводить к невосстановимым сбоям в протоколе GRASP и протокол должен возвращать подходящий код ошибки через API, чтобы агенты ASA также могли восстановиться.

Недопустима зависимость GRASP от энергонезависимого хранилища. Все условия ошибок при работе и такие события, как смена адресов, отказы сетевых интерфейсов, циклы сна и пробуждения CPU, должны обрабатываться так, чтобы протокол GRASP продолжал корректную и защищённую работу (параграф 2.5.1).

Автоматический узел обычно использует один экземпляр GRASP, который является общим для нескольких агентов ASA. Возможные исключения рассмотрены ниже.

2.3. Высокоуровневое устройство протокола

В этом параграфе рассматривается модель поведения и общее устройство GRASP, поддерживающее обнаружение, синхронизацию и согласование при работе в качестве платформы для разных технических задач.

Базовая платформа

Устройство протокола является достаточно общим и не зависит от содержимого синхронизации или согласования. Техническое содержимое меняется в соответствии с задачами и парами партнёров.

Множество экземпляров

Обычно на автоматическом узле будет использоваться один основной экземпляр протокольной машины GRASP, а каждый агент ASA будет запускать независимый асинхронный процесс. Однако возможен запуск на одном узле нескольких экземпляров GRASP, возможно с разными параметрами защиты (параграф 2.5.2). В таких случаях каждый экземпляр должен независимо прослушивать групповые адреса link-local для GRASP и все экземпляры должны пробуждаться такими групповыми пакетами для корректной работы обнаружения и лавинной рассылки.

Инфраструктура защиты

Как отмечено выше, протокол не имеет встроенных функций защиты и опирается на внешнюю инфраструктуру.

Однотипное обнаружение, синхронизация и согласование

Метод обнаружения, а также методы синхронизации и согласования устроены одинаково и могут комбинироваться, когда это полезно, обеспечивая быстрый режим работы, как описано в параграфе 2.5.4. Эти процессы могут выполняться и независимо.

Для некоторых задач (особенно связанных со службами прикладного уровня) можно применять иные механизмы обнаружения, например, основанные на DNS [RFC7558]. Выбор остаётся за разработчиками отдельных ASA.

Однородный шаблон для технических задач

Объекты синхронизации и согласования определены по однородным шаблонам. Содержащиеся в задачах значения передаются в простом двоичном формате или в виде комплексного объекта. Базовый протокол использует краткое двоичное представление (Concise Binary Object Representation или CBOR) [RFC8949], которое можно легко расширить в соответствии с будущими требованиями.

Гибкая модель для синхронизации

GRASP поддерживает синхронизацию между двумя узлами, которую можно применять неоднократно для синхронизации небольшого числа устройств. Протокол также поддерживает лавинный режим без запроса (unsolicited flooding), когда большой группе устройств (возможно всем автоматическим узлам) нужна одна и та же техническая задача.

Для некоторых параметров сети могут лучше подойти традиционные механизмы лавинной рассылки, такие как протокол DNCP (Distributed Node Consensus Protocol) [RFC7787]. GRASP может сосуществовать с DNCP.

Простая модель «инициатор-ответчик» для согласования

Многочисленные согласования очень сложно моделировать и сходимость их не гарантируется. GRASP использует простую двухстороннюю модель и может косвенно поддерживать многостороннее согласование.

Организация контекста согласования или синхронизации

Технические задачи, передаваемые GRASP, организуются в соответствии с функцией или службой. Задачи разных функций и служб хранятся раздельно, поскольку они могут согласовываться и синхронизироваться разными сторонами или иметь разные времена откликов. Обычно один агент ASA управляет небольшим набором тесно связанных задач с версией этого ASA в каждом соответствующем автоматическом узле. Более подробное рассмотрение этого вопроса выходит за рамки документа.

Запросы и отклики в процедурах согласования

Инициатор может согласовать конкретную задачу с соответствующими агентами ASA. Он может запросить относящуюся к делу информацию для корректировки своей локальной конфигурации, а также попросить партнёра установить соответствующую конфигурацию. Инициатор также может запросить моделирование или предсказание результатов, отправляя некоторые условия пробного прогона (dry-run).

Кроме традиционного отклика «да» или «нет» ответчик может возвратить значение предлагаемого варианта для соответствующей задачи. Это приведёт к началу двухстороннего согласования, завершающегося компромиссом между парой ASA.

Схождение процедур согласования

Для обеспечения сходимости, когда ответчик предлагает новое значение или условие в ответе при согласовании, такому предложению следует быть как можно ближе к исходному запросу или предыдущему предложению. Предлагаемое на следующем этапе согласования значение следует выбирать между парой значений двух предшествующих этапов. GRASP предоставляет механизмы, гарантирующие схождение (или отказ) за небольшое число шагов, ограниченное тайм-аутом или максимальным числом итераций.

Расширяемость

Протокол GRASP осознанно не имеет номера версии и может быть расширен путём добавления новых типов сообщений или опций. Сообщение Invalid (M_INVALID) применяется для обозначения нераспознанных сообщений или опций, переданных другой реализацией. При обычном использовании новая семантика добавляется путём определения новых задач для синхронизации или согласования.

2.4. Краткий обзор операций

Предполагается, что экземпляр GRASP работает как отдельный модуль ядра, обеспечивающий интерфейс API (такой как [RFC8991]) для различных агентов ASA. Эти ASA могут работать без специальных привилегий, если они не требуются по иным причинам (таким как настройка адресов IP или манипуляции с таблицами маршрутизации).

Используемые ASA механизмы GRASP строятся на основе задач GRASP, определённых как структуры данных с административной информацией, такой как уникальные имена задач и их текущие значения. Протокол не ограничивает формат и размер значений за исключением того, что они должны быть пригодны для последовательной передачи в CBOR, и на практике это ограничений не задаёт.

GRASP обеспечивает несколько механизмов, перечисленных ниже.

- Механизм обнаружения (M_DISCOVERY, M_RESPONSE), позволяющий агенту ASA найти другие ASA, которые поддерживают данную задачу.
- Механизм запроса согласования (M_REQ_NEG), который позволяет ASA запустить согласования задачи с другим ASA. После запуска согласования процесс становится симметричным и имеется сообщение пошагового согласования (M_NEGOTIATE) для каждого ASA. Для согласования поддерживаются также функции M_WAIT, M_END.
- Механизм синхронизации (M_REQ_SYN), с помощью которого агент ASA может запросить текущее значение объекта у другого ASA. Функция M_SYNCH позволяет ASA ответить на запрос синхронизации.
- Лавинный механизм (M_FLOOD) позволяет агенту ASA разослать текущее значение задачи через сеть AN так, что любой ASA может получить его. Одним из применений являются анонсы, избавляющие от необходимости обнаруживать широко применяемые задачи.

Некоторые примеры сообщений и простого потока представлены в Приложении A.

2.5. Базовые свойства и механизмы GRASP

2.5.1. Требование внешнего механизма защиты

Протокол GRASP не задаёт транспортную защиту, поскольку он предназначен для применения в разных средах. Каждое решение по адаптации GRASP **должно** задавать «подложку» защиты и транспорта, используемую GRASP в этом решении. Подложка **должна** обеспечивать передачу и приём сообщений GRASP лишь между членами группы с взаимным доверием, в которой работает GRASP. Каждый член такой группы является экземпляром GRASP и узлом связанного графа. Группа и граф создаются подложкой защиты и транспорта и называется доменом GRASP. Подложка должна поддерживать индивидуальные (unicast) сообщения между любыми членами группы и групповые (link-local) сообщения между смежными членами. Сообщения между членами группы и не входящими в группу элементами должны отвергаться. В этой модели защита обеспечивается через включение в группу, но любой член группы доверия может атаковать всю сеть, пока не будет отозван (исключён) из группы.

Подложка **должна** применять криптографическую аутентификацию членов группы и контролировать целостность сообщений GRASP. Это может быть реализовано в домене сквозным или поэтапным (hop by hop) способом. Подложка защиты и транспорта **должна** обеспечивать механизмы удаления из группы потерявших доверие членов. Если подложка не требует и не применяет шифрования сообщений GRASP, любая служба, применяющая GRASP в таком решении, **должна** обеспечивать защиту и шифрование элементов сообщения, раскрытие которых может создавать угрозу атаки.

Подложкой защиты и транспорта для GRASP в инфраструктуре ANI служит плоскость управления ACP. Если явно не сказано иное, далее в документе предполагается именно эта подложка. ACP требует использовать шифрование,

поэтому GRASP в инфраструктуре ANI может считать свои сообщения шифрованными. Доменом GRASP служит ACP, где все узлы автоматического домена соединены шифрованными виртуальными каналами. ACP использует для сообщений защиту hop-by-hop (проверка подлинности и шифрование). Для удаления узлов применяется стандартный отзыв сертификатов PKI или завершение срока действия для краткосрочных сертификатов. Более подробное описание приведено в [RFC8994].

Как отмечено в параграфе 2.3, некоторые операции GRASP по взаимному соглашению могут выполняться через границу административного домена без использования ACP. Такие операции **должны** быть ограничены отдельным экземпляром GRASP со своей копией всех структур данных GRASP, работающим в отдельном домене GRASP со своей подложкой защиты и транспорта. В наиболее простом случае каждое междоменное соединение GRASP (точка-точка) может использовать свой домен, а подложка защиты и транспорта может создаваться на основе протокола защиты транспортного или сетевого уровня. Этот вопрос оставлен для будущих спецификаций.

Имеются исключения из правил для подложки защиты и транспорта в сильно ограниченных подмножествах GRASP, рассчитанных на организацию подложки защиты и транспорта, описанную в следующем параграфе.

2.5.2. DULL GRASP

Некоторым службам может потребоваться использование незащищённых сообщений GRASP для обнаружения, откликов и лавинной отправки без возможности использования имеющихся защищённых связей как часть процесса организации защищённых связей, таких как защищённая подложка GRASP.

Такие операции не защищены по своей природе и их необходимо настраивать для работы на локальном канале (link-local), чтобы минимизировать риск вредоносных воздействий. Возможные примеры включают обнаружение кандидатов в соседи по ACP [RFC8994], прокси начальной загрузки [RFC8995], службы инициализации в сетях, использующих GRASP, но не являющихся полностью автоматическими (т. е. без ACP). Такое использование **должно** ограничиваться операциями на локальном канале одного интерфейса, а также **должно** ограничиваться одним незащищённым экземпляром GRASP со своей копией всех структур данных GRASP. Этот экземпляр называют DULL (Discovery Unsolicited Link-Local).

Правила для экземпляра DULL приведены ниже.

- Инициатор может передавать групповые (link-local) сообщения Discovery или Flood Synchronization, в которых должно быть задано значение счётчика интервалов пересылки (loop count) 1 для предотвращения операций вне локального канала (off-link). Передача незапрошенных сообщений GRASP иных типов **недопустима**.
- Ответчик **должен** отбрасывать без уведомления все сообщения, где счётчик интервалов отличается от 1.
- Ответчик должен отбрасывать без уведомления все сообщения для задачи GRASP, не являющейся напрямую частью частью службы, требующей этого незащищённого режима.
- Ответчику **недопустимо** ретранслировать групповые сообщения.
- Сообщение Discovery Response **должно** указывать адрес link-local.
- В сообщение Discovery Response **недопустимо** включать опцию Divert.
- Узел **должен** отбрасывать без уведомления все сообщения, отправленные с адреса, отличного от link-local.

Для минимизации доступности трафика посторонним **следует** снижать трафик GRASP, используя лишь сообщения Flood Synchronization для анонсирования задач и соответствующих локаторов (адресов) вместо отправки сообщений Discovery и Discovery Response. Детали этого выходят за рамки этого документа.

2.5.3. Использование транспортного уровня

Все сообщения GRASP после их преобразования в строку байтов CBOR передаются с использованием выбранного транспортного протокола. Транспортные протоколы для домена GRASP определяет подложка защиты и транспорта, как описано в параграфе 2.5.1.

Сообщения GRASP для обнаружения и лавинной отправки разработаны для лавинной передачи (flooding) в масштабе домена GRASP с использованием поэтапной (hop-by-hop) групповой (link-local) пересылки между смежными узлами GRASP. Подложка защиты и транспорта GRASP нужна для задания способа транспортировки групповых пакетов link-local. Это может быть транспорт без гарантии доставки (UDP), но **следует** выбирать надёжный транспорт (например, TCP). Если подложка задаёт для сообщений обнаружения и лавинной передачи транспорт без гарантии доставки, такой как UDP, **недопустимо** использовать фрагментацию IP по причине возможных потерь, особенно в случае лавинной рассылки через несколько этапов. Поэтому протокол GRASP **должен** на уровне пользовательского API задать предельный размер сообщений при обнаружении и лавинной отправке, чтобы не возникало фрагментации. Для транспорта IPv6 это означает, что размер пакетов IPv6 с такими сообщениями должен быть не более 1280 байтов (если нет информации о том, что MTU во всем домене GRASP имеет большее значение).

Все остальные сообщения GRASP передаются индивидуально между членами группы из домена GRASP. Для них **должен** применяться транспорт с гарантией доставки, поскольку протокол GRASP не поддерживает обнаружения ошибок, повтора передачи и управления потоком данных. Если подложка защиты и транспорта не задаёт иное, **должен** применяться протокол TCP.

