# АНАЛИЗ ПРОТОКОЛОВ MESH-СЕТЕЙ

*П.А. Балаклеевский, О.С. Терновой*
*Алтайский государственный университет, г. Барнаул*

В настоящее время актуальным является вопрос об обеспечении удаленных от линий связи территорий доступом к сети Интернет. К таким территориям можно отнести как малые населенные пункты, так и места, которые находятся на удалении от них, например, сельскохозяйственные поля, где сеть необходима для модернизации производства за счет внедрения погодных датчиков или меток для автоматизированной сельскохозяйственной техники, управляемой оператором удаленно. Традиционные провайдеры зачастую не считают целесообразным обеспечивать данные территории сетевым покрытием. Однако, современные технологии позволяют решить данную проблему возможностями самих пользователей. Реализовать подобную идею возможно с использованием ячеистых одноранговых сетей, или сетей, использующих распределенные вычисления, именуемых mesh-сетями. Предложение данного решения в качестве альтернативного доступа в Интернет обусловлено тем, что для создания и получения доступа к подобной сети достаточно иметь домашний роутер, а в качестве узла сети может быть не только сетевое оборудование, но и любая другая техника, имеющая сетевой интерфейс, от ноутбуков и смартфонов, до устройств интернета-вещей, проблему покрытия подобной сети можно решить с помощью добавления в состав сети направленных антенн, которые способны увеличить покрытие одного узла сети до 15 километров.

В данной статье будут рассмотрены суть работы сетей с распределенными вычислениями, а также существующие в данной области протоколы с целью выбора оптимального решения с точки зрения простоты использования, модификации, стабильности соединения, поддержки основных операционных систем, а также безопасности передачи и хранения информации. При выделении оптимального протокола для поставленной задачи будет предложено возможное улучшение протокола с точки зрения обеспечения информационной безопасности пользователей распределенной сети.

Структурную схему mesh-сетей можно наблюдать на рисунке 1:

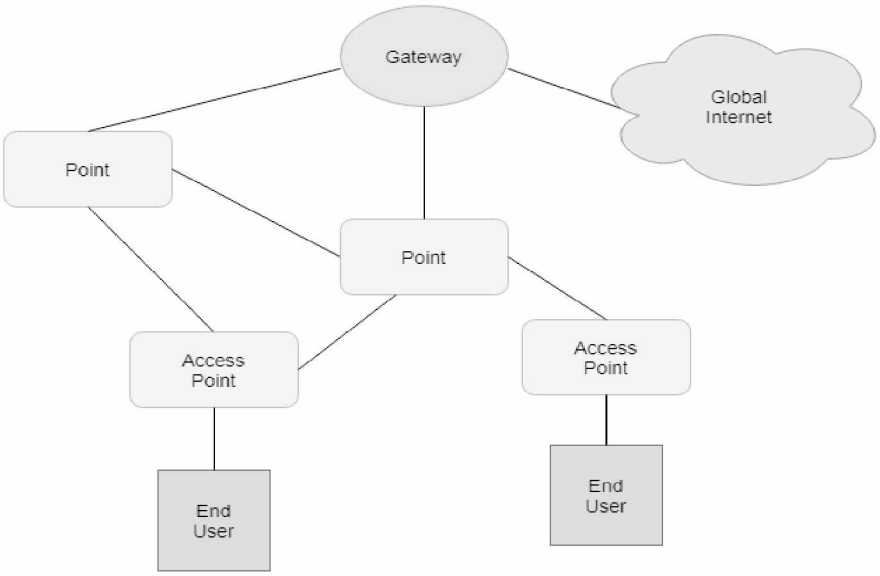


Рисунок 1 – Структурная схема Mesh-сети

Здесь представлены основные участники функционирующей распределенной сети: Gateway – непосредственно источник сети, к которому подключается канал Uplink; Access Point – точка доступа для подключения клиентов, точками доступа могут являться не только маршрутизаторы, но и сами рабочие станции пользователей; Mesh Point – точка доступа, использующаяся для транзита пакетов внутри сети (трафик проходит через них из учета оптимального количества хопов в сети для уменьшения задержки); End Users – конечные узлы (пользователи) сети, как правило имеют один сетевой выход из-за чего к ним нельзя подключить больше одного соединения. Стоит отметить, что данные роли не зафиксированы за участниками сети, а могут изменяться при изменении структуры сети. Например, при отключении Point остальные участники просчитывают новые пути маршрутизации для доступа к потерянным соседям. Также да данном изображении можно наблюдать, что, если один из узлов сети имеет доступ к сети Интернет, последним могут пользоваться все узлы сети от лица узла, предоставляющего доступ [1].