Подложкой защиты и транспорта для GRASP в инфраструктуре ANI является плоскость управления ACP. Если явно не указано иное, в этом документе при описании сообщений GRASP предполагается такая подложка. В ACP для индивидуальных сообщений GRASP применяется протокол TCP. Сообщения GRASP для обнаружения и лавинной передачи также используют TCP, эти сообщения link-local пересылаются путём репликации всем смежным узлам GRASP на канале через соединения TCP с этими узлами GRASP. Поэтому GRASP в ANI не ограничивает размер сообщений для обнаружения и лавинной отправки из за фрагментации. При создании ACP с использованием экземпляра DULL GRASP, применяется естественная групповая передача UDP для обнаружения соседей ACP и GRASP на каналах.

Для групповых (link-local) пакетов UDP протокол GRASP прослушивает стандартный порт GRASP Listen (параграф 2.6). Транспортные соединения для обнаружения и лавинной передачи на узлах-трансляторах завершаются в экземплярах GRASP (например, GRASP ASA) так, что групповая (link-local) адресация с поэтапной лавинной рассылкой сообщений M_DISCOVERY и M_FLOOD, а также поэтапная пересылка откликов M_RESPONSE и их кэширование на пути работают корректно.

Индивидуальные транспортные соединения, используемые для синхронизации и согласования, могут завершаться непосредственно в агентах ASA, реализующих задачи, поэтому такому трафику не нужно проходить через экземпляры GRASP. Для этого ASA прослушивают свой динамически выделенный порт, который сообщается партнёру в процессе обнаружения. Можно завершать индивидуальные транспортные соединения в экземплярах GRASP и передавать трафик агенту ASA или от него, если это удобней для реализации (например, для отвязывания агентов ASA от сетевых соединений).

2.5.4. Механизм и процедуры обнаружения

2.5.4.1. Разделение механизмов обнаружения и согласования

Хотя обнаружение и согласование или синхронизация определены с GRASP, это отдельные механизмы. Процесс обнаружения может работать независимо от процесса согласования или синхронизации. При получении сообщения Discovery (параграф 2.8.4) узлу следует передать в ответ Discovery Response, в котором указать себя как ответчика при обнаружении или перенаправить инициатора к более подходящему агенту ASA. Однако этот отклик можно отложить, если получателю нужно ретранслировать сообщение Discovery дальше, как описано в параграфе 2.5.4.4.

За действием по обнаружению (M_DISCOVERY) обычно следует действие для согласования (M_REQ_NEG) или синхронизации (M_REQ_SYN). Результат обнаружения может применяться протоколом согласования для выбора ASA, с которым инициатор будет выполнять согласование.

Инициатор обнаружения для данной задачи не обязан реагировать на эту задачу как участник согласования, ответчик при синхронизации или источник лавинной передачи. Например, ASA может выполнить обнаружение даже если он хочет выступать лишь инициатором синхронизации или согласования. Такому агенту ASA не требуется отвечать на сообщения Discovery.

Обнаружение GRASP можно применять без последующего согласования или синхронизации. В этом случае обнаруженная задача просто используется как имя в процессе обнаружения, а последующие операции между партнёрами выполняются без участия GRASP.

2.5.4.2. Обзор процесса обнаружения

Полный процесс обнаружения начинается с передачи группового сообщения Discovery (M_DISCOVERY) в локальный канал. Поддерживающие задачу соседи по локальному каналу будут напрямую отвечать сообщениями Discovery Response (M_RESPONSE). Соседи с несколькими интерфейсами могут отвечать кэшированными сообщениями Discovery Response. При отсутствии кэшированного отклика такой сосед будет транслировать сообщение Discovery в другие свои интерфейсы GRASP. Если получивший ретранслированное сообщение Discovery узел поддерживает задачу для обнаружения, он будет отвечать на полученное сообщение Discovery. При наличии кэшированного отклика будет он передаваться в ответ, а при отсутствии будет повторяться ретрансляция другим узлам. Счётчик интервалов и тайм-аут обеспечивают завершение итерационного процесса. Дополнительные детали приведены в параграфе 2.5.4.4.

Сообщение Discovery **может** передаваться по индивидуальному адресу соседу и тому **следует** обрабатывать сообщение как в случае групповой передачи, за исключением того, что при использовании TCP отклик будет в том же сокете, который принял запрос. Однако этом метод не гарантирует успешной доставки в общем случае.

2.5.4.3. Процедуры обнаружения

Обнаружение запускается как операция на канале (on-link). Опция Divert может предложить инициатору связаться для обнаружения с расположенным вне канала агентом ASA. Если подложка защиты и транспорта в домене GRASP (параграф 2.5.3) использует групповую адресацию UDP link-local, инициатор передаёт сообщение по групповому адресу ALL_GRASP_NEIGHBORS (link-local, параграф 2.6), который должны прослушивать все узлы GRASP, чтобы выступать при обнаружении в качестве ответчиков. Поскольку этот порт уникален на устройстве, это относится к функциям экземпляра GRASP, а не отдельных ASA. В результате каждому агенту ASA нужно зарегистрировать поддерживаемые им задачи в локальном экземпляре GRASP.

Если ASA в соседнем устройстве поддерживает запрошенную задачу для обнаружения, устройству **следует** отвечать на групповой пакет индивидуальным сообщением Discovery Response (параграф 2.8.5) с опциями адреса (locator, параграф 2.9.5), если адрес не является временно недоступным. В ином случае при наличии у соседа кешированных данных об агенте ASA, поддерживающем задачу для обнаружения (найденную раньше), ему **следует** отвечать сообщением Discovery Response с опцией Divert, указывающей нужного ответчика. Однако **не следует** возвращать кэшированный отклик через интерфейс, получивший эти сведения, поскольку искомый партнёр будет отвечать напрямую, если он продолжает работать.

Если у устройства нет сведений о запрошенной задаче и он не является ретранслятором при обнаружении (см. параграф 2.5.4.4), он **должен** просто отбросить сообщение Discovery.

Инициатор обнаружения **должен** установить для процесса обнаружения разумный тайм-аут. Предлагается использовать значение 100 мсек, умноженное на счётчик интервалов (loop count), встроенный в задачу. Если в течение заданного времени не получено сообщение Discovery Response, **можно** повторить Discovery с генерацией нового Session ID (параграф 2.7). Следующие повторы **следует** выполнять с экспоненциальным ростом интервала для снижения нагрузки в пиковые периоды. Детали алгоритма увеличения интервалов зависят от варианта применения для искомой задачи, но **должны** соответствовать рекомендациям [RFC8085] для объёмного группового трафика. Частые повторы могут быть признаком DoS-атаки (denial-of-service).

После того, как устройство GRASP обнаружит локатор (адрес) ответчика, поддерживающего искомую задачу, ему **следует** кэшировать эту информацию, включая индекс интерфейса [RFC3493], через который она получена. Эта кэш-запись **может** использоваться в будущем для согласования или синхронизации, а локатор **следует** передавать, когда

это уместно, в опции Divert инициаторам других обнаружений. Механизм кэширования **должен** включать срок действия каждой записи. Это время выводится из параметра ttl в каждом сообщении Discovery Response. Кэшированные записи **должны** игнорироваться или удаляться по истечении срока действия. В некоторых средах могут незапланированно меняться адреса. В таких случаях срок действия записей **следует** делать коротким по сравнению со средним сроком действия адресов. Механизм обнаружения должен отслеживать текущий адрес узла, чтобы сообщения Discovery Response всегда указывали корректные адреса.

Если для одной задачи найдено несколько ответчиков, **следует** кэшировать все, если это не ведёт к нехватке ресурсов. Метод выбора ответчиков для кэширования зависит от реализации и **должен** быть доступен каждому агенту ASA, а реализации GRASP **следует** задавать принятый по умолчанию выбор.

Поскольку кэш для ответчиков при обнаружении имеет конечный размер, записи могут удаляться в любой момент. В таких случаях потребуется повторять обнаружения. Если агент ASA по какой-либо причине завершает работу, его локатор может некоторое время сохраняться в кэше, а попытки соединения будут приводить к отказу. Агенты ASA должны быть устойчивы к таким отказам.

2.5.4.4. Ретрансляция при обнаружении

Экземпляр GRASP с несколькими интерфейсами канального уровня (обычно работает как маршрутизатор) **должен** поддерживать обнаружение на всех интерфейсах GRASP. Такие экземпляры называются ретрансляторами (relaying instance). Экземпляры DULL (параграф 2.5.2) всегда имеют лишь 1 интерфейс, поэтому им **недопустимо** ретранслировать сообщения при обнаружении.

Если ретранслятор получает на данном интерфейсе сообщение Discovery для конкретной задачи, которую он не поддерживает и для которой в данный момент нет кэшированного отклика, он **должен** транслировать запрос, передав новое сообщение Discovery по групповому адресу link-local через другие свои интерфейсы GRASP. Ретранслируемое сообщение Discovery **должно** иметь такой же идентификатор Session ID и поле initiator, что и в принятом сообщении (см. параграф 2.8.4). Адрес IP в поле initiator служит лишь для устранения неоднозначности Session ID и никогда не применяется для адресации сообщений Response. Пакеты откликов передаются ретранслирующему экземпляру, а не исходному инициатору.

Сообщение M_DISCOVERY не кодирует транспортный адрес инициатора или ретранслятора. Поэтому пакеты откликов должны передаваться по адресу транспортного уровня в соединении, где получено сообщение M_DISCOVERY. Если сообщение M_DISCOVERY транслировалось через надёжное поэтапное (hop-by-hop) транспортное соединение, отклик просто возвращается в то же соединение.

Если ретрансляция M_DISCOVERY была групповой (link-local, например, UDP), отклик возвращается через надёжное транспортное соединение hop-by-hop с тем же номером порта отправителя в групповой передаче link-local. Поэтому , если применяется групповая передача link-local и требуются сообщения M_RESPONSE (это происходит практически во всех экземплярах GRASP, за исключением ограниченного применения экземпляров DULL в инфраструктуре ANI), протокол GRASP должен быть способен связать номер порта UDP, откуда были отправлены исходные групповые (link-local) сообщения M_DISCOVERY с тем же номером порта гарантированного транспорта hop-by-hop (например, TCP), чтобы иметь возможность отвечать на транспортные соединения ответчиков, которые хотят передать сообщения M_RESPONSE назад. Отметим, что это не обязательно порт GRASP_LISTEN_PORT.

Ретранслирующий экземпляр **должен** уменьшить значение loop count в задаче и ему **недопустимо** транслировать сообщение Discovery если значение счётчика снизилось до 0. Ретранслятор также **должен** ограничивать общую скорость трансляции сообщений Discovery разумным значением, чтобы смягчить возможные DoS-атаки. Например, в качестве предельной скорости можно установить кратное небольшому числу значение наблюдаемой скорости сообщений Discovery в процессе нормальной работы. Транслирующий экземпляр **должен** кэшировать значение Session ID и адрес инициатора из каждого транслируемого сообщения Discovery, пока не будет получено сообщение Discovery Response или процесс обнаружения не завершится по тайм-ауту. Чтобы не возникало петель, **недопустимо** транслировать сообщения Discovery с данными Session ID и адресом инициатора более одного раза. Это предотвратит петли при обнаружении и смягчит возможную перегрузку.

Поскольку ретранслятор не знает значения тайм-аута, установленного исходным инициатором, ему **следует** устанавливать подходящий тайм-аут для транслируемого сообщения Discovery. Предлагается использовать значение 100 мсек, умноженное на остающееся значение счётчика интервалов (loop count).

Результаты обнаружения, полученные транслирующим экземпляром, **должны** передаваться как сообщение Discovery Response в ответ на вызвавшее ретрансляцию сообщение Discovery.

2.5.4.5. Быстрый режим - обнаружение с согласованием или синхронизацией

Сообщение Discovery **может** включать опцию задачи, что позволяет применять быстрый режим согласования (параграф 2.5.5.1) или синхронизации (параграф 2.5.6.3). Быстрый режим в настоящее время ограничен одной задачей для простоты проектирования и реализации. В будущем возможны расширения, позволяющие использовать в быстром режиме несколько задач для повышения эффективности.

2.5.5. Процедуры согласования

Инициатор согласования создаёт транспортное соединение с агентом ASA на другой стороне, используя адрес, протокол и порт, полученные при обнаружении. Затем он передаёт запрос согласования (используя M_REQ_NEG), включая в него конкретную задачу для согласования. Можно запросить у партнёра создание конкретной конфигурации, а также запросить то или иное моделирование или предсказание результата, передав пробную (dry-run) конфигурацию. Детали, включая отличие между пробным запуском и реальным изменением конфигурации, будут описаны отдельно для каждого типа задач согласования. Любое состояние, связанное с пробным прогоном, такое как временное резервирование ресурсов для последующего применения в реальной работе, полностью зависит от разработчика соответствующего агента ASA.

Для каждого сеанса согласования в целом применяется тайм-аут (по умолчанию GRASP_DEF_TIMEOUT мсек, см. параграф 2.6), инициализируемый при отправке запроса (параграф 2.8.6). Если в течение заданного времени не получен какой-либо отклик, запрос согласования **можно** повторить с новым Session ID (параграф 2.7). При

последующих повторах интервал **следует** экспоненциально увеличивать. Детали алгоритма роста интервалов зависят от варианта применения и задачи запроса.

Если партнёр может сразу же применить запрошенную конфигурацию, он будет незамедлительно давать положительный ответ (O_ACCEPT) в сообщении Negotiation End (M_END), который завершает фазу согласования. В иных случаях согласование продолжается с использованием M_NEGOTIATE. Партнёр может ответить предлагаемой конфигурацией, которую он может применить (обычно с использованием меньших по сравнению с запросом ресурсов). Далее происходит взаимное согласования с использованием сообщений Negotiate (M_NEGOTIATE), пока между агентами ASA не будет достигнут компромисс.

Процедура согласования завершается, когда один из партнёров отправит сообщение Negotiation End (M_END) с опцией Accept (O_ACCEPT) или Decline (O_DECLINE), которое не требует отклика. Согласование может также завершиться по сбою (эквивалентно Decline), если возникнет тайм-аут или счётчик интервалов достигнет 0. При завершении по любой причине транспортное соединение **следует** закрыть. Отказ в транспортной сессии считается сбоем согласования.

Процедура согласования относится к одной задаче и одному партнёру. Обе стороны согласования могут в то же время выполнять согласование с другими ASA или согласовывать между собой другую задачу. Таким образом, предполагается использование GRASP в многопотоковом режиме или его логическом эквиваленте. У некоторых задач согласования могут быть ограничения на использование нескольких потоков, например, для предотвращения чрезмерного выделения ресурсов.

Некоторые конфигурационные действия, например, переключение длины волны в оптической сети, могут выполняться достаточно долго. Соответствующие агенты ASA должны учитывать это, а протокол GRASP позволяет при необходимости вносить задержку в процесс согласования (параграф 2.8.9, M_WAIT).

2.5.5.1. Быстрый режим

Сообщение Discovery **может** включать опцию Negotiation Objective. Это соответствует отправке инициатором последовательности M_DISCOVERY за которой сразу следует M_REQ_NEG. Это влияет на ядро GRASP, поскольку оно должно аккуратно передавать содержимое опции Negotiation Objective агенту ASA, чтобы он мог оценить задачу напрямую. При наличии опции Negotiation Objective агент ASA отвечает сообщением M_NEGOTIATE (или M_END с O_ACCEPT, если предложение сразу подходит ему) вместо M_RESPONSE. Если принявший узел не поддерживает быстрый режим, обнаружение происходит как обычно.

Сообщение Discovery Response может прийти от ответчика, не поддерживающего быстрый режим, до того, как прибудет сообщение Negotiation. В этом случае быстрый режим не сработает.

Этот быстрый режим может сократить взаимодействие между узлами, что позволит повысить эффективность. Однако в сети, где быстрый режим поддерживает лишь часть узлов, поведение усложняется и может зависеть от времени. Поэтому функцию быстрого согласования по умолчанию **следует** отключать.

2.5.6. Синхронизация и процедуры лавинной рассылки

2.5.6.1. Синхронизация по индивидуальным адресам

Инициатор синхронизации создаёт транспортное соединение с агентом ASA на другой стороне, используя адрес, протокол и порт, полученные при обнаружении. Затем он передаёт партнёру сообщение Request Synchronization (M_REQ_SYN, параграф 2.8.6), указывающую конкретную задачу для синхронизации. Партнёр отвечает сообщением Synchronization (M_SYNCH, параграф 2.8.10) с текущим значением синхронизируемой задачи. Других сообщений не требуется и транспортное соединение **следует** закрыть. Отказ в транспортном соединении считается сбоем синхронизации.

Если не получено никакого отклика в течение заданного времени (по умолчанию GRASP_DEF_TIMEOUT мсек, параграф 2.6), запрос синхронизации **можно** повторить с новым Session ID (параграф 2.7). При последующих повторах интервал между ними **следует** экспоненциально увеличивать. Детали алгоритма увеличения интервалов зависят от варианта применения для соответствующей задачи.

2.5.6.2. Лавинная рассылка

В только что описанном случае обмен сообщениями происходит по индивидуальным адресам и связан лишь с одной задачей для синхронизации. Для больших групп узлов, которым нужно синхронизировать одни данные, можно применять лавинную синхронизацию. Для этого инициатор **может** передать незапрошенное сообщение Flood Synchronization (параграф 2.8.11) с одной или несколькими опциями Synchronization Objective, если спецификация соответствующих задач разрешает это. Сообщение передаётся по групповому адресу ALL_GRASP_NEIGHBORS (параграф 2.6). Приём групповых лавинных сообщений является функцией ядра GRASP, как при групповом обнаружении (параграф 2.5.4.3).

Чтобы при лавинной передаче не возникало петель, отправитель сообщения Flood Synchronization **должен** установить подходящее значение счётчика интервалов (loop count) в задачах (по умолчанию GRASP_DEF_LOOPCT). Также нужен специальный механизм для предотвращения избыточного группового трафика, которых должен определяться как часть спецификации соответствующих задач для синхронизации. Это может быть простое ограничение скорости или более сложный механизм вроде алгоритма Trickle [RFC6206].

Устройство GRASP с несколькими интерфейсами канального уровня (обычно маршрутизатор) **должно** поддерживать лавинную синхронизацию на всех интерфейсах GRASP. При получении группового сообщения Flood Synchronization на данном интерфейсе оно **должно** транслироваться как групповое (link-local) сообщение Flood Synchronization во все остальные интерфейсы GRASP. Транслируемое сообщение **должно** иметь то же значение Session ID, что и принятое сообщение, а также **должно** помечаться IP-адресом исходного инициатора.

Лавинная рассылка на канальном уровне поддерживается GRASP путём установки для loop count значения 1 и передачи с адресом отправителя link-local. Лавинная передача с адресом отправителя link-local и счётчиком интервалов, отличным от 1 недопустима и такие сообщения **должны** отбрасываться.

Трансляторы **должны** уменьшать loop count в первой задаче и **недопустимо** транслировать сообщение Flood Synchronization при достижении счётчиком 0. Трансляторы также **должны** ограничивать скорость ретрансляции Flood Synchronization разумным значением для смягчения возможных DoS-атак. Например, скорость можно ограничить значением наблюдаемой скорости лавинной передачи, умноженным на небольшой коэффициент. Транслятор должен кэшировать Session ID и адрес инициатора для каждого ретранслируемого сообщения Flood Synchronization на время, не превышающее 2*GRASP_DEF_TIMEOUT мсек. Для предотвращения петель **недопустимо** транслировать сообщения Flood Synchronization с кэшированными значениями Session ID и адреса инициатора более 1 раза. Это позволит избежать петель при синхронизации и смягчит возможную перегрузку.

Этот механизм надёжен в случае спящих узлов, новых узлов, присоединяющихся к сети, а также узлов, возвращающихся в сеть после отказа. Агенту ASA, инициирующему лавинную передачу, **следует** повторять её с подходящей частотой, которая **должна** соответствовать рекомендациям [RFC8085] для группового трафика с невысоким объёмом. ASA также **следует** выступать ответчиком при синхронизации для соответствующих задач. Таким образом, узлы, которым нужна задача, вовлечённая в лавинную передачу, могут подождать следующей лавинной отправки или запросить индивидуальную (unicast) синхронизацию для этой задачи.

Для групповых сообщений лавинной синхронизации применимы правила безопасности параграфа 2.5.1. На практике это означает, что такие сообщения **недопустимо** передавать и они **должны** игнорироваться при получении, если не нет работающей плоскости управления ACP или эквивалентной по силе защиты. Однако по причине слабой защиты групповых сообщение link-local (раздел 3), задачам, участвующим в лавинной синхронизации, **не следует** включать нешифрованных приватных данных, а принимающему агенту ASA следует проверять действительность этих задач.

2.5.6.3. Быстрый режим (связывание обнаружения и синхронизации)

Сообщение Discovery **может** включать опцию Synchronization Objective. В этом случае сообщение Discovery выступает одновременно как Request Synchronization для индикации ответчику при обнаружении возможности ответить инициатору обнаружения напрямую сообщением Synchronization (параграф 2.8.10) с данными синхронизации для быстрой обработки, если задача для обнаружения поддерживает соответствующую задачу для синхронизации. Влияние организации сети аналогично описанному в параграфе 2.5.5.1.

Возможны случаи, когда сообщение Discovery Response будет приходить от ответчика, не поддерживающего быстрый режим, до прихода сообщения Synchronization. В этом случае быстрый режим не сработает.

Этот быстрый режим может сократить взаимодействие между узлами, что позволит повысить эффективность. Однако в сети, где быстрый режим поддерживает лишь часть узлов, поведение усложняется и может зависеть от времени. Поэтому функцию быстрой синхронизации по умолчанию **следует** отключать и **можно** включать или отключать через Intent.

2.6. Константы GRASP

ALL_GRASP_NEIGHBORS

Групповой адрес с зоной действия на локальном канале (link-local), используемый поддерживающими GRASP устройствами для обнаружения соседей с поддержкой GRASP (т. е. включённых в канал - on-link). Все поддерживающие GRASP устройства являются членом этой группы.

- Групповой адрес IPv6 - ff02::13.
- Групповой адрес IPv4 - 224.0.0.119.

GRASP_LISTEN_PORT (7017)

Общеизвестный пользовательский порт UDP которое каждое устройство с поддержкой GRASP **должно** прослушивать на предмет групповых (link-local) сообщений при использовании экземпляром GRASP протокола UDP для сообщений M_DISCOVERY или M_FLOOD. Этот порт **можно** также использовать для прослушивания индивидуальных сообщений TCP или UDP в простых реализациях GRASP (параграф 2.5.3).

GRASP_DEF_TIMEOUT (60000 мсек)

Принятый по умолчанию тайм-аут, используемый для определения незавершённых операций (отказов).

GRASP_DEF_LOOPCT (6)

Принятое по умолчанию значение счётчика интервалов (loop count), используемое для определения незавершённых операций и предотвращения петель.

GRASP_DEF_MAX_SIZE (2048)

Принятый по умолчанию максимальный размер сообщения в байтах.

2.7. Идентификатор сессии (Session ID)

Идентификатором сессии служит 32-битовое необрабатываемое (opaque) значение, используемое для различения сессий между парой устройств. Новое значение Session ID **должно** создаваться инициатором для каждого нового сообщения Discovery, Flood Synchronization, Request. Все отклики и последующие сообщения в той же процедуре обнаружения, синхронизации или согласования **должны** применять одно значение Session ID.

Для Session ID **следует** обеспечивать очень низкую вероятность локальных конфликтов. Идентификаторы **должны** создаваться генератором псевдослучайных чисел (pseudorandom number generator или PRNG) с использованием созданной локально затравки (seed), которая вряд ли будет использована другим устройством в той же сети. PRNG **следует** быть криптостойким [RFC4086]. При выделении нового Session ID протокол GRASP **должен** проверять, не используется ли это значение, а также **следует** проверить, что оно не применялось недавно (просмотр кэша сессий). В маловероятном случае конфликта GRASP **должен** создать новое значение идентификатора. Однако имеется ненулевая вероятность того, что два узла сгенерируют одно значение Session ID. Поэтому при обмене Session ID через GRASP принимающая сторона **должна** пометить идентификатор IP-адресом инициатора, чтобы предотвратить неоднозначность. В крайне маловероятном случае создание двумя партнёрами сессий с одним значением Session ID тег позволит различить эти сессии. Групповые сообщения GRASP и отклики на них, транслируемые между каналами, по этой причине включают глобальный (публичный) IP-адрес инициатора.

С крайне малой вероятностью два партнёра могут начать одновременные сеансы согласования с одним Session ID. В зависимости от реализации это может привести к путанице между сессиями. Меры предотвращения этого описаны в параграфе 2.8.6.

2.8. Сообщения GRASP

2.8.1. Обзор сообщений

В этом параграфе определён формат и типы сообщений GRASP. Не указанные здесь типы являются резервными. Определённые в настоящее время сообщения перечислены ниже.

Discovery и Discovery Response (M_DISCOVERY, M_RESPONSE).

Request Negotiation, Negotiation, Confirm Waiting, Negotiation End (M_REQ_NEG, M_NEGOTIATE, M_WAIT, M_END).

Request Synchronization, Synchronization, Flood Synchronization (M_REQ_SYN, M_SYNCH, M_FLOOD).

No Operation и Invalid (M_NOOP, M_INVALID).

2.8.2. Формат сообщений GRASP

Сообщения GRASP используют идентичные заголовки и область опций с меняющимся форматом. Заголовки и опции передаются в сжатом двоичном представлении (Concise Binary Object Representation или CBOR) [RFC8949]. В этой спецификации они описываются на языке краткого определения данных CDDL (Concise Data Definition Language) [RFC8610]. Для каждого элемента в этом описании применяется фрагментарный формат CDDL. Полная нормативная спецификация CDDL для протокола GRASP приведена в разделе 4 и включает константы, такие как типы сообщений.

Каждое сообщение GRASP (кроме No Operation) содержит Session ID (параграф 2.7). Затем размещается цепочка опций. Во фрагментарной форме CDDL каждое сообщение GRASP имеет показанный ниже вид.

```
grasp-message = (message .within message-structure) / noop-message

message-structure = [MESSAGE_TYPE, session-id, ?initiator, *grasp-option]

MESSAGE_TYPE = 0..255
session-id = 0..4294967295 ; до 32 битов
grasp-option = any
```

MESSAGE_TYPE указывает тип сообщения, который определяет ожидаемые опции. Любые полученные опции, которые не соответствуют MESSAGE_TYPE **следует** просто отбрасывать.

Сообщение No Operation (noop - нет операций) описано в параграфе 2.8.13. Значения MESSAGE_TYPE определены в разделе 4. Остальные элементы сообщений описаны ниже и формально определены в разделе 4.

При получении неизвестного MESSAGE_TYPE в индивидуальном (unicast) сообщении **можно** передавать в ответ сообщение Invalid (параграф 2.8.12). В ином случае сообщение **может** быть записано в системный журнал (log) и **должно** быть отброшено. Если неизвестный тип MESSAGE_TYPE получен в групповом сообщении это **может** быть записано в системный журнал, а сообщение **должно** быть отброшено.