На данный момент наиболее популярными протоколами ячеистых сетей являются следующие решения: DTN; B.A.T.M.A.N; IEEE 802.11S; CJDNS.

Протокол DTN (delay-tolerant networking) изначально предназначен для использования в нестабильных окружающих средах. Принцип работы основан на работе группы маршрутизаторов, в роли операционной системы которых выступает дистрибутив Linux OpenWRT. Данное решение позволяет использовать обычные маршрутизаторы в качестве точек доступа к сети DTN путем прошивки программного обеспечения. Точки могут работать в режимах «точка-точка» и «точка-многоточка», при этом определяя оптимальный путь отправки сообщений. Сами сообщения в данном протоколе представляют из себя пакеты (bundle), как правило состоящие из минимум двух блоков: в первом хранится идентификатор получателя (EID), который может указывать как на один, так и на группу узлов сети, тем самым осуществляя групповую рассылку, второй же хранит передаваемые данные. Время хранения пакетов данных в данном протоколе несколько больше, чем в остальных протоколах, что используется при сценарии использования в сложных условиях окружающей среды для исключения потери данных при передаче.

Не смотря на продуманную систему сохранения соединения между узлами, DTN не имеет встроенного решения в плане шифрования сообщений. Так как DTN зачастую функционирует в сетях с нестабильным соединением криптографические алгоритмы не всегда могут работать корректно из-за замедления процесса обмена ключами. Также стоит отметить, что DTN он не имеет функции авто-назначения адреса, при котором клиент способен выбирать себе адрес, а также заменять его при переходе в другую подсеть. Помимо этого, протокол DTN не имеет возможности объединять сети, что сказывается на невозможности реализации одного из основных преимуществ Mesh-сетей, а именно высокой масштабируемости. Последним же недостатком является отсутствием возможности работы в фоновом режиме, наряду с обычным интернет - соединением, что может поставить под сомнение использование данного протокола рядовым пользователем [2].

Следующим протоколом для рассмотрения является B.A.T.M.A.N. Принцип работы данного протокола основывается на идее о разделении знаний об оптимальных маршрутах между участниками сети на все участвующие узлы. Каждый узел поддерживает только информацию об оптимальном следующем «прыжке» (hope) по отношению к другим соседним узлам. Таким образом, потребность в знании о возможном изменении топологии сети становится ненужной, тем самым ослабляя нагрузку на ограниченные ресурсы процессоров маршрутизаторов.

Протокол B.A.T.M.A.N. аналогично DTN способен встраиваться в систему Linux, включая дистрибутив OpenWRT, причем существует возможность интеграции в само ядро Linux, позволяя создавать собственные дистрибутивы уже со встроенной

поддержкой B.A.T.M.A.N. Также присутствует поддержка автоконфигурируемой маршрутизации, что означает отсутствие необходимости в ручной настройке маршрутизации.

Данный протокол имеет аналогичные с DTN недостатки, такие как отсутствие авто-назначения адресов, объединения сетей, шифрования трафика внутри сети, а также фонового режима работы, поэтому B.A.T.M.A.N. слабо подходит для настройки и использования рядовым пользователем, а также создания защищенной сети на предполагаемом предприятии, что не входит в поставленную изначально задачу [3].

IEEE 802.11s по своей структуре напоминает B.A.T.M.A.N., однако имеет иные системы метрик, а именно он требует, чтобы все устройства, входящие в сеть, поддерживали метрику времени передачи в канале, или Airtime Link Metric. Данная метрика задается следующей формулой:

$$C_a = \frac{(O+\frac{B_t}{r})}{(1-e_t)} \quad (1)$$

где $B_t$– число битов в текстовом пакете (обычно 8192 бит); $O$ – накладные расходы доступа к каналу, которые включают в себя заголовки пакетов, кадры доступа и т.д.; $r$ – скорость передачи данных в канале (Мбит/с); $e_t$– вероятность возникновения ошибки.