2.8.3. Размер сообщения

Узлы GRASP **должны** поддерживать приём индивидуальных сообщений размером по меньшей мере GRASP_DEF_MAX_SIZE байтов. Узлам GRASP **недопустимо** передавать индивидуальные сообщения размером больше GRASP_DEF_MAX_SIZE, если больший размер явно не разрешён для соответствующей задачи. Например, можно использовать согласование GRASP для установки большего размера сообщений.

Используемому в GRASP синтаксическому анализатору сообщений следует знать GRASP_DEF_MAX_SIZE или согласованный больший размер, чтобы можно было защититься от чересчур больших сообщений.

Максимальный размер групповых сообщений M_DISCOVERY и M_FLOOD зависит от технологии канального уровня или используемого уровня адаптации канала.

2.8.4. Сообщение Discovery

Фрагментарная форма CDDL для сообщения Discovery имеет вид

```
discovery-message = [M_DISCOVERY, session-id, initiator, objective]
```

Инициатор обнаружения передаёт сообщение Discovery для запуска обнаружения конкретной задачи. Инициатор передаёт все сообщения Discovery по протоколу UDP через порт GRASP_LISTEN_PORT по групповому адресу link-local ALL_GRASP_NEIGHBORS на каждом интерфейсе канального уровня, используемом протоколом GRASP. Затем он прослушивает индивидуальные отклики TCP на данном порту и сохраняет результаты обнаружения, включая задачи и соответствующие индивидуальные адреса (локаторы).

Прослушиваемый порт TCP **должен** совпадать с портом UDP, используемым для групповой передачи Discovery на данном интерфейсе. В реализации с одним экземпляром GRASP на узле это **может** быть порт GRASP_LISTEN_PORT. Для поддержки на узле нескольких экземпляров механизму обнаружения в каждом экземпляре GRASP требуется найти на каждом интерфейсе динамический порт для передачи групповых пакетов (link-local) UDP и прослушивания откликов TCP до запуска обнаружения.

Поле initiator в сообщении содержит уникальный в глобальном масштабе IP-адрес инициатора с единственной целью - устранить неоднозначность Session ID (совпадение с другими узлами). Если у инициатора нет такого адреса IP, он **должен** использовать адрес link-local, который с высокой вероятностью будет уникальным для этой задачи (например, использовать [RFC7217]). Определение для узла глобально уникального адреса IP зависит от реализации.

Сообщение Discovery **должно** включать в точности 1 из указанных ниже опций.

- Опция Discovery (параграф 2.10.1). Для loop count **должно** быть установлено подходящее значение, чтобы предотвратить петли при обнаружении (по умолчанию GRASP_DEF_LOOPCT). Если инициатору обнаружения нужны отклики или от его канала, для loop count **должно** устанавливаться значение 1.
- Опция Negotiation (параграф 2.10.1) применяется для обнаружения и указания того, что задача для обнаружения **может** напрямую отвечать инициатору сообщением Negotiation для быстрой обработки, если она

может выступать партнёром соответствующего согласования. Отправитель такого сообщения Discovery **должен** запустить таймер согласования и установить для loop count, как это делается для сообщения Request Negotiation (параграф 2.8.6).

- Опция Synchronization (параграф 2.10.1) применяется для обнаружения и указания того, что задача для обнаружения **может** напрямую отвечать инициатору сообщением Synchronization для быстрой обработки, если она может выступать партнёром соответствующей синхронизации. Для loop count **должно** устанавливаться подходящее значение, чтобы предотвратить петли при обнаружении (по умолчанию GRASP_DEF_LOOPCT).

Как отмечено в параграфе 2.5.4.2, сообщения Discovery **можно** передавать по индивидуальному адресу партнёра, при этом его **следует** обрабатывать так же как при групповой передаче.

2.8.5. Сообщение Discovery Response

Фрагментарная форма CDDL для сообщения Discovery Response имеет вид

```
response-message = [M_RESPONSE, session-id, initiator, ttl,
                    (+locator-option // divert-option), ?objective]

ttl = 0..4294967295 ; в миллисекундах
```

Получившему сообщение Discovery узлу **следует** передавать ответное сообщение Discovery Response тогда и только тогда, когда он может ответить на запрос обнаружения.

Сообщение **должно** содержать Session ID и поле initiator из сообщения Discovery.

Сообщение **должно** содержать поле ttl, указывающее срок действия отклика целым положительным числом миллисекунд. для того, чтобы отклик был действительным. Значение 0 предполагает срок действия значительно больше GRASP_DEF_TIMEOUT (параграф 2.6). Рекомендуется значение в 10 раз больше GRASP_DEF_TIMEOUT.

Сообщение **может** включать копию задачи обнаружения из сообщения Discovery.

Сообщение передаётся отправителю Discovery по протоколу TCP через порт, использованный для Discovery, как описано в параграфе 2.8.4. При ответе на ретранслированное сообщение Discovery Response передаётся транслятору, а не исходному инициатору. Во всех случаях транспортную сессию **следует** закрывать после передачи Discovery Response. Отказ в транспортной сессии считается отсутствием отклика.

Если отвечающий узел поддерживает задачу для обнаружения, он **должен** включить хотя бы одну опцию локации любого типа (параграф 2.9.5) для указания своего местоположения. Допускается последовательность разнотипных опций локации (например, адрес IP и имя FQDN).

Если отвечающий узел сам не поддерживает задачу для обнаружения, но знает его локатор, **следует** ответить на сообщение Discovery опцией Divert (параграф 2.9.2), содержащей опцию локации или комбинацию таких опций, указывающую искомую задачу. Обработка Discovery Response описана более подробно в параграфе 2.5.4.

2.8.6. Запросы согласования и синхронизации

Фрагментарные формы CDDL для сообщений Request Negotiation и Request Synchronization имеют вид

```
request-negotiation-message = [M_REQ_NEG, session-id, objective]
request-synchronization-message = [M_REQ_SYN, session-id, objective]
```

Запрашивающий синхронизацию или согласование узел передаёт соответствующий запрос по индивидуальному адресу партнёра, используя подходящий протокол и номер порта (по результату обнаружения). Если при обнаружении была возвращено доменное имя FQDN, сначала выполняется распознавание (определение адреса).

Сообщение Request **должно** включать относящуюся к делу опцию задачи. В случае Request Negotiation эта опция **должна** содержать запрашиваемое значение.

При отправке Request Negotiation инициатор **должен** запустить таймер согласования для нового потока. По умолчанию для таймера установлено значение GRASP_DEF_TIMEOUT мсек. Если этот тайм-аут не был изменён сообщением Confirm Waiting (параграф 2.8.9), инициатор по завершении отсчёта таймера будет считать согласование неудачным.

При отправке Request Synchronization инициатору **следует** запустить таймер синхронизации (по умолчанию GRASP_DEF_TIMEOUT мсек). По завершении отсчёта таймера инициатор считает синхронизацию неудачной.

При отправке запроса инициатор **должен** установить для loop count в опции задачи значение, заданное спецификацией опции или (если оно не задано) установить GRASP_DEF_LOOPCT.

При получении узлом запроса для задачи, которую в данный момент не прослушивает ни один агент ASA, он **должен** сразу же закрыть соответствующий сокет для индикации инициатору. Это позволяет избежать ненужных тайм-аутов, если, например, ASA преждевременно завершил работу, но ядро GRASP прослушивает сокеты от его имени.

Для предотвращения крайне маловероятного конфликта, когда два узла одновременно запрашивают между собой сессии с одним Session ID (параграф 2.7), узел при получении запроса **должен** проверить, что полученное значение Session ID не используется локально. В случае конфликта узел **должен** отбросить запрос, а инициатор обнаружит это по тайм-ауту.

2.8.7. Сообщение Negotiation

Фрагментарный формат CDDL для сообщения Negotiation имеет вид

```
negotiation-message = [M_NEGOTIATE, session-id, objective]
```

Партнёр по согласованию передаёт Negotiation в ответ на Request Negotiation, Negotiation или Discovery в быстром режиме. Процесс согласования может выполняться в несколько этапов.

Сообщение Negotiation **должно** включать подходящую опцию Negotiation Objective с обновлением значения в соответствии с ходом согласования. Отправитель **должен** уменьшить значение loop count на 1. Если этот счётчик достигает 0, передавать сообщение **недопустимо**. В этом случае сеанс согласования завершается по тайм-ауту.

2.8.8. Сообщение Negotiation End

Фрагментарный формат CDDL для сообщения Negotiation End имеет вид

```
end-message = [M_END, session-id, accept-option / decline-option]
```

Партнёр по согласованию передаёт Negotiation End для завершения процедуры согласования. Сообщение **должно** включать опцию Accept (параграф 2.9.3) или Decline (параграф 2.9.4). Сообщение может передать любой из партнёров.

2.8.9. Сообщение Confirm Waiting

Фрагментарный формат CDDL для сообщения Confirm Waiting имеет вид

```
wait-message = [M_WAIT, session-id, waiting-time]
waiting-time = 0..4294967295 ; в миллисекундах
```

Отвечающий узел передаёт Confirm Waiting, чтобы попросить запросивший узел дождаться последующего ответа на согласование. Это может быть связано с тем, что локальному узлу требуется больше времени на согласование или это согласование зависит от другого выполняемого в данный момент. В сообщение **недопустимо** включать какие-либо опции. При получении указанное в сообщении время переопределяет время ожидания с перезапуском таймера текущего согласования (параграф 2.8.6).

Отвечающему узлу **следует** передавать Negotiation, Negotiation End или другое сообщение Confirm Waiting до завершения отсчёта таймера согласования. В противном случае при завершении отсчёта у инициатора тот **должен** будет счесть процедуру согласования неудачной.

2.8.10. Сообщение Synchronization

Фрагментарный формат CDDL для сообщения Synchronization имеет вид

```
synch-message = [M_SYNCH, session-id, objective]
```

Узел, принявший Request Synchronization или Discovery в быстром режиме, возвращает индивидуальное сообщение Synchronization с данными синхронизации в форме опции GRASP для конкретной задачи синхронизации из Request Synchronization.

2.8.11. Сообщение Flood Synchronization

Фрагментарный формат CDDL для сообщения Flood Synchronization имеет вид

```
flood-message = [M_FLOOD, session-id, initiator, ttl,
                 +[objective, (locator-option / [])]]

ttl = 0..4294967295 ; в миллисекундах
```

Узел **может** инициировать лавинную передачу, отправляя незапрошенное сообщение Flood Synchronization с данными синхронизации. Его **можно** передать в порт GRASP_LISTEN_PORT по групповому (link-local) адресу ALL_GRASP_NEIGHBORS в соответствии с правилами параграфа 2.5.6. Адрес инициатора указывается в соответствии с описанием для сообщения Discovery (параграф 2.8.4) лишь для устранения неоднозначности Session ID.

Сообщение **должно** содержать поле ttl для проверки действительности содержимого в форме положительного целого числа миллисекунд. Принятого по умолчанию значение нет, 0 означает неограниченный срок действия.

Данные синхронизации представляются в форме опций GRASP для конкретных задач синхронизации. В счётчиках интервалов (loop count) **должны** устанавливаться подходящие значения для предотвращения петель (по умолчанию GRASP_DEF_LOOPCT).

За каждой опцией задачи **может** следовать опция локации (параграф 2.9.5), связанная с этой задачей. При отсутствии локации **должна** включаться пустая опция.

Получивший сообщение Flood Synchronization узел **должен** кэшировать задачи из него для использования локальными агентами ASA. Каждая кэшированная задача **должна** помечаться опцией локации, откуда она была получена, или пустым тегом, если была передана пустая опция локации. Если последующее сообщение Flood Synchronization содержит задачу с тем же именем и тегом, соответствующая кэшированная копия задачи **должна** переписываться. Если последующее сообщение Flood Synchronization для задачи с тем же именем имеет другой тег, в кэше **должна** создаваться новая запись.

> Примечание. Цель этого механизма заключается в том, чтобы позволить получателю лавинных значений различать отправителей для одной задачи и при необходимости взаимодействия использовать локатор, протокол и порт из опции локации. Для многих задач это механизм не нужен и они будут рассылаться с пустой опцией локации.

Кэшированные записи **должны** игнорироваться или удаляться по истечении срока действия.

2.8.12. Сообщение Invalid

Фрагментарный формат CDDL для сообщения Invalid имеет вид

```
invalid-message = [M_INVALID, session-id, ?any]
```

Реализация **может** передавать это сообщение в ответ на принятое индивидуальное сообщение, которое сочтено недействительным. Значение Session ID **должно** копироваться из полученного сообщения. В содержимом сообщения **следует** указывать диагностические данные, такие как частичная копия недействительного сообщения, вплоть до максимального размера сообщения. Получатель M_INVALID **может** просто отбрасывать такие сообщения. Однако их можно применять для поддержки расширяемости, поскольку они показывают, что удалённый узел не поддерживает новое или устаревшее сообщение или опцию. Сообщение M_INVALID **недопустимо** передавать в ответ на M_INVALID.

2.8.13. Сообщение No Operation

Фрагментарный формат CDDL для сообщения No Operation имеет вид

```
noop-message = [M_NOOP]
```

Реализация **может** передавать это сообщение при практической необходимости инициализировать сокет. Получатель **должен** просто игнорировать сообщение.

2.9. Опции GRASP

В этом параграфе определены опции GRASP для сигнализации при согласовании и синхронизации. В будущем могут быть добавлены новые опции.

2.9.1. Формат опций GRASP

Опциям GRASP **следует** быть массивами CBOR, которые **должны** начинаться с целого числа без знака, указывающего тип опции. Формальные определения типов даны в разделе 4. Могут определяться опции GRASP, инкапсулирующие другие опции GRASP.

2.9.2. Опция Divert

Опция Divert служит для перенаправления запроса GRASP на другой узел, который более подходит для данного согласования или синхронизации. Это может быть перенаправление на объект, известный как партнёр для конкретного согласования или синхронизации (на канале или вне его), или принятый по умолчанию шлюз. Опция Divert **должна** инкапсулироваться лишь в сообщения Discovery Response. В других местах эту опцию **следует** просто игнорировать.

Инициатор обнаружения **может** игнорировать опцию Divert, если ему нужны лишь прямые сообщения Discovery Response.

Фрагментарная форма CDDL для опции Divert имеет вид

```
divert-option = [O_DIVERT, +locator-option]
```

Вложенные опции локации (параграф 2.9.5) указывают перенаправленные задачи в отклике на сообщение Discovery.

2.9.3. Опция Accept

Опция Accept применяется для указания партнёру по согласованию, что предложенные параметры приемлемы. Опция Accept **должна** инкапсулироваться лишь в сообщения Negotiation End, в других сообщениях её **следует** просто игнорировать.

Фрагментарная форма CDDL для опции Accept имеет вид

```
accept-option = [O_ACCEPT]
```

2.9.4. Опция Decline

Опция Decline применяется для указания партнёру по согласованию, что предложенные параметры приемлемы и согласование на этом завершается. Опция Decline **должна** инкапсулироваться лишь в сообщения Negotiation End, в других сообщениях её **следует** просто игнорировать.

Фрагментарная форма CDDL для опции Decline имеет вид

```
decline-option = [O_DECLINE, ?reason]
reason = text  ; необязательное сообщение об ошибке (UTF-8)
```

Примечание. Агент ASA может отказаться от предложенного значения и повторно запустить согласование. При этом реализация может передать опцию Decline или продолжить согласование сообщением Negotiation с опцией задачи, содержащей пустое значение или новое значение, которое может привести к сходимости согласования.

2.9.5. Опции локации

Опции локации служат для представления сведений о доступности для ASA, устройства или интерфейса. Опции включают Locator IPv6 Address, Locator IPv4 Address, Locator FQDN, Locator URI.

Поскольку агенты ASA обычно работают как независимые пользовательские программы, опции локации нужны для указания локатора сетевого уровня, а также транспортного протокола и номера порта для доступа к задаче. По этой причине опции локации для адресов IP и имён FQDN включают эти сведения явно. В опции Locator URI эта информация встроена в URI.

Примечание. Предполагается, что все локаторы в этих опциях входят в область действия домена GRASP. Как сказано в параграфе 2.2, протокол GRASP не предназначен для работы в несвязанных областях адресации и именования.

2.9.5.1. Опция Locator IPv6 Address

Фрагментарная форма CDDL для опции Locator IPv6 Address имеет вид

```
ipv6-locator-option = [O_IPv6_LOCATOR, ipv6-address, transport-proto, port-number]
ipv6-address = bytes .size 16

transport-proto = IPPROTO_TCP / IPPROTO_UDP
IPPROTO_TCP = 6
IPPROTO_UDP = 17
port-number = 0..65535
```

Опция содержит двоичное представление адреса IPv6, за которым указаны номера протокола и порта.

Примечание 1. Адрес IPv6 **должен** обычно иметь глобальную область действия. Однако во время инициализации для конкретных задач **может** применяться адрес link-local (параграф 2.5.2). В этом случае соответствующее сообщение Discovery Response **должно** передаваться через интерфейс, к которому относится адрес link-local.

Примечание 2. Адрес IPv6 link-local **недопустимо** применять в этой опции, когда она включена в опцию Divert.

Примечание 3. Значения IPPROTO берутся из существующего реестра IANA Protocol Numbers для протоколов TCP и UDP. Если для GRASP потребуются отсутствующие в реестре значения, потребуется новый реестр для значений вне диапазона 0 - 255.

2.9.5.2. Опция Locator IPv4 Address

Фрагментарная форма CDDL для опции Locator IPv4 Address имеет вид

```
ipv4-locator-option = [O_IPv4_LOCATOR, ipv4-address, transport-proto, port-number]
ipv4-address = bytes .size 4
```

Опция содержит двоичное представление адреса IPv4, за которым указаны номера протокола и порта.

Примечание. Если оператор применяет внутреннюю трансляцию адресов IPv4, эту опцию **недопустимо** включать в опцию Divert.

2.9.5.3. Опция Locator FQDN

Фрагментарная форма CDDL для опции Locator FQDN имеет вид

```
fqdn-locator-option = [O_FQDN_LOCATOR, text, transport-proto, port-number]
```

Опция содержит полное доменное имя FQDN, за которым указаны номера протокола и порта.

Примечание 1. Имена FQDN, которые могут быть недействительны в сети, такие как имена Multicast DNS [RFC6762], **недопустимо** применять для опции, помещаемой в опцию Divert.

Примечание 2. При обычной работе GRASP применение этой опции не предполагается. Она предназначена для специальных задач, таких как обнаружение внешних служб.

2.9.5.4. Опция Locator URI

Фрагментарная форма CDDL для опции Locator URI имеет вид

```
uri-locator-option = [O_URI_LOCATOR, text, transport-proto / null, port-number / null]
```

Опция содержит идентификатор URI для задачи, за которым указаны номера протокола и порта (или пустые значения, если номера не нужны) [RFC3986].

Примечание 1. Идентификаторы URI, которые могут быть недействительны в сети, такие как имена Multicast DNS [RFC6762], **недопустимо** применять для опции, помещаемой в опцию Divert.

Примечание 2. При обычной работе GRASP применение этой опции не предполагается. Она предназначена для специальных задач, таких как обнаружение внешних служб, поэтому не описана подробно в спецификации.

2.10. Опции задачи

2.10.1. Формат опций задачи

Опция objective служит для идентификации задачи в задачах для обнаружения, согласования или синхронизации. Все задачи **должны** иметь фрагментарный формат CDDL

```
objective = [objective-name, objective-flags, loop-count, ?objective-value]

objective-name = text
objective-value = any
loop-count = 0..255
```

Все задачи указываются уникальным именем в форме строки UTF-8 [RFC3629] с побайтовым сравнением. В имена базовых задач **недопустимо** включать двоеточие (:) и они **должны** регистрироваться в IANA (раздел 5). Имена заданных приватно задач **должны** включать хотя бы одно двоеточие. Строка, предшествующая последнему двоеточию в имени, **должна** быть уникальной в глобальном масштабе и тем или иным способом идентифицировать организацию или лицо, определившее задачу. Для создания уникальных в глобальном масштабе строк **можно** применять три указанных ниже варианта.

1. Строка, представляющая десятичное значение 32-битового идентификатора организации (Private Enterprise Number или PEN) [RFC5612], однозначно указывающего организацию, определившую задачу.
2. Строка имени FQDN, однозначно указывающая организацию или лицо, определившее задачу.
3. Строка адреса электронной почты, однозначно указывающая организацию или лицо, определившее задачу.

GRASP считает имя задачи необрабатываемой (opaque) строкой. Например, EX1, 32473:EX1, example.com:EX1, example.org:EX1, user@example.org:EX1 указывают 5 различных задач.

Поле objective-flags описано в параграфе 2.10.2.

Поле loop-count служит для прерывания согласования, как описано в параграфе 2.8.7, а также для прерывания обнаружения, как описано в параграфе 2.5.4, и лавинной отправки, как описано в параграфе 2.5.6.2. Оно помещается в objective, а не в сообщение GRASP, поскольку для ASA является свойством самой задачи.

Поле objective-value указывает действительное значение задачи для согласования или синхронизации. Его формат определяет спецификация задачи и это может быть простое значение или структура данных любого типа, если её можно представить в CBOR. Поле является необязательным в сообщениях Discovery и Discovery Response.

2.10.2. Поле objective-flags

Задача может относиться лишь к обнаружению, обнаружению и согласованию или обнаружению и синхронизации. Это указывается в задаче логическими полями флагов.

```
objective-flags = uint .bits objective-flag
objective-flag = &(
```

```
      F_DISC: 0    ; действителен для обнаружения
      F_NEG: 1     ; действителен для согласования
      F_SYNCH: 2   ; действителен для синхронизации
      F_NEG_DRY: 3 ; согласование является «пробным прогоном»
    )
```

Эти биты независимы и могут комбинироваться, например (F_DISC и F_SYNCH), (F_DISC и F_NEG), (F_DISC, F_NEG и F_NEG_DRY).

Отметим, что в данной сессии согласования задача должна использоваться для согласования или пробного прогона. Смешивание этих двух режимов не допускается.

2.10.3. Общее рассмотрение опций объекта

Как отмечено выше, опциям задач должны выделяться уникальные имена. Пока определённые частным образом опции следуют приведённым выше правилам, этот документ не ограничивает выбор имён, но заинтересованным сторонам **следует** публиковать используемые имена. Имена задаются строками UTF-8 для удобства разработки локализованных опций задач. Для общего применения **рекомендуется** применять для имён подмножество символов ASCII в кодировке UTF-8. Разработчикам, планирующим использовать имена с символами, отличными от ASCII, настоятельно рекомендуется ознакомиться с [RFC8264] или его преемником для понимания возникающих сложностей. Поскольку GRASP сравнивает имена побайтово, все проблемы профилирования и канонизации Unicode **должны** быть указаны в спецификации опции.

Все опции задач **должны** соответствовать шаблонам CBOR, определенным выше как objective, а также **должны** заменять все поля any действительными определениями данных CBOR для соответствующего применения. Опции задач, не содержащие полей кроме loop-count, могут применяться лишь для задач обнаружения и **должны** включаться лишь в сообщения Discovery и Discovery Response.

Опции Negotiation Objective содержат задачи для согласования, которые меняются в зависимости от функций и услуг. Они **должны** передаваться лишь в сообщениях Discovery, Request Negotiation или Negotiation. Инициатор согласования **должен** устанавливать исходное значение loop-count в соответствии со спецификацией задачи или GRASP_DEF_LOOPCT (если спецификация не задаёт значение). Для большинства случаев в запросах согласования следует указывать начальные значения. Поэтому опции Negotiation Objective **должны** всегда полностью включаться в сообщение Request Negotiation или сообщение Discovery в быстром режиме. При отсутствии начального значения в поле value **следует** устанавливать значение null, определённое в CBOR.

Опции Synchronization Objective похожи, но **должны** передаваться лишь в сообщениях Discovery, Discovery Response, Request Synchronization или Flood Synchronization. Они включают поле value только в сообщениях Synchronization и Flood Synchronization.

Задачи разными путями взаимодействуют с агентами ASA, которые используют их. Вопросы проектирования ASA рассмотрены в [ASA-GUIDELINES].

2.10.4. Организация опций задачи

Базовые опции задач **должны** быть заданы в общедоступных документах и их **следует** разрабатывать для применения в механизмах согласования или синхронизации, описанных выше. Как уже было отмечено, каждым потоком GRASP обрабатывается одна задача для согласования. Поэтому задаче для согласования, основанной на конкретной функции или действии, **следует** быть организованной в виде отдельной опции GRASP. **Не рекомендуется** объединять несколько задач для согласования в одну опцию, а также разделять одну функцию или или действие между несколькими опциями согласования.

Важно понимать, что согласование GRASP не поддерживает целостность транзакций. Если такая целостность нужна для конкретной задачи, она должна обеспечиваться агентом ASA. Например, ASA может потребоваться обеспечить участие в каждый момент лишь в одном потоке согласования. Такому ASA потребуется прекратить прослушивание входящих запросов на согласование перед созданием исходящего запроса согласования.

Задачи для синхронизации **следует** организовывать в одну опцию GRASP.

Некоторые задачи поддерживают не один режим работы. Примером может служить задача для согласования с режимом пробного запуска (dry-run), где согласование заключается в проверке возможности реализации запрошенных изменений на другой стороне, и реальным рабочим режимом (live), как описано в параграфе 2.5.5. Семантика таких режимов определяется в спецификации задачи. Таким задачам **следует** включать флаги, задающие режим.

Особо следует обращать внимание на то, что GRASP сам по себе не является протоколом защищённых транзакций. Любое состояние, связанное с пробным запуском, такое как временное резервирование ресурсов для последующего использования в рабочем режиме, полностью определяет разработчик соответствующего агента ASA.

Как отмечено в параграфе 2.1, значение задачи может включать множество параметров. Эти параметры можно разделить на два класса - обязательные, которые представляются фиксированными полями, и необязательные, представленные в какой-либо форме структуры данных, встроенной в CBOR. Формат может наследоваться от имеющихся протоколов управления и настройки, при этом опция задачи служит носителем формата. Структура данных может определяться формальным языком, но это задаёт спецификация задачи. В зависимости от контекста может применяться ABNF, RBNF, XML Schema, YANG и п., протокол GRASP может работать с разными формами. Единственным ограничением является необходимость отображения формата в CBOR.

Не рекомендуется смешивать в одной задаче параметры, сильно различающиеся по времени отклика. Для них лучше задавать отдельные задачи.

Все задачи **должны** поддерживать обнаружение GRASP. Однако, как отмечено в параграфе 2.3, агенты ASA могут применять иные методы обнаружения.

Обычно задача GRASP будет относиться к конкретным техническим параметрам, как разъяснено в параграфе 2.1. Однако можно определять абстрактные задачи для управления и координации агентов ASA. Можно также определять специальные задачи для таких целей, как доверенная начальная загрузка и формирование ACP.