В стандарте IEEE 802.11s используется гибридный беспроводной mesh-протокол маршрутизации (HWMP), работающий в реактивном и проактивном режимах, которые отличаются методом построения таблиц маршрутизации, в первом случае обновление информации об узлах происходит непосредственно перед передачей трафика, а во втором – периодически по запросу корневого узла. [4]

Для исключения зацикливания широковещательных рассылок от узлов сети при инициации обмена сообщениями вводятся дополнительные параметры: порядковый номер назначения, или Destination Sequence Number (DSN) и DSN инициатора Originator's DSN (OSN). Именно последний является порядковым номером при рассылке пакетов поиска пути, тогда как DSN встроено в каждый узел mesh – сети. Перед началом поиска пути DSN узла-отправителя увеличивается на единицу и записывается в OSN пакета PREQ. Каждый узел при получении данного пакета сравнивает значение OSN с ранее известным для этого же отправителя и пересылает пакет дальше при условии, что текущий OSN в пакете больше или равен сохраненному ранее значению, в противном случае пакет отбрасывается. [5]

Однако, не смотря на продуманность алгоритма маршрутизации стандарт IEEE 802.11s имеет ряд недостатков, таких как отсутствие end-to-end шифрования трафика внутри сети, отсутствует возможность объединять уже сформированные сети, а также, как и в B.A.T.M.A.N. исключена возможность авто-конфигурирования адресов новых узлов сети.

CJDNS в отличие от остальных представленных в данной работе протоколов, отличается наличием возможности объединения созданных ранее подсетей, а также наличием шифрования внутри сети, что больше подходит для создания необходимой сети. Уже сейчас на основе Cjdns создается проект Hyperboria, целью которого является создание масштабной распределенной сети, способной создать конкуренцию традиционному Интернету.

Функционирование каждого узла, основанного на протоколе Cjdns основано на трех основных компонентах, заложенных в одном устройстве, а именно коммутатор, маршрутизатор и криптографический модуль CryptoAuth. Помимо подхода получения оптимальных маршрутов в Cjdns существует централизованный подход получения маршрутов, с использованием суперноды (суперузла) – узла в сети Cjdns, который сканирует сеть на наличие новых пользователей системы и при их наличии заносит их

данные в базу. С помощью подобного узла, любой пользователь сети может запросить оптимальный маршрут до любого узла сети и получить его. [6]

Модуль CryptoAuth используется для аутентификации узлов сети и шифрования пакетов во время их передачи. В данном модуле используется система из публичных и приватных ключей, которые выдаются каждому пользователю сети при подключении. При использовании протокола Cjdns невозможно реализовать атаки типа «человек посередине», а также DPI. Первая атака невозможна по причине того, что передаваемая информация зашифрована публичным ключом получателя и может быть расшифрована его приватным ключом, промежуточные же узлы выполняют лишь транзитную роль, без возможности чтения содержимого пакета. Атака DPI в свою очередь не может быть выполнена по этой же причине, сторонние узлы, не участвующие в обмене не способны анализировать трафик в принципе, что, собственно, и осуществляется при DPI. Таким образом в протоколе Cjdns обеспечивается сохранность персональных данных и иной информации, не планирующейся быть публичной. [6]

Более подробно результаты проведенного анализа приведены в таблице 1:

Таблица 1 - Сравнение основных особенностей протоколов маршрутизации Mesh-сетей

| Протокол/ Особенности | DTN | B.A.T.M.A.N. | IEEE 802.11s | CJDNS |
|---|---|---|---|---|
| Авто-назначение адреса | - | - | - | + |
| Авто-конфигурация маршрутизации | + | + | + | + |
| Распределенная маршрутизация | + | + | + | + |
| Поддержка IPv4 | + | + | + | - |
| Поддержка IPv6 | + | + | + | + |
| Шифрование трафика | - | - | - | + |
| Поддержка Linux | + | + | + | + |
| Интеграция в ядро Linux | - | - | - | + |
| Поддержка Windows | + | - | Ведется разработка | + |
| Поддержка Mac OS X | + | + | + | + |
| Работа в фоновом режиме | - | - | + | + |
| Объединение сетей через Интернет | - | - | + | + |