Для гарантированного схождения требуется ограниченное число циклов обмена или время ожидания у каждой задачи для согласования. Поэтому в определении каждой задачи **следует** чётко указывать это, например, принятым по умолчанию значением loop count или тайм-аутом, чтобы согласование всегда можно было завершить корректно. В ином случае будут применяться принятые по умолчанию настройки GRASP.

Должна быть чётко определённая процедура принятия решения о невозможности согласования и определения в таких случаях последующих событий (например, устранение тупика, возврат к услугам best-effort). Это **должно** задаваться в спецификациях задач для согласования.

2.10.5. Опции для экспериментов и примеров

Имена EX0 - EX9 зарезервированы для экспериментальных опций. Выделение нескольких имён обусловлено тем, что в эксперименте может одновременно участвовать несколько опций. Весьма вероятно, что эти опции будут иметь разный смысл в различных приложениях, поэтому **не следует** их использовать без явного рассмотрения человеком и **недопустимо** применять эти опции в неуправляемых сетях, таких как домашние.

Эти имена **рекомендуются** также для примеров в документации.

3. Вопросы безопасности

Успех атаки на узел с поддержкой согласования может быть чрезвычайно опасным, поскольку может сделать конфигурацию узла совершенно нежелательной и оказать негативное влияние на партнёров узла. Поэтому для узлов и сообщений GRASP требуется полная защита. Как отмечено в параграфе 2.5.1, протокол GRASP **должен** работать в защищённой среде, такой как плоскость управления ACP [RFC8994], за исключением ограниченных экземпляров, описанных в параграфе 2.5.2.

Проверка подлинности

Для автоматических сетей (AN) требуется криптографически аутентифицированное отождествление (identity) каждого устройства. Небезопасно полагать, что большая сеть физически защищена от вредных воздействий или всему персоналу можно доверять. Каждый автоматический узел **должен** быть способен предоставить своё отождествление и подтвердить подлинность своих сообщений. Протокол GRASP полагается на отдельный внешний механизм защиты для поддержки аутентификации, защиты целостности данных и предотвращения повторного использования (replay).

Поскольку протокол GRASP должен разворачиваться в имеющейся защищённой среде, он сам не определяет ничего, относящегося к привязкам доверия и удостоверяющим центрам. Например, в ACP [RFC8994] все узлы могут доверять друг другу и агенты ASA устанавливаются на них. Если GRASP применяется временно без внешних механизмов защиты, например, при начальной загрузке системы (параграф 2.5.1), Session ID (параграф 2.7) будет служить одноразовым значением (nonce) для обеспечения ограниченной защиты от внедрения откликов посторонними. Полный анализ защиты процесса начальной загрузки приведён в [RFC8995].

Проверка полномочий и роли

GRASP не зависит от ролей и возможностей отдельных ASA и задач, которые уполномочены поддерживать конкретные ASA. Реализация может поддерживать меры предосторожности, позволяя лишь одному агенту ASA на данном узле менять данную задачу, но в некоторых случаях это может оказаться неуместным. Например, для работы может быть полезно разрешить одновременную работу старой и новой версии на одном ASA в течение переходного периода. Рассмотрение таких вопросов выходит за рамки этой спецификации.

Приватность и конфиденциальность

Протокол GRASP предназначен для управления сетью, включающего элементы сети, но не конечные хосты. Поэтому наличие персональных данных в сигнальном протоколе не предполагается и протокол не должен влиять на персональную приватность. Тем не менее, передающие персональные данные приложения нельзя исключать. Кроме того, могут согласовываться потоки трафика, VPN и т. п., что может представлять интерес для анализа трафика. Операторы обычно хотят скрыть от посторонних детали топологии сети и плотность трафика. Внутренние атаки нельзя исключить в большой сети, поэтому механизм защиты для протокола **должен** обеспечивать конфиденциальность сообщений. По этой причине параграф 2.5.1 требует использования ACP или иной защиты.

Защита группового трафика на локальном канале

У GRASP нет разумной замены групповой (link-local) сообщений Discovery и Flood Synchronization, которые передаются в открытом виде без аутентификации. Сообщения передаются лишь через интерфейсы в автоматическую сеть AN (см. параграфы 2.1 и 2.5.1), однако они доступны для перехвата на канале и могут быть подделаны подключёнными к каналу злоумышленниками. При обнаружении сообщения Discovery Response являются индивидуальными и защищены (параграф 2.5.1), поэтому не могут быть перехвачены посторонними. В случае лавинной синхронизации перехватчик на канале может получить рассылаемые лавинно задачи, но на такие сообщения ответы не передаются. Некоторые предосторожности для Flood Synchronization рассмотрены в параграфе 2.5.6.2.

Защита от DoS-атак

Обнаружение в GRASP частично полагается на незащищённые групповые (link-local) сообщения. Поскольку участвующие в GRASP маршрутизаторы иногда передают сообщения Discovery из одного канала в другой, это может быть использовано для организации DoS-атак. Некоторые меры снижения угроз описаны в параграфе 2.5.4. Однако вредоносный код, размещённый в ACP, всегда может организовать DoS-атаку, состоящую из ложных сообщений Discovery, либо из ложных откликов Discovery Response. Важно, чтобы межсетевые экраны предотвращали вход в домен сообщений GRASP от неизвестных источников.

Защита при начальной загрузке и обнаружении

Узел не может доверять трафику GRASP от других узлов, пока среда защиты (например, ACP) не задаст привязки доверия и не обеспечит аутентификацию трафика путём проверки сертификатов других узлов. Кроме того, пока узел не будет зарегистрирован [RFC8995], о не может считать, что другие узлы способны проверить подлинность его трафика. Поэтому обнаружение GRASP в фазе начальной загрузки нового устройства не будет защищённым. Защита согласования и синхронизации невозможна до завершения регистрации (см. параграф 2.5.2).

Защита обнаруженных локаторов

Возвращаемый GRASP адрес IP **должен** относиться к защищённой среде (параграф 2.5.1). Если возвращается FQDN или URI, получившему его агенту ASA **недопустимо** считать, что локатор находится в защищённой среде.

4. CDDL-спецификация GRASP

```
<CODE BEGINS> file "grasp.cddl"
grasp-message = (message .within message-structure) / noop-message

message-structure = [MESSAGE_TYPE, session-id, ?initiator, *grasp-option]

MESSAGE_TYPE = 0..255
session-id = 0..4294967295 ; до 32 битов
grasp-option = any

message /= discovery-message
discovery-message = [M_DISCOVERY, session-id, initiator, objective]

message /= response-message ; отклик на Discovery
response-message = [M_RESPONSE, session-id, initiator, ttl,
                    (+locator-option // divert-option), ?objective]

message /= synch-message ; отклик на запрос синхронизации
synch-message = [M_SYNCH, session-id, objective]

message /= flood-message
flood-message = [M_FLOOD, session-id, initiator, ttl,
                 +[objective, (locator-option / [])]]

message /= request-negotiation-message
request-negotiation-message = [M_REQ_NEG, session-id, objective]

message /= request-synchronization-message
request-synchronization-message = [M_REQ_SYN, session-id, objective]

message /= negotiation-message
negotiation-message = [M_NEGOTIATE, session-id, objective]

message /= end-message
end-message = [M_END, session-id, accept-option / decline-option]

message /= wait-message
wait-message = [M_WAIT, session-id, waiting-time]

message /= invalid-message
invalid-message = [M_INVALID, session-id, ?any]

noop-message = [M_NOOP]

divert-option = [O_DIVERT, +locator-option]

accept-option = [O_ACCEPT]

decline-option = [O_DECLINE, ?reason]
reason = text  ; необязательное сообщение об ошибке (UTF-8)

waiting-time = 0..4294967295 ; в миллисекундах
ttl = 0..4294967295 ; в миллисекундах

locator-option /= [O_IPv4_LOCATOR, ipv4-address, transport-proto, port-number]
ipv4-address = bytes .size 4

locator-option /= [O_IPv6_LOCATOR, ipv6-address,
                   transport-proto, port-number]
ipv6-address = bytes .size 16

locator-option /= [O_FQDN_LOCATOR, text, transport-proto,
                   port-number]

locator-option /= [O_URI_LOCATOR, text,
                   transport-proto / null, port-number / null]

transport-proto = IPPROTO_TCP / IPPROTO_UDP
IPPROTO_TCP = 6
IPPROTO_UDP = 17
port-number = 0..65535

initiator = ipv4-address / ipv6-address

objective-flags = uint .bits objective-flag

objective-flag = &(
  F_DISC: 0    ; действительно для обнаружения
  F_NEG: 1     ; действительно для согласования
  F_SYNCH: 2   ; действительно для синхронизации
  F_NEG_DRY: 3 ; согласование является пробным прогоном
)

objective = [objective-name, objective-flags, loop-count, ?objective-value]
```

```
   objective-name = text ; см. параграф "Формат опций задачи"

   objective-value = any

   loop-count = 0..255

   ; Константы для типов сообщений и опций
   M_NOOP = 0
   M_DISCOVERY = 1
   M_RESPONSE = 2
   M_REQ_NEG = 3
   M_REQ_SYN = 4
   M_NEGOTIATE = 5
   M_END = 6
   M_WAIT = 7
   M_SYNCH = 8
   M_FLOOD = 9
   M_INVALID = 99

   O_DIVERT = 100
   O_ACCEPT = 101
   O_DECLINE = 102
   O_IPv6_LOCATOR = 103
   O_IPv4_LOCATOR = 104
   O_FQDN_LOCATOR = 105
   O_URI_LOCATOR = 106
   <CODE ENDS>
```

5. Взаимодействие с IANA

Документ определяет базовый протокол автоматической сигнализации GRASP (GeneRic Autonomic Signaling Protocol).

В параграфе 2.6 указаны приведённые ниже групповые адреса link-local, выделенные IANA для GRASP.

Субреестр Link-Local Scope Multicast Addresses в реестре IPv6 Multicast Address Space

```
   Address(es):  ff02::13
   Description:  ALL_GRASP_NEIGHBORS
   Reference:  RFC 8990
```

Субреестр Local Network Control Block (224.0.0.0 - 224.0.0.255 (224.0.0/24)) в реестре IPv4 Multicast Address Space

```
   Address(es):  224.0.0.119
   Description:  ALL_GRASP_NEIGHBORS
   Reference:  RFC 8990
```

В параграфе 2.6 указан пользовательский порт GRASP_LISTEN_PORT, выделенный IANA для использования GRASP с протоколами UDP и TCP

```
   Service Name:  grasp
   Port Number:  7017
   Transport Protocol:  udp, tcp
   Description  GeneRic Autonomic Signaling Protocol
   Assignee:  IESG <iesg@ietf.org>
   Contact:  IETF Chair <chair@ietf.org>
   Reference:  RFC 8990
```

Агентство IANA создало реестр GeneRic Autonomic Signaling Protocol (GRASP) Parameters с двумя субреестрами GRASP Messages and Options и GRASP Objective Names. Субреестр GRASP Messages and Options содержит пары, состоящие из имени и десятичного целого числа. Новые значений **должны** выделяться по процедуре Standards Action, определённой в [RFC8126]. Назначенные этим документом значение приведены в таблице 1.

Таблица 1. Значения субреестра GRASP Messages and Options.

Значение	***Сообщение или опция***
0	M_NOOP
1	M_DISCOVERY
2	M_RESPONSE
3	M_REQ_NEG
4	M_REQ_SYN
5	M_NEGOTIATE
6	M_END
7	M_WAIT
8	M_SYNCH
9	M_FLOOD
99	M_INVALID
100	O_DIVERT
101	O_ACCEPT
102	O_DECLINE
103	O_IPv6_LOCATOR
104	O_IPv4_LOCATOR
105	O_FQDN_LOCATOR
106	O_URI_LOCATOR

Субреестр GRASP Objective Names содержит строки UTF-8, в которых **недопустимо** включение двоеточий (:), согласно параграфу 2.10.1. Новые значения **должны** выделяться по процедуре Specification Required, заданной в [RFC8126].

Для облегчения экспертной оценки новых задач в их спецификации следует включать точное описание формата задачи и достаточными для независимой реализации разъяснениями семантики (см. параграф 2.10.3). Если новая

задача похожа именем или назначением на ранее зарегистрированную, в спецификации следует обосновать необходимость этого.

Начальные значения, выделенные в этом документе, приведены в таблице 2.

Таблица 2. Исходные значения субреестра GRASP Objective Names.

Имя	***Документ***
EX1	RFC 8990
EX2	RFC 8990
EX3	RFC 8990
EX4	RFC 8990
EX5	RFC 8990
EX6	RFC 8990
EX7	RFC 8990
EX8	RFC 8990
EX9	RFC 8990

6. Литература

6.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/info/rfc8994>.

6.2. Дополнительная литература

[ADNCP] Stenberg, M., "Autonomic Distributed Node Consensus Protocol", Work in Progress, Internet-Draft, draft-stenberg-anima-adncp-00, 5 March 2015, <https://tools.ietf.org/html/draft-stenberg-anima-adncp-00>.

[ASA-GUIDELINES] Carpenter, B., Ciavaglia, L., Jiang, S., and P. Peloso, "Guidelines for Autonomic Service Agents", Work in Progress, Internet-Draft, draft-ietf-anima-asa-guidelines-00, 14 November 2020, <https://tools.ietf.org/html/draft-ietf-anima-asa-guidelines-00>.

[IGCP] Behringer, M. H., Chaparadza, R., Xin, L., Mahkonen, H., and R. Petre, "IP based Generic Control Protocol (IGCP)", Work in Progress, Internet-Draft, draft-chaparadza-intarea-igcp-00, 25 July 2011, <https://tools.ietf.org/html/draft-chaparadza-intarea-igcp-00>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2334] Luciani, J., Armitage, G., Halpern, J., and N. Doraswamy, "Server Cache Synchronization Protocol (SCSP)", RFC 2334, DOI 10.17487/RFC2334, April 1998, <https://www.rfc-editor.org/info/rfc2334>.

[RFC2608] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, DOI 10.17487/RFC2608, June 1999, <https://www.rfc-editor.org/info/rfc2608>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC3416] Presuhn, R., Ed., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, DOI 10.17487/RFC3416, December 2002, <https://www.rfc-editor.org/info/rfc3416>.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, DOI 10.17487/RFC3493, February 2003, <https://www.rfc-editor.org/info/rfc3493>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC5612] Eronen, P. and D. Harrington, "Enterprise Number for Documentation Use", RFC 5612, DOI 10.17487/RFC5612, August 2009, <https://www.rfc-editor.org/info/rfc5612>.

[RFC5971] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", RFC 5971, DOI 10.17487/RFC5971, October 2010, <https://www.rfc-editor.org/info/rfc5971>.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <https://www.rfc-editor.org/info/rfc6206>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <https://www.rfc-editor.org/info/rfc6733>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <https://www.rfc-editor.org/info/rfc6887>.

[RFC7558] Lynn, K., Cheshire, S., Blanchet, M., and D. Migault, "Requirements for Scalable DNS-Based Service Discovery (DNS-SD) / Multicast DNS (mDNS) Extensions", RFC 7558, DOI 10.17487/RFC7558, July 2015, <https://www.rfc-editor.org/info/rfc7558>.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <https://www.rfc-editor.org/info/rfc7575>.

[RFC7576] Jiang, S., Carpenter, B., and M. Behringer, "General Gap Analysis for Autonomic Networking", RFC 7576, DOI 10.17487/RFC7576, June 2015, <https://www.rfc-editor.org/info/rfc7576>.

[RFC7787] Stenberg, M. and S. Barth, "Distributed Node Consensus Protocol", RFC 7787, DOI 10.17487/RFC7787, April 2016, <https://www.rfc-editor.org/info/rfc7787>.

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <https://www.rfc-editor.org/info/rfc7788>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8264] Saint-Andre, P. and M. Blanchet, "PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols", RFC 8264, DOI 10.17487/RFC8264, October 2017, <https://www.rfc-editor.org/info/rfc8264>.

[RFC8368] Eckert, T., Ed. and M. Behringer, "Using an Autonomic Control Plane for Stable Connectivity of Network Operations, Administration, and Maintenance (OAM)", RFC 8368, DOI 10.17487/RFC8368, May 2018, <https://www.rfc-editor.org/info/rfc8368>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8991] Carpenter, B., Liu, B., Ed., Wang, W., and X. Gong, "GeneRic Autonomic Signaling Protocol Application Program Interface (GRASP API)", RFC 8991, DOI 10.17487/RFC8991, May 2021, <https://www.rfc-editor.org/info/rfc8991>.

[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <https://www.rfc-editor.org/info/rfc8993>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/info/rfc8995>.

Приложение A. Примеры формата сообщений

Для читателей, не знакомых с CBOR, в этом приложении даны примеры сообщений GRASP в соответствии с синтаксисом CDDL, заданным в разделе 4. Каждое сообщение приведено трижды в разных форматах:

1. диагностическая нотация CBOR;
2. похожая нотация с указанием имён константами (детали представления флагов опущены);
3. шестнадцатеричное представление формата CBOR в линии передачи.

Длинные строки разделены на несколько строк.

A.1. Discovery

Инициатор (2001:db8:f000:baaa:28cc:dc4c:9703:6781) передаёт групповое сообщение Discovery для поиска задачи EX1

```
[1, 13948744, h'20010db8f000baaa28ccdc4c97036781', ["EX1", 5, 2, 0]]
[M_DISCOVERY, 13948744, h'20010db8f000baaa28ccdc4c97036781',
              ["EX1", F_SYNCH_bits, 2, 0]]
h'84011a00d4d7485020010db8f000baaa28ccdc4c9703678184634558310502 00'
```

Партнёр (2001:0db8:f000:baaa:f000:baaa:f000:baaa) отвечает на сообщение, указывая локатор

```
[2, 13948744, h'20010db8f000baaa28ccdc4c97036781', 60000,
              [103, h'20010db8f000baaaf000baaaf000baaa', 6, 49443]]
[M_RESPONSE, 13948744, h'20010db8f000baaa28ccdc4c97036781', 60000,
              [O_IPv6_LOCATOR, h'20010db8f000baaaf000baaaf000baaa',
               IPPROTO_TCP, 49443]]
h'85021a00d4d7485020010db8f000baaa28ccdc4c9703678119ea6084186750
  20010db8f000baaaf000baaaf000baaa0619c123'
```

A.2. Лавинная синхронизация

Инициатор передаёт групповое сообщение Flood Synchronization. Одна задача имеет пустой локатор. Отклика нет.

```
[9, 3504974, h'20010db8f000baaa28ccdc4c97036781', 10000,
             [["EX1", 5, 2, ["Example 1 value=", 100]],[] ] ]
[M_FLOOD, 3504974, h'20010db8f000baaa28ccdc4c97036781', 10000,
             [["EX1", F_SYNCH_bits, 2, ["Example 1 value=", 100]],[] ] ]
h'85091a00357b4e5020010db8f000baaa28ccdc4c97036781192710
  828463455831050282704578616d706c6520312076616c75653d186480'
```

A.3. Синхронизация

После обнаружения задачи EX2 инициатор передаёт сообщение Request Synchronization

```
[4, 4038926, ["EX2", 5, 5, 0]]
[M_REQ_SYN, 4038926, ["EX2", F_SYNCH_bits, 5, 0]]
h'83041a003da10e8463455832050500'
```

Партнёр возвращает значение

```
[8, 4038926, ["EX2", 5, 5, ["Example 2 value=", 200]]]
[M_SYNCH, 4038926, ["EX2", F_SYNCH_bits, 5, ["Example 2 value=", 200]]]
h'83081a003da10e8463455832050582704578616d706c6520322076616c75653d18c8'
```

A.4. Пример простого согласования

После обнаружения задачи EX3 инициатор передаёт сообщение Request Negotiation

```
[3, 802813, ["EX3", 3, 6, ["NZD", 47]]]
[M_REQ_NEG, 802813, ["EX3", F_NEG_bits, 6, ["NZD", 47]]]
h'83031a000c3ffd8463455833030682634e5a44182f'
```

Партнёр сразу же принимает предложенное. Отметим, что задачу можно не указывать, поскольку запрос воспринят.

```
[6, 802813, [101]]
[M_END , 802813, [O_ACCEPT]]
h'83061a000c3ffd811865'
```

A.5. Пример полного согласования

Инициатор передаёт индивидуальное сообщение Request Negotiation

```
[3, 13767778, ["EX3", 3, 6, ["NZD", 410]]]
[M_REQ_NEG, 13767778, ["EX3", F_NEG_bits, 6, ["NZD", 410]]]
h'83031a00d21462846345583303068 2634e5a4419019a'
```

Ответчик начинает согласование (внося предложение)

```
[5, 13767778, ["EX3", 3, 6, ["NZD", 80]]]
[M_NEGOTIATE, 13767778, ["EX3", F_NEG_bits, 6, ["NZD", 80]]]
h'83051a00d21462846345583303068 2634e5a441850'
```

Инициатор продолжает согласование (понижая запрос и уменьшая значение loop count)

```
[5, 13767778, ["EX3", 3, 5, ["NZD", 307]]]
[M_NEGOTIATE, 13767778, ["EX3", F_NEG_bits, 5, ["NZD", 307]]]
h'83051a00d21462846345583303058 2634e5a44190133'
```

Ответчик просит дополнительное время

```
[7, 13767778, 34965]
[M_WAIT, 13767778, 34965]
h'83071a00d21462198895'
```

Ответчик продолжает согласование (снижая своё предложение)

```
[5, 13767778, ["EX3", 3, 4, ["NZD", 120]]]
[M_NEGOTIATE, 13767778, ["EX3", F_NEG_bits, 4, ["NZD", 120]]]
h'83051a00d21462846345583303048 2634e5a441878'
```

Инициатор продолжает согласование (понижая запрос)

```
[5, 13767778, ["EX3", 3, 3, ["NZD", 246]]]
[M_NEGOTIATE, 13767778, ["EX3", F_NEG_bits, 3, ["NZD", 246]]]
h'83051a00d21462846345583303038 2634e5a4418f6'
```

Ответчик отвергает согласование

```
[6, 13767778, [102, "Insufficient funds"]]
[M_END , 13767778, [O_DECLINE, "Insufficient funds"]]
h'83061a00d2146282186672496e7375666669636 9656e742066756e6473'
```

Согласование завершается отказом. Если любая из сторон передаст [M_END, 13767778, [O_ACCEPT]], согласование будет успешным, сходясь к значению из предыдущего M_NEGOTIATE. За исключением исходного M_REQ_NEG процесс является симметричным.

Приложение B. Требования к Discovery, Synchronization, Negotiation

В этом приложении рассматриваются требования к возможностям обнаружения, согласования и синхронизации. Основными пользователями протокола являются автоматические агенты служб (ASA), поэтому требования в основном выражаются функциями, необходимыми для ASA. Одно физическое устройство может включать несколько ASA, а один агент ASA может управлять несколькими техническими задачами. Если техническая задача управляется несколькими ASA, их нужно координировать вне протокола GRASP. Более того, требования к самим ASA, такие как обработка Intent [RFC7575], выходят за рамки этого документа.

B.1. Требования к обнаружению

D1

ASA могут быть разработаны для управления любым настраиваемым устройством или программой, как указано в приложении B.2. Базовое требование заключается в том, чтобы протокол мог представлять и обнаруживать технические задачи любого сорта (как определено в параграфе 2.1) среди произвольного набора узлов.

В сети AN приходится считать, что при первом запуске устройства оно не имеет сведений о партнёрах, структуре сети и своей роли в сети. Агент(ы) ASA внутри устройства находятся в такой же ситуации. В некоторых случаях при запуске в устройстве новой прикладной сессии устройству или ASA также может не хватать информации о соответствующих партнёрах. Например, может потребоваться настройка ресурсов на множестве других устройств, скоординированных и согласованных между собой, чтобы не тратить лишних ресурсов. Может потребоваться обновление параметров защиты для нового устройства или пользователя. Относящиеся к делу партнёры могут различаться для разных технических задач. Поэтому обнаружение приходится повторять, чтобы найти партнёров, подходящих для каждой задачи, которую инициатору обнаружения нужно обслуживать. Из этих соображений выводятся следующие 3 требования.

D2

При первом запуске ASA у агента может не быть сведений о конкретной сети, к которой агент подключён. Поэтому процесс обнаружения должен быть способен поддерживать любой сетевой сценарий, предполагая лишь, что устройство загружается с заводскими настройками.

D3

При запуске агента ASA он не должен требовать настройки сведений о местоположении партнёров для их обнаружения.

D4

Если ASA поддерживает несколько технических задач, соответствующие партнёры при поиске могут быть разными для различных задач. Таким образом, требуется механизм, с помощью которого агент ASA может раздельно обнаруживать партнерские ASA для каждой из технических задач, которыми нужно управлять.

D5

После обнаружения ASA обычно выполняет согласование или синхронизацию для соответствующих задач. Это следует разрешать, удобно связывая обнаружение с согласованием и синхронизацией. Можно предоставлять дополнительный механизм, позволяющий объединить обнаружение с согласованием или синхронизацией в одном протокольном обмене.

D6

Некоторые задачи могут быть значимыми лишь на локальном канале, о другие сохраняют значимость в маршрутизируемой сети и требуют операций вне канала (off-link). Таким образом, партнёры могут быть прямыми соседями по каналу L2 или более удалёнными, доступными лишь через уровень L3. Поэтому нужны механизмы обнаружения агентов ASA, поддерживающих определённые технические задачи, на локальном канале и за его пределами.

D7

Процессу обнаружения следует быть достаточно гибким для особых случаев, таких как указанные ниже.

- В процессе инициализации устройство должно быть способно организовать взаимное доверие с автоматическими узлами в сети и участвовать в механизме проверки подлинности. Хотя это неизбежно начинается с обнаружения, случай является особым, поскольку доверие ещё не организовано. Этот вопрос рассматривается в [RFC8995]. Требуется, чтобы после организации доверия с устройством все агенты ASA в нем наследовали свидетельства устройства и также становились доверенными. Это не препятствует наличию у устройства нескольких свидетельств (credential).
- В зависимости от типа сети может потребоваться обнаружение других централизованных функций, таких как операционный центр (Network Operations Center или NOC) [RFC8368]. Протокол должен быть способен поддерживать такое обнаружение в процессе инициализации, как и обнаружение в процессе работы.

D8

Процессу обнаружение недопустимо создавать избыточный трафик и он должен учитывать спящие узлы.

D9

Требуется механизм обслуживания устаревших результатов обнаружения.

B.2. Требования к синхронизации и согласованию

Сети AN должны быть способны управлять разнотипными параметрами и учитывать множество измерений, таких как задержки, нагрузка, неиспользуемые или ограниченные ресурсы, конфликты запроса ресурсов, настройки защиты, экономия энергии, балансировка нагрузки и т. п. Информацию о состоянии и метрику ресурсов узлам требуется использовать совместно для динамической настройки и мониторинга ресурсов. Хотя это может быть достигнуты с помощью имеющихся протоколов (если они есть), новый протокол должен быть способен поддерживать обмен параметрами, включая взаимную синхронизацию, даже если согласование, как таковое, не требуется. В общем случае эти параметры применяются не ко всем участвующим узлам, а лишь к их части.