С точки зрения защиты информации и формирования защищенной сети наилучшим решением будет являться протокол Cjdns прежде всего из-за встроенной поддержки криптозащиты, наличием авто-назначения адресов узлов при их подключении, что способствует наиболее активному расширению сети, а также способностью работать в фоновом режиме наряду с сетью Интернет и объединять через него mesh-сети, находящиеся на удаленном расстоянии друг от друга, что может пригодится при построении сети предприятия, имеющей несколько филиалов или в регионах с низкой плотностью населения.

Перспективы развития mesh-сетей, и в частности протокола CJDNS заключаются в добавлении новых возможностей и упрощении установки нового программного обеспечения, поддерживающего режим mesh. Предлагается создать собственный дистрибутив, основанный на ОС Linux OpenWRT с добавлением модуля CJDNS, а также встроенным графическим интерфейсом, позволяющим более наглядно настраивать функционал точки доступа.

Несмотря на то, что протокол CJDNS имеет встроенную поддержку ассиметричного шифрования трафика, у данного подхода могут иметься некоторые проблемы в плане взлома закрытых ключей, что в ближайшее время станет актуальным, так как мощность вычислительных машин будет позволять расшифровывать сообщения, зашифрованные одним ассиметричным шифрованием. Для решения этой возможной угрозы предлагается внедрить альтернативный способ шифрования данных, основанный на нескольких алгоритмах, таких как Twofish и RSA. Первый позволит оптимизировать настройку времени шифрования и повысит сложность ключа за счет своей блочной структуры. В основном, алгоритм шифрования Twofish имеет 16 раундов шифрования, а 128-битный шифрованный текст создается после завершения 16-го раунда. Twofish обеспечивает эффективную безопасность, так как он был достаточно подробно проанализирован и на сегодняшний момент технологии позволяют сломать только пять раундов алгоритма. [7] Использование же алгоритма RSA позволит приблизится к стандартному шифрованию, использующимся в протоколе Cjdns, так как первый использует в своей структуре ассиметричное шифрование с использованием открытых и закрытых ключей. Помимо использования данных алгоритмов шифрования предлагается внедрение дополнительного хэширования трафика на основе алгоритмов SHA-1 или MD5, выполняющихся перед этапом шифрования.

Таким образом, предложенный алгоритм шифрования будет являться наиболее безопасным за счет одновременного использования ассиметричного алгоритма RSA и функций хэширования, а также оптимальным с точки зрения времени обработки сообщений за счет использования наработок симметричного шифрования Twofish, что в теории даст качественный прирост к обеспечению информационной безопасности протокола Cjdns.

**Библиографический список.**


1. WiFi Mesh сеть [Электронный ресурс]\\ Nastroisam. 2018. URL: https://nastroisam.ru/wifi-mesh-network-chto-takoe/;
2. RFC 4838 – Delay-Tolerant Networking Architecture [Электронный ресурс]\\ Tools.ietf. 2007. URL: https://tools.ietf.org/html/rfc4838#section-7;
3. B.A.T.M.A.N. Concept – Open-Mesh [Электронный ресурс]\\ Open-mesh. 2006-2016. URL: https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept;
4. Вишневский В., Лаконцев Д., Сафонов А., Шпилев С. Маршрутизация в широкополосных беспроводных mesh-сетях стандарта IEEE 802.11s // Электроника: Наука, технология, бизнес. - 2008. - № 6(88). - С. 64-71;
5. HOWTO /open80211s Wiki [Электронный ресурс]\\ Github. 2019. URL: https://github.com/o11s/open80211s/wiki/HOWTO;
6. Cjdns – теория и практика [Электронный ресурс]\\ Netwhood. 2019. URL: http://netwhood.online/2018/10/21/cjdns-theory-and-practice/;
7. Mirza F., Murphy S. An Observation on the Key Schedule of Twofish / Proceedings of the second AES candidate conference. - 1999. – С.151-154.