SN1

Базовым требованием к протоколу является способность представлять, обнаруживать, синхронизировать и согласовывать почти любой тип сетевых параметров между выбранными подмножествами участвующих узлов.

SN2

Согласование выполняется с помощью итераций запросов и откликов, которые гарантированно должны завершаться (успех или отказ). Хотя для каждого случая должны быть определены правила разрыва (tie-breaking), протоколу следует иметь некие общие механизмы для предотвращения петель или «зависания», такие как ограничение числа пересылок или тайм-ауты.

SN3

Синхронизация должна быть возможна для малых и больших групп устройств.

SN4

Чтобы не «изобретать велосипед», протоколу следует поддерживать инкапсуляцию форматов данных, используемых имеющимися протоколами настройки (такими как NETCONF и YANG), когда это удобно.

SN5

Вмешательство человека в сложных ситуациях дорого и чревато ошибками, поэтому желательна синхронизация и согласование параметров множества устройств без участия человека, когда координация устройств может повысить общую производительность сети. Отсюда следует, что требования протокола к ресурсам должны быть достаточно скромными, чтобы протокол можно было реализовать на любом устройстве, которое без этого требовало бы привлечения человека. Работа протокола на узлах с ограниченными возможностями рассмотрена в [RFC8993].

SN6

Вмешательство людей в больших сетях часто заменяется использованием вертикальных (top-down) систем сетевого управления (network management system или NMS). Поэтому протоколу, как части инфраструктуры сетей AN, следует поддерживать возможность работы на любом устройстве, которому иначе потребовалось бы привлечение NMS, а также сосуществования с NMS и такими протоколами, как SNMP и NETCONF.

SN7

Предполагается, что автоматические функции будут реализованы в отдельных ASA, но протокол должен быть достаточно общим, чтобы позволять это. Некоторые примеры представлены ниже.

- Зависимости и конфликты. При выборе конфигурации для данного устройства может потребоваться информация от соседей. Это можно реализовать через согласование или синхронизацию, если этого достаточно. Однако элемент в соседнем узле может зависеть от информации от его соседей, для получения которой может потребоваться другая синхронизация или согласование. В результате могут возникать зависимости и конфликты между процедурами согласования и синхронизации. Разрешение таких зависимостей и конфликтов является делом вовлечённого в них ASA. Для этого нужны чёткие границы и механизмы схождения при согласовании, а также те или иные механизмы предотвращения циклических зависимостей и неконтролируемого роста дерева зависимостей. Разработчики ASA должны обеспечить такие механизмы. Роль протокола ограничивается двухсторонней сигнализацией между агентами ASA и предотвращением петель в этой сигнализации.
- Восстановление после отказов и обнаружение неисправных устройств следует обеспечивать автоматически, насколько это возможно. Роль протокола ограничивается обнаружением, синхронизацией и согласованием. Эти процессы могут выполняться в любое время и ASA может потребоваться повторение любого из них при обнаружении таких событий, как отказ партнёра по согласованию.
- Поскольку основной целью является минимизация участия человека, нужно, чтобы сеть могла «думать наперёд» перед изменением своих параметров. Одним из аспектов этого является использование агентами ASA базы знаний для предсказания поведения сети, выходящее за рамки сигнального протокола. Другим аспектом является прогнозирование эффекта изменений путём пробного прогона (dry run) согласования до фактического внесения изменений. Поэтому сигнализация пробного прогона является желательным свойством протокола.

Отметим, что требуется поддерживать журнал управления (log), мониторинг, предупреждения и инструменты для вмешательства. Однако это связано скорее со свойствами отдельных ASA, а не самого протокола. В документе [RFC8368] рассматривается связывание агентов с традиционными системами OAM (Operations, Administration, and Maintenance) через автоматическую плоскость управления ACP [RFC8994].

SN8

Протокол сможет работать с широким спектром технических задач, охватывая все типы сетевых параметров. Поэтому протоколу нужен гибкий и легко расширяемый формате для описания задач. На более позднем этапе может оказаться желательным принятие явной информационной модели. Один из вопросов заключается в приспособлении имеющейся модели или разработке новой.

B.3. Конкретные технические требования

T1

Следует обеспечить разработчикам ASA удобство определения новых технических задач, а программистам - их выражения без чрезмерного влияния на эффективность исполнения и размер кода. В частности, должно быть удобно реализовать агенты ASA независимо один от другого как программы пользовательского пространства, а не код ядра, если возможна такая модель программирования. Классы устройств, на которых может работать протокол, рассматриваются в[RFC8993].

T2

Протоколу следует быть просто расширяемым в случаях, когда изначально заданных механизмов обнаружения, синхронизации и согласования окажется недостаточно.

T3

Для универсальности платформы формату данных (payload) протокола следует быть независимым от транспортного протокола и версии IP. В частности, следует поддерживать работу по протоколу IPv6 и IPv4. Однако некоторые функции, такие как групповая адресация на канале, могут зависеть от версии IP. По умолчанию предпочтительней IPv6.

T4

Протокол должен обеспечивать возможность доступа к партнёрам вне канала (off-link) по маршрутизируемым адресам, т. е. не должен ограничиваться операциями на локальном канале (link-local).

T5

Должна обеспечиваться возможность применения внешнего механизма обнаружения, если он подходит для данной технической задачи. Иными словами, обнаружение GRASP не должно быть предварительным условием для согласования и синхронизации GRASP.

T6

Протокол должен быть способен различать одновременные операции с одним или несколькими партнёрами, особенно при возникновении ожидания.

T7

Распространение Intent выходит за рамки этого документа, однако протокол не должен исключать его использования для распространения Intent.

T8

Устройствам следует предоставлять возможность передачи отчётов системе управления. Для некоторых событий должна быть возможность генерации сигналов оператору, а также должна быть предусмотрена возможность экстренного вмешательства (например, для приостановки синхронизации или согласования на некорректно работающем устройстве). Эти функции могут не использовать сам протокол сигнализации, но устройство протокола не должно исключать такое использование.

T9

Поскольку протокол может напрямую менять конфигурацию устройств и оказывать существенное влияние на работу сети, все протокольные обмены должны быть полностью защищены от подставных сообщений и MITM-атак, а также максимально защищены от DoS-атак. Требуется также механизм шифрования для защиты от нежелательного мониторинга. Однако наличие функций защиты в самом протоколе не требуется и можно использовать имеющуюся защищённую среду.

Приложение C. Анализ возможностей современных протоколов

В этом приложении рассматриваются имеющиеся протоколы, связанные с требованиями Приложения B, с целью оценки возможности их выполнения отельным протоколом или их комбинацией.

Многие протоколу включают ту или иную форму обнаружения, но все они представляются предназначенными лишь для определённого применения. Протокол обнаружения служб (Service Location Protocol или SLP) [RFC2608] обеспечивает обнаружение сервиса в управляемых сетях, но требует настройки своих серверов. Обнаружение служб через DNS (DNS-Based Service Discovery или DNS-SD) [RFC6763] в комбинации с Multicast DNS (mDNS) [RFC6762] обеспечивает обнаружение служб для небольших сетей с одним канальным уровнем. В [RFC7558] предпринята попытка расширить это для более крупных автоматических сетей, но решение ещё не стандартизовано. Однако SLP и DNS-SD представляются ориентированными в первую очередь на службы уровня приложений, а не на задачи L2 и L3, относящиеся к базовой настройке сети. Протоколы SLP и DNS-SD основаны на передаче текста.

Простой протокол сетевого управления (Simple Network Management Protocol или SNMP) [RFC3416] использует модель «запрос-отклик», не подходящую для согласования между партнёрами. В NETCONF [RFC6241] применяется модель RPC, позволяющая позитивные и негативные отклики от целевой системы, но этого недостаточно для согласования.

Имеется много протоколов с элементарными возможностями согласования, например, DHCPv6 (Dynamic Host Configuration Protocol for IPv6) [RFC8415], ND (Neighbor Discovery) [RFC4861], PCP (Port Control Protocol) [RFC6887], RADIUS (Remote Authentication Dial-In User Service) [RFC2865], Diameter [RFC6733] и т. п. Большинство из них относится к протоколам настройки или управления, однако они предоставляют лишь простую модель «запрос-отклик» в контексте «ведущий-ведомый» или очень ограниченные возможности согласования.

Есть несколько протоколов сигнализации с элементами согласования. Например, протокол резервирования ресурсов (Resource ReSerVation Protocol или RSVP) [RFC2205] был разработан для согласования параметров качества обслуживания на пути групповых или индивидуальных потоков. Протокол RSVP специализирован для сквозных потоков. Более общее решение предлагает протокол GIST (General Internet Signalling Transport) [RFC5971], однако он пытается решить слишком много проблем, что делает протокол сложным, а также нацелен на передачу сигналов для потока через большое число узлов пересылки (hop), а не на сигнализацию между парой устройств. Тем не менее, не следует полностью исключать возможность расширения RSVP или GIST в качестве протокола синхронизации и согласования. Эти протоколы не представляются подходящими напрямую для обнаружения партнёров.

Протокол RESTCONF [RFC8040] предназначен для передачи сведений NETCONF, представленных на языке YANG, по протоколу HTTP, включая прохождение через узлы-посредники HTML. Хотя протокол обеспечивает эффективное решение в контексте централизованной настройки сложных сетей, он не приспособлен для интерактивного согласования между устройствами-партнерами, особенно простыми, которые могут не обрабатывать YANG.

Протокол DNCP (Distributed Node Consensus Protocol) [RFC7787] определён в качестве базового протокола синхронизации состояний с предложенным профилем использования для управления домашней сетью (Home Networking Control Protocol или HNCP) [RFC7788] при настройке маршрутизаторов Homenet. Применение DNCP для автоматических сетей AN предложено в [ADNCP]. Согласно [RFC7787]:

- протокол DNCP разработан для обеспечения каждому участвующему узлу возможности публиковать набор TLV (Type-Length-Value) размером до 64 Кбайт и обеспечивать общее представление публикуемых данных;
- DNCP подходит в основном для данных, которые меняются нечасто;
- при необходимости часто и быстро менять состояния предпочтительно использовать дополнительный канал «точка-точка».

Протокол DNCP имеет ряд специфических свойств,

- Каждый участвующий узел имеет уникальный идентификатор.
- Сообщения DNCP представляются последовательностью объектов TLV и передаются в индивидуальных пакетах UDP или TCP с необязательной защитой (D)TLS.
- Групповая адресация применяется только для обнаружения соседей DNCP, когда можно снизить уровень защиты.
- Синхронизация состояний поддерживается лавинной рассылкой с использованием алгоритма Trickle. Взаимная синхронизация и согласование не поддерживаются.

- Профиль HNCP для DNCP разработан для применения между соединёнными напрямую соседями на общем канале с использованием UDP и адресов IPv6 link-local.

DNCP не соответствует требованиям к протоколу согласования общего назначения, поскольку он разработан специально для лавинной синхронизации. В профиле HNCP этот протокол ограничен сообщениями link-local и протоколом IPv6. Тем не менее, это по меньшей мере интересный пример такого стиля взаимодействия между устройствами без центрального органа, обеспечивающий проверенный метод синхронизации в масштабе сети за счёт лавинной рассылки.

Протокол синхронизации серверных кэшей (Server Cache Synchronization Protocol или SCSP) [RFC2334] описывает метод синхронизации и репликации кэшей между группой узлов.

Несколько лет назад был предложен основанный на IP базовый протокол управления (Generic Control Protocol или IGCP) [IGCP], предназначенный для обмена информацией и согласования, но без прямого обнаружения партнёров. У этого протокола много общего с описанным здесь.

Ни одно из упомянутых выше решений не соответствует полностью требованиям к базовому обнаружению, синхронизации состояний и согласованию в едином решении. Многие из протоколов предполагают работу в традиционном вертикальном (top-down) сценарии или модели «север-юг», а не в гибком варианте одноранговой (партнерского) взаимодействия. Многие из протоколов так или иначе специализированы. В результате не удалось найти комбинацию существующих протоколов, соответствующую требованиям Приложения B. Не найдено также путей расширения имеющихся протоколов в соответствии с этими требованиями.

Благодарности

Основной вклад в предварительные версии этого документа внёс Sheng Jiang, а также важно было участие Toerless Eckert. Существенная часть первоначального рецензирования была представлена Joel Halpern, Barry Leiba, Charles E. Perkins, Michael Richardson. William Atwood оказал значительную помощь в отладке прототипа реализации.

Важные комментарии внесли Michael Behringer, Jéferson Campos Nobre, Laurent Ciavaglia, Zongpeng Du, Yu Fu, Joel Jaeggli, Zhenbin Li, Dimitri Papadimitriou, Pierre Peloso, Reshad Rahman, Markus Stenberg, Martin Stiemerling, Rene Struik, Martin Thomson, Dacheng Zhang, члены исследовательской группы Network Management, рабочей группы ANIMA и IESG.

Адреса авторов

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany
Email: cabo@tzi.org

Brian Carpenter (редактор)
School of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand
Email: brian.e.carpenter@gmail.com

Bing Liu (редактор)
Huawei Technologies Co., Ltd
Q14, Huawei Campus
Hai-Dian District
No.156 Beiqing Road
Beijing
100095
China
Email: leo.liubing@huawei.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru