
Table of Contents

| | |
|---|---------|
| Introduction | 1.1 |
| WiFi | 1.2 |
| Лекция 1: Многошаговые сети: Manet | 1.2.1 |
| Лекция 2: Многошаговые сети: Критерии | 1.2.2 |
| Решение ДЗ | 1.2.2.1 |
| Лекция 3: Mesh сети | 1.2.3 |
| Лекция 4: Mesh + МССА | 1.2.4 |
| Лекция 5: Высокоскоростной WiFi: Обзор | 1.2.5 |
| Лекция 6: 802.11ax | 1.2.6 |
| Лекция 7: 802.11n, 802.11ac | 1.2.7 |
| Лекция 8: 802.11ad, ay: WiFi в mm диапазоне | 1.2.8 |

As i want so that

Все лекции в одном [pdf](#)

Современные сети WiFi

Хоров Евгений Михайлович

1997г — первая версия wifi

Все что мы изучали раньше (...) это было просто, теперь все будет сложнее

План Курса

- многошаговые сети wifi
- высокоскоростные сети wifi
- плотные сети
- WiFi 60ГГц
- Актуальные задачи и новые направления

Будут кр и самостоятельные работы

$$X = T * I * F$$

- T -- оценка за тесты
- I -- оценка за индивидуальную работу со стандартом и статьями
- F -- финальная оценка

Многошаговые сети Wifi

Задачи:

- Увеличить покрытие
- Повысить емкость (хз что такое емкость)

Идея — передавать цепочкой через промежуточные узлы

802.11s (появились многошаговые сети)

Название

Существует 2 схожих названия для многошаговых сетей

- Mesh сеть
 - IEEE
 - сеть в которой маршрутизация работает на канальном уровне
 - стационарная сеть
- MANET — Mobile Ad hoc NETwork
 - IETF
 - маршрутизация выполняется на сетевом уровне
 - мобильная сеть

Мы будем понимать многошаговую сеть следующим образом:

Многошаговая беспроводная сеть — сеть, в которой доставка информации возможна через промежуточные узлы, выступающие в роли ретрансляторов

Задачи построения многошаговых сетей

(По сути это все нужно для обеспечения маршрутизации)

- обнаружение соседей
- управление соединений с ними (логическими, для Health-статусов)
- Оценка качества соединений (чтобы решать задачу маршрутизации оптимальным образом)
- Рассылка сетевой информации
- Построение маршрутов
- Ретрансляция

Особенность — динамическая сетевая среда, очень быстро меняются условия

Лавинная рассылка

Широковещательная рассылка с TTL, не меньше чем диаметр сети

Плюсы:

- простота
- устойчивость к высоким скоростям движения узлов
- относительная надежность доставки при наличии большой избыточности маршрутов
- “естественная оптимальность“ пути (НО много узлов делят один канал => лишние ожидания)

Минусы:

- большие накладные расходы (каждый пакет повторяется до N-1 раз)
- необходимость механизма разрешения / избегания коллизий широковещательных пакетов
 - Необходимо, т.к. на широковещательные пакеты нет ACK
- Высокие потери (видимо эффективности), если маршрут (или его часть) единственный

Протоколы маршрутизации

OLSR и NHDP

Optimized Link State Routing Protocol — RFC 3626 — <https://tools.ietf.org/html/rfc3626>

OLSR v2 — RFC 7131

Мы изучаем первую версию протокола

Neighborhood Discovery Protocol (**NHDP**) — RFC 6130

- Проактивный протол, рассылающий информацию о наличии соединений
- Не учитывает качество соединений. Оптимизация объема рассылаемой информации

Работает как программа поверх канального уровня.

Программа рассылает UDP пакеты, через которые происходит обмен информации, по ним составляются таблицы маршрутизации

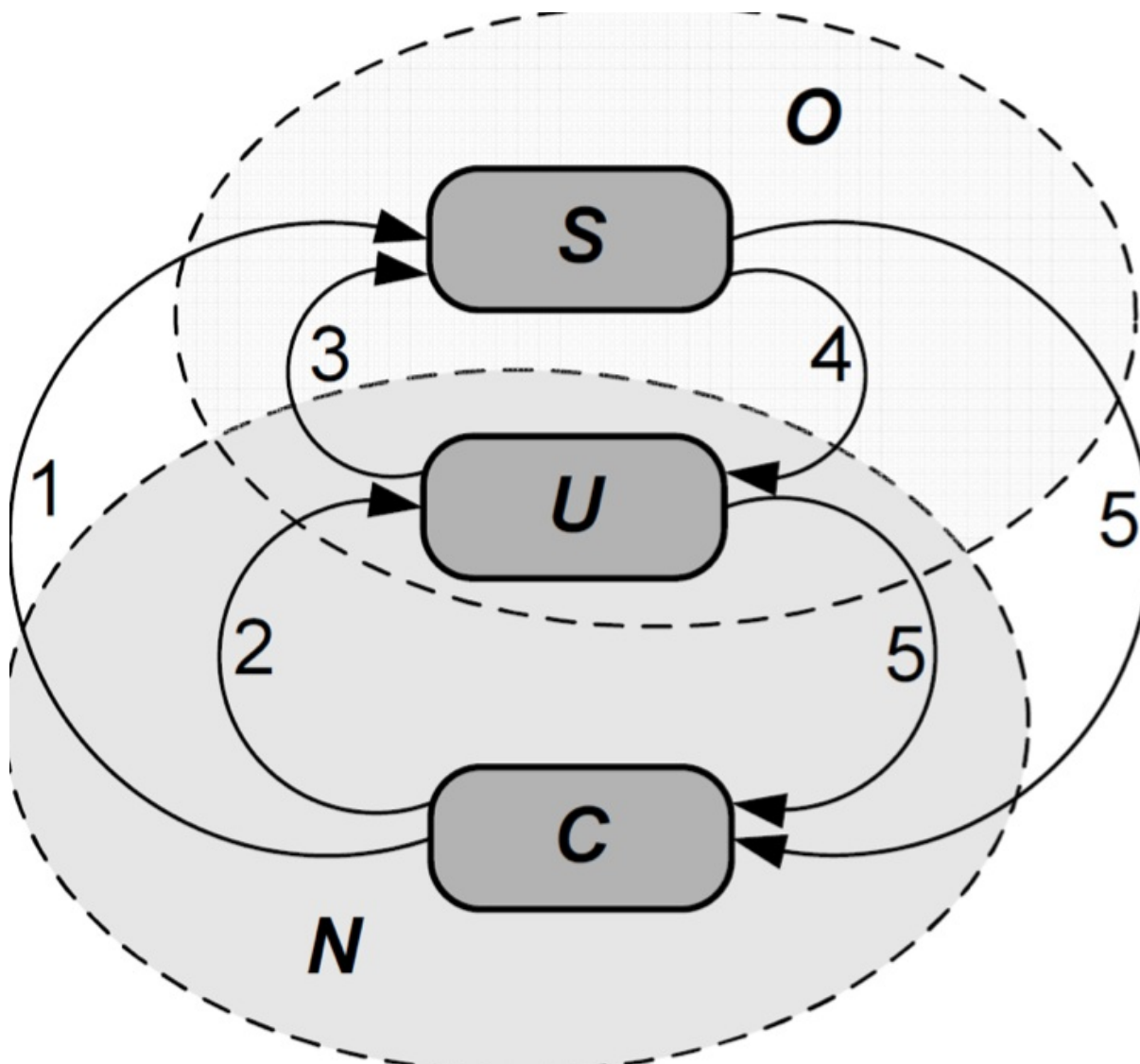
Обнаружение соседей

[illegible]

- периодическая рассылка Hello сообщений (Link Sensing)
 - Интервал HELLO_INTERVAL = 2s
 - Не пересылаются дальше

- Проблема синхронизации

- Джиттер — от планируемого времени генерации пакета отнимается случайное время $[0 \dots \text{MAXJITTER}]$, $\text{MAXJITTER} = \text{HELLO_INTERVAL} / 4$
- Actual message interval = MESSAGE_INTERVAL - jitter



- Hello сообщение содержит:
 - список (адресов) соседей про которые знает станция
 - Состояния соединений (содержится в поле link code)
 - S = symmetric (получили hello и там есть наш адрес)
 - U = однонаправленные (услышали о ком-то, но он о нас не знает) (Heard)
 - C = Закрытые / потерянные (Closed/Lost) — переходим если в течение NEIGH_HOLD_TIME не пришло не одного Hello сообщения.
- Также содержит информацию о том, считает ли сосед принимающую станцию MPR

Знаем полную топологию в двухшаговой окрестности узла.

MPR узлы (Multipoint Relay) — те симметричные одношаговые соседи, через которые можно достигаться до двухшаговых.

The idea of multipoint relays is to minimize the overhead of flooding messages in the network by reducing redundant retransmissions in the same region.

The neighbors of node N which are *NOT* in its MPR set, receive and process broadcast messages but do not retransmit broadcast messages received from node N.

Each node selects its MPR set from among its 1-hop symmetric neighbors. This set is selected such that it covers (in terms of radio range) all symmetric strict 2-hop nodes.

The MPR set of N, denoted as MPR(N), is then an arbitrary subset of the symmetric 1-hop neighborhood of N which satisfies the following condition: every node in the symmetric strict 2-hop neighborhood of N must have a symmetric link towards MPR(N).

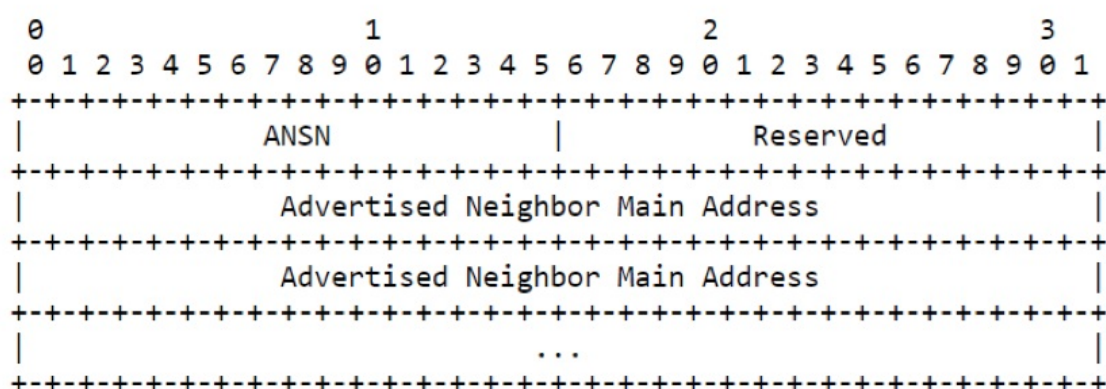
Each node maintains information about the set of neighbors that have selected it as MPR. This set is called the "Multipoint Relay Selector set" (MPR selector set) of a node. A node obtains this information from periodic HELLO messages received from the neighbors.

Пытается построить минимальное количество MPR узлов

Построение через жадный алгоритм

A. Qayyum, L. Viennot, A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. 35th Annual Hawaii International Conference on System Sciences (HICSS'2001).

Рассылка сообщений TC (Topology Control)



- Topology Control
- Рассылается broadcastom
- Раз в TC_INTERVAL (5 секунд)
- Пересылаются MPR узлами
- Содержат информацию только об MPR Selector Set
- Высылает список адресов (только тех, кто выбрал данный узел в качестве MPR) с которыми установлены симметричные соединения
- Ретранслируются дальше только MPR-ми

Таким образом получается подсеть, в которой не увеличиваются длины маршрутов (по метрике hop-count)

Поиск маршрута и таблица маршрутизации

1. R_dest_addr R_next_addr R_dist (число шагов) R_iface_addr
2. R_dest_addr R_next_addr R_dist R_iface_addr
3. ...

R_dest_addr — куда нужно доставить пакет

R_next_addr — адрес next-хопа

R_dist — метрика

R_iface_addr — интерфейс

Обновляется каждый раз, когда происходят изменения.

Заполняется по очереди: сначала по информации о соседях, затем двухшаговых соседях, затем по остальным соединениям в порядке их удаления от узла.

Обновляется каждый раз (конкретно полностью), когда происходят изменения:

- link set (содержит информацию о соединениях с соседями и их статусе)
- neighbor set (содержит информацию о соседях и их статусе)
- 2-hop neighbour set (содержит информацию о двухшаговых соседях)
- the topology set (содержит информацию о “дальних” соединениях, полученную из ТС)
- Multiple Interface Association Information Base (содержит информацию о соответствии нескольких адресов одному узлу (распространяется через MIB сообщения))

1. Потворить прошлый семестр (нужна будет в этом году)

Дополнительная информация

Also, OLSR does not require sequenced delivery of messages. Each control message contains a sequence number which is incremented for each message. Thus the recipient of a control message can, if required, easily identify which information is more recent - even if messages have been re-ordered while in transmission.

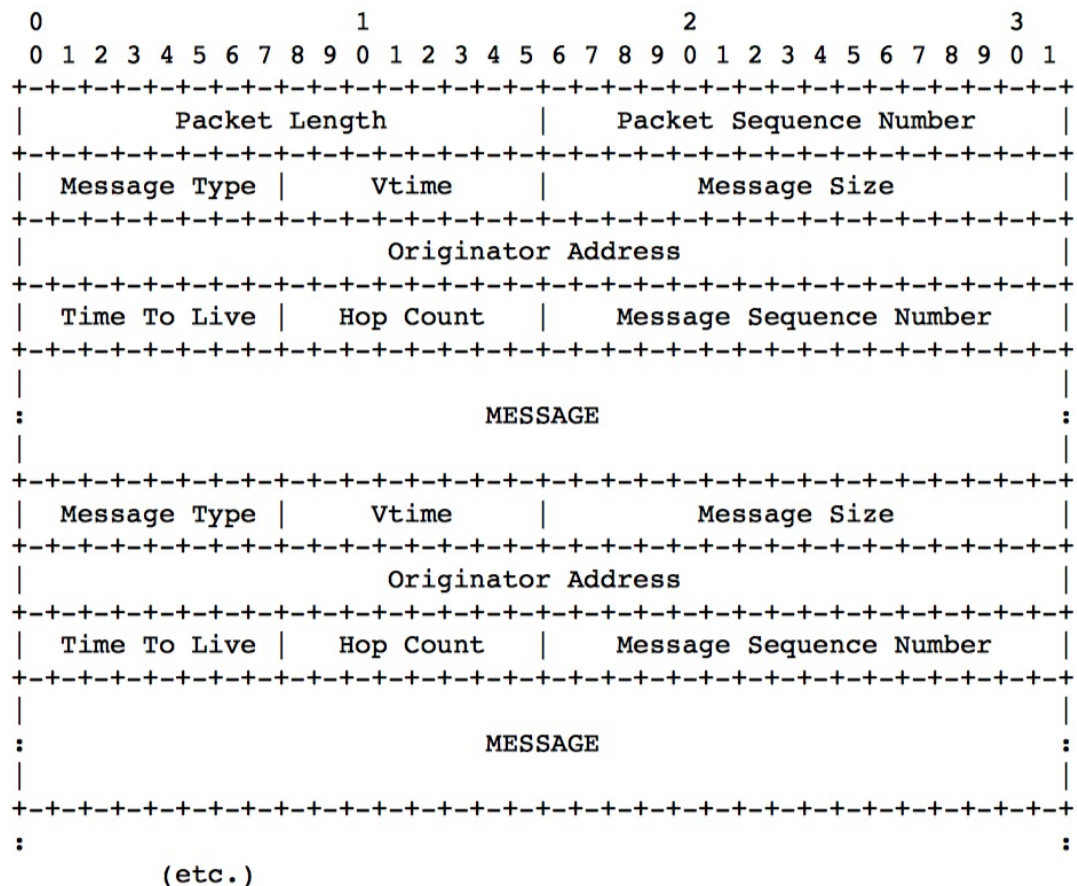
OLSR does not require any changes to the format of IP packets. Thus any existing IP stack can be used as is: the protocol only interacts with routing table management.

Формат пакета

OLSR communicates using a unified packet format for all data related to the protocol. The purpose of this is to facilitate extensibility of the protocol without breaking backwards compatibility. This also provides an easy way of piggybacking different "types" of information into a single transmission, and thus for a given implementation to optimize towards utilizing the maximal frame-size, provided by the network. These packets are embedded in UDP datagrams for transmission over the network. The present document is presented with IPv4 addresses.

Packets in OLSR are communicated using UDP. Port 698 has been assigned by IANA for exclusive usage by the OLSR protocol.

The basic layout of any packet in OLSR is as follows (omitting IP and UDP headers):



- The Packet Sequence Number (PSN) MUST be incremented by one each time a new OLSR packet is transmitted.
 - Нужно для того, чтобы понимать какую информацию считать более свежей
- The IP address of the interface over which a packet was transmitted is obtainable from the IP header of the packet.
- Vtime

This field indicates for how long time after reception a node MUST consider the information contained in the message as valid, unless a more recent update to the information is received. The validity time is represented by its mantissa (four highest bits of Vtime field) and by its exponent (four lowest bits of Vtime field). In other words:

$$\text{validity time} = C(1 + a/16) 2^b \text{ [in seconds]}$$

where a is the integer represented by the four highest bits of Vtime field and b the integer represented by the four lowest bits of Vtime field. The proposed value of the scaling factor C is specified in section 18.

- Originator Address

This field contains the main address of the node, which has originally generated this message. This field SHOULD NOT be confused with the source address from the IP header, which is changed each time to the address of the intermediate interface which is re-transmitting this message. The Originator Address field MUST NEVER be changed in retransmissions. — WTF

- Hop Count

This field contains the number of hops a message has attained. Before a message is retransmitted, the Hop Count MUST be incremented by 1.

Initially, this is set to '0' by the originator of the message.

- Message Sequence Number

While generating a message, the "originator" node will assign a unique identification number to each message. This number is inserted into the Sequence Number field of the message. The sequence number is increased by 1 (one) for each message originating from the node. Message sequence numbers are used to ensure that a given message is not retransmitted more than once by any node.

Недостатки NHDP

NHDP открывает соединения по первому полученному HELLO сообщению и закрывает по тайм-ауту. Недостатки:

- Ненадежность (низкая вероятность успешной передачи)
 - например получили только 1/3 от hello сообщений
 - не удовлетворяет требованию к доле доставленных пакетов из-за ограничения на число попыток передачи
 - высокое потребление канальных ресурсов из-за большого числа повторов передачи
 - хочется накладывать требования надежности
- Нестабильность (большая флуктуация состояния соединения)
 - постоянное изменение состояний узлов после новых сообщений
 - возможные ошибки маршрутизации (что-то вроде счетчика до бесконечности, следствие — циклы)

Возможное исправление:

Увеличение объема статистических данных на основании которого принимается решение об открытии соединения

- например наблюдать за параметрами hello сообщений (например уровень сигнала)
 - это трудно, тк протокол работает на уровне приложений, нужны драйверы и все такое
 - => нужны другие методы

Критерий надежности

Пусть задана некоторая вероятность $= p_0$

Пусть вероятность успешной доставки $= p$

Если $p > p_0$ то нужно close, если $p < p_0$ то нужно Symmetric

Критерий стабильности

Состояние то симметричное то не симметричное

T_s — среднее время жизни линка (логического соединения (когда соединение симметрично))

T_N — среднее время нахождения в Not Sym.

$g = \frac{1}{T_s + T_N}$ — link fluctuation (колебания)

T_{update} — интервал обновления топологии (характерный период рассылки сетевой информации)

Сам критерий:

$$\forall p \Rightarrow \frac{1}{g(p)} \gg 2T_{update}$$

Критерий оперативности

Важно, чтобы T_{delay} (задержка до установки логического соединения) было много меньше T_{link} (время физического соединения)

$$T_{delay} \ll T_{link}$$

Математическая модель

Говорим, что если хотя бы один из критериев не выполняется — у протокола маршрутизации большие проблемы.

Нужно построить математическую модель протокола, чтобы вычислять все эти значения.

Необходимо определить:

π_s — вероятность того, что соединение симметрично

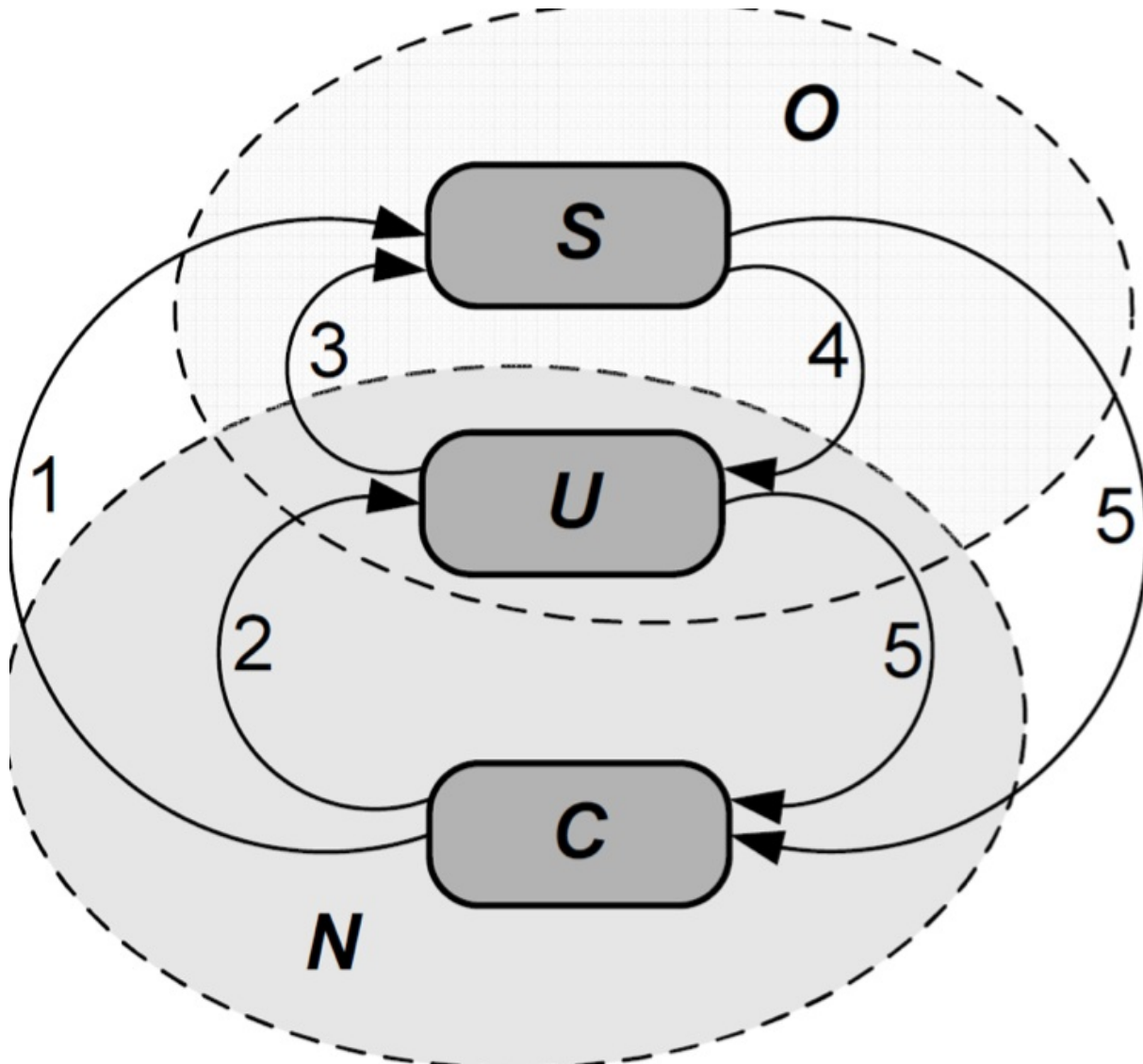
T_S — среднее время когда соединение симметрично

T_N — среднее время когда соединение несимметрично

Упрощаем протокол, с точки зрения модели

Новая диаграмма состояний: (точнее правила перехода)

Важно обратить внимание на состояния $O = S + U$ и $N = U + C$



- Переходы из C в U и из C в S происходит, когда узел получает r HELLO подряд от соседа [мб что-то улучшить, дополнительная степень свободы]

(Считаем, что HELLO генерируется строго периодически)

- Переход в C, когда потеряли s HELLO подряд. [тупо упрощаем модель]

Критерий Вальда — [link](#)

Сумма случайного числа одинаково-распределенных независимых случайных величин

Тождество Вальда

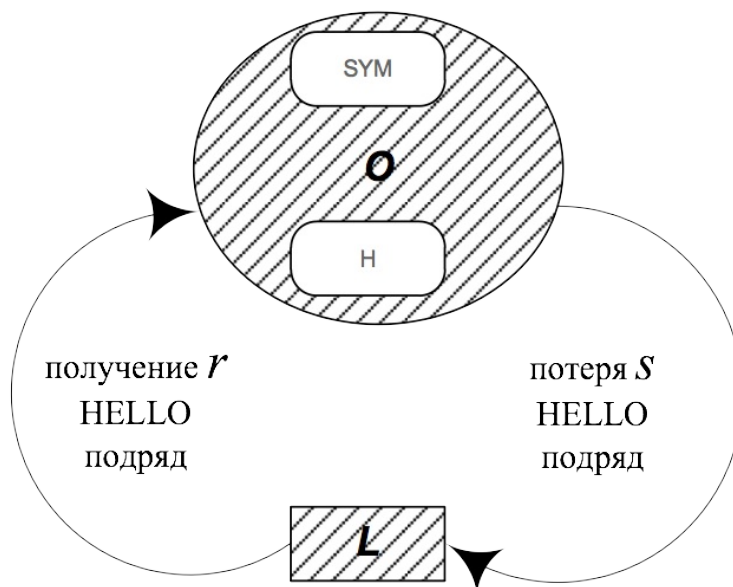
$$E(\sum_{i=1}^N X_i) = E(N)E(X)$$

Определение вероятности нахождения в О(С)

Процесс $J_{oc}(t)$ перехода между состояниями О и С является On-Off процессом

($\langle X \rangle$ — средняя длительность состояния X)

$$\pi_0 = \frac{\langle T_o \rangle}{\langle T_o \rangle + \langle T_c \rangle}$$

Определение средних длительностей состояний О,С

ДЗ 1

$$\langle T_o \rangle = \frac{1 - (1-p)^s}{p(1-p)^s}$$

$$\langle T_c \rangle = \frac{1 - p^r}{p^r(1-p)}$$

(p вероятность единицы)

(выводится через формулу Вальда)

1#0001#1#01#01#0000000# (1 получили Hello, 0 потеряли)

Цикл продолжается до первой единицы или до s нулей.

Вероятность того, что длина цикла = s: $(1 - p)$

Какова средняя длина цикла?

Каково среднее число циклов?

Короче нужно ответить на эти вопросы и вывести формулы выше

Определение вероятности нахождения в состоянии S

Доказать утверждение

ДЗ 2

$$\pi_s = \pi_o^2$$

Подсказка: Состояние симметричное с точки зрения узла X, если оно открыто с точки зрения узла X и в последнем принятом HELLO от узла Y оно было указано как открытое

Доказать утверждение

ДЗ 3

$\langle Ts \rangle \approx \frac{\langle T_o \rangle}{2}$ (средняя длительность пересечений — смотреть картинку)

Подсказка:

- Важно знать, что отрезки одинаково распределены
- Считать, что нам известно распределение длины отрезка, оно вконец сократится
- Это классическая задача из тервера

ДЗ 4

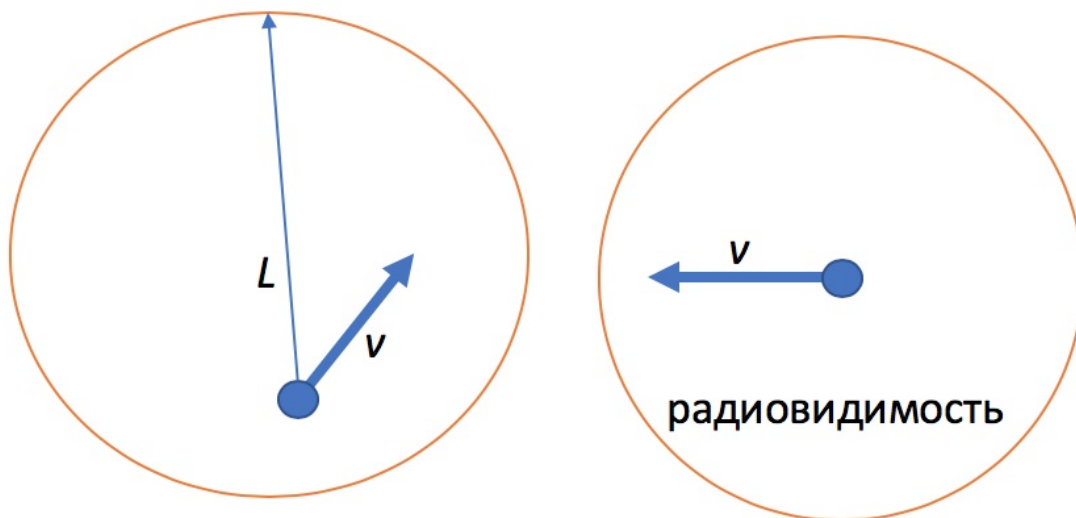
Найти T_N среднее время в нахождении $C + U$

Итого: 4 ДЗ

ДЗ 5*

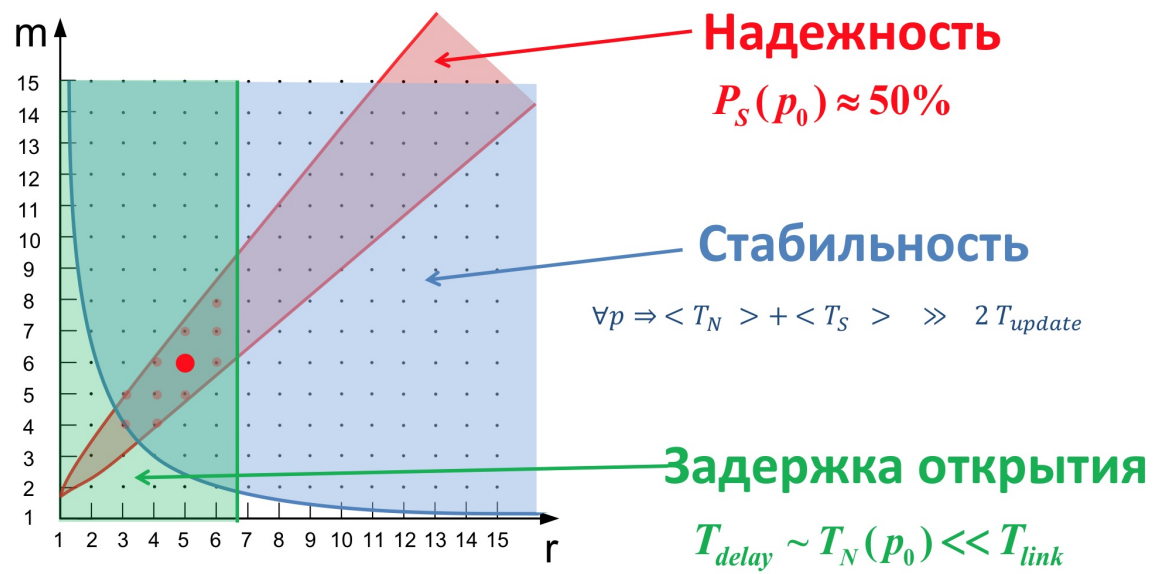
- 5-е дз со * (Среднее время жизни физического соединения)

$\langle T_{link} \rangle = \frac{\pi^2 * L}{8v}$ среднее время движущихся объектов в зоне слышимости



Настройка параметров

Тут что-то говорится про выбор r , s и влияние на разные критерии, проще смотреть слайд.



Что такое здесь m — неясно (видимо s)

Дз к лекции 2 — Аналитическая модель OLSR

Автор: Войнов Никита Группа: 521

ДЗ 1

Доказать

$$\langle T_O \rangle = \frac{1 - (1-p)^s}{p(1-p)^s}$$

$$\langle T_C \rangle = \frac{1-p^r}{p^r(1-p)}$$

Рассмотрим состояние O . Пусть успешному получению HELLO соответствует 1, отсутствию HELLO — 0.

Процесс выходит из состояния O при получении s нулей подряд.

Разобьем нахождение в состоянии O на циклы, цикл заканчивается при получении первой 1 или при получении s нулей.

Среднюю продолжительность состояния O можно определить как

$$\langle T_O \rangle = E(\sum_{i=1}^N X_i) = E(N)E(X),$$

где N количество циклов, X_i продолжительность цикла i .

Пусть $q = (1 - p)$.

Найдем $E(N)$ и $E(X)$.

$E(X)$

Вероятность того, что продолжительность цикла равна

1: p

2: $(q) * p$

s-1: $(q)^{s-2} * p$

s: $(q)^{s-1}$

Тогда

$$E(X) = s * q^{s-1} + p * \sum_{i=1}^{s-1} i * q^{i-1} = s * q^{s-1} + p * \frac{d}{dq} \left(\frac{q - q^s}{1 - q} \right) = \frac{1 - q^s}{1 - q}$$

$E(N)$

Вероятность того, что количество циклов равно

1: q^s

2: $q^s(1 - q^s)$

n: $q^s(1 - q^s)^{n-1}$

Тогда

$$E(N) = q^s \sum_{i=1}^{\infty} i * (1 - q^s)^{i-1} = - \frac{q^s}{s * q^{s-1}} * \frac{d}{dq} \left(\frac{1 - q^s}{q^s} \right) = \frac{1}{q^s}$$

Исходя из (1) и (2):

$$\langle T_O \rangle = \frac{1 - (1-p)^s}{p * (1-p)^s}$$

Проведем замену $1 - p \rightarrow p, s \rightarrow r$ получим:

$$\langle T_C \rangle = \frac{1-p^r}{p^r(1-p)}$$

ДЗ 2

$$\pi_s = \pi_o^2$$

Доказательство:

Состояние симметричное с точки зрения узла X, если оно открыто с точки зрения узла X и в последнем принятом HELLO от узла Y оно было указано как открытое.

Вероятность каждого из событий π_o , события независимы => доказано.

ДЗ 3

[1]

Рассмотрим 2 независимых процесса узлов A и B, меняющие состояния O и C, J^A и J^B . Определим процесс J_{SN} , который в состоянии S, если оба процесса в состоянии O, иначе он в состоянии N.

Докажем

$$\langle T_S \rangle = \frac{\langle T_O \rangle}{2}$$

Пусть f_t распределение продолжительности состояния O, а $F(t)$ его функция распределения. Положим, что $J_{SN}(t)$ меняет свое состояние на S в момент t_0 . Считаем, что J^A переходит в O в момент t_0 , а J^B уже там был.

Рассмотрим временной интервал в течение которого J^B остается в состоянии O, включая t_0 . Очевидно, что вероятность, что продолжительность данного интервала = τ равна $\frac{\tau f_\tau}{\langle T_O \rangle}$.

Рассмотрим часть интервала начинающуюся в t_0 . Функция плотности распределения этой части определяется следующим образом:

$$g(t) = \sum_{\tau > t} \frac{\tau f_\tau}{\langle T_O \rangle} * \frac{1}{\tau} = \frac{(1-F(t))}{\langle T_O \rangle}$$

Данная плотность отвечает следующей функции распределения:

$$G(t) = \frac{1}{\langle T_O \rangle} \int_0^t (1 - F(x)) dx$$

J_{SN} переходит из S, при первом выходе из O для J^A или J^B . Следовательно вероятность того, что продолжительность S не меньше x равна $(1 - F(x))(1 - G(x))$

По свойству неотрицательной случайной величины получаем

$$\langle T_S \rangle = \int_0^\infty (1 - F(x))(1 - G(x)) dx$$

$$\text{Т.к. } \frac{d}{dx} (1 - G(x)) = -\frac{1-F(x)}{\langle T_O \rangle}$$

$$\langle T_S \rangle = -\frac{\langle T_O \rangle}{2} (1 - G(x))^2 \Big|_0^\infty = \frac{\langle T_O \rangle}{2}$$

Доказано

ДЗ 4

Найти $\langle T_N \rangle$

Смена S и N — on-off процесс, поэтому аналогично π_o :

$$\pi_s = \frac{\langle TS \rangle}{\langle TS \rangle + \langle TN \rangle} = \pi_o^2$$

$$\langle T_N \rangle = \frac{\langle T_S^2 \rangle}{\pi_o} - \langle T_S \rangle$$

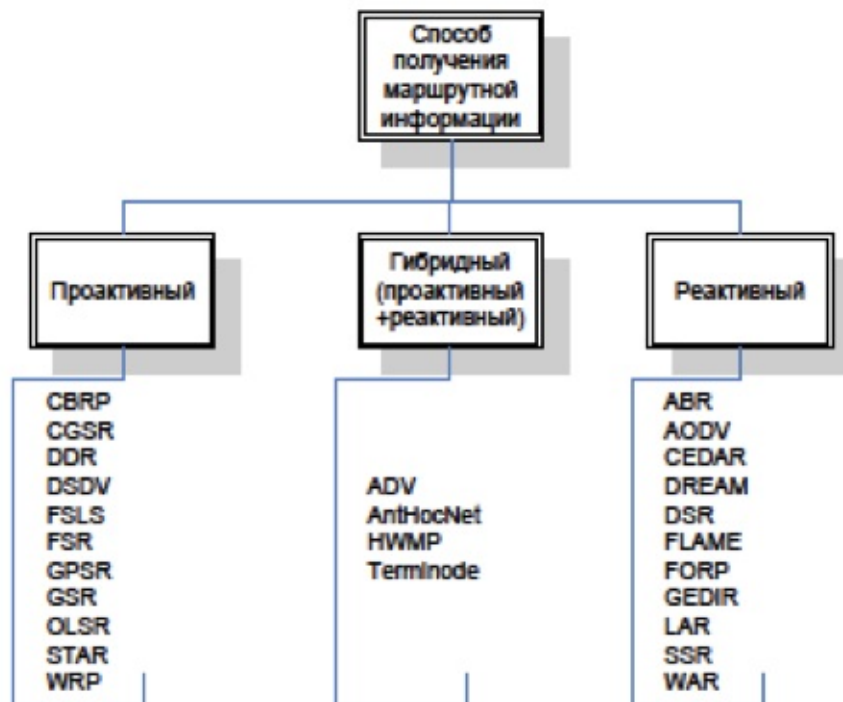
,где

$$\pi_0 = \frac{\langle T_o \rangle}{\langle T_o \rangle + \langle T_c \rangle}$$

Источники

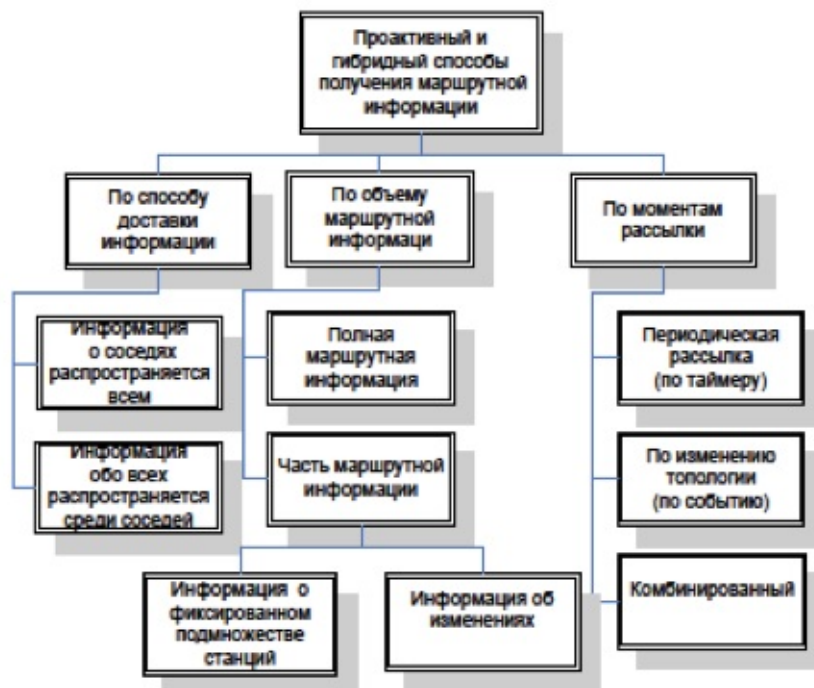
1. Khorov E. et al. Analytical study of neighborhood discovery and link management in OLSR //2012 IFIP Wireless Days. – IEEE, 2012. – С. 1-6.

Способ получения сетевой (маршрутной) информации



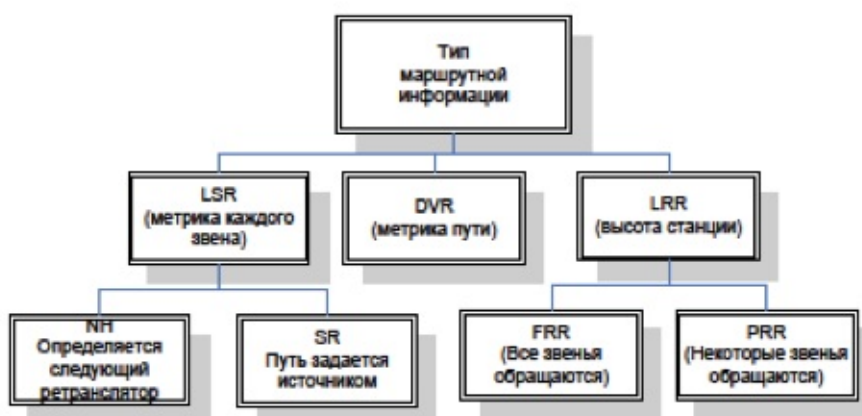
- Проактивный
 - информация о маршрутах посылается периодически (не зависит не от чего, кроме протокола)
 - например OLSR, DSDV
- Гибридный (проактивный + реактивный):
 - можно ограничивать области применения реактивного и проактивного подходов для разных маршрутов
 - HWMP
- Реактивный:
 - если что-то нужно, то информация рассылается, если нет — то нет. То есть сетевая информация ищется строго по запросу.
 - Например ADV.

Классификация проактивных протоколов



- выделяется подход с прореживанием broadcast сообщений
 - Каждую секунду шлем bc с маршрутами с TTL=1
 - Каждую 2 секунды шлем bc с маршрутами с TTL=2
 - Каждую 4 секунды шлем bc с маршрутами с TTL=4
 - ...
 - 256 (или диаметр сети)
 - такой подход сильно уменьшает оверхед
- FishEye, FSLs используют технику выше + HSLs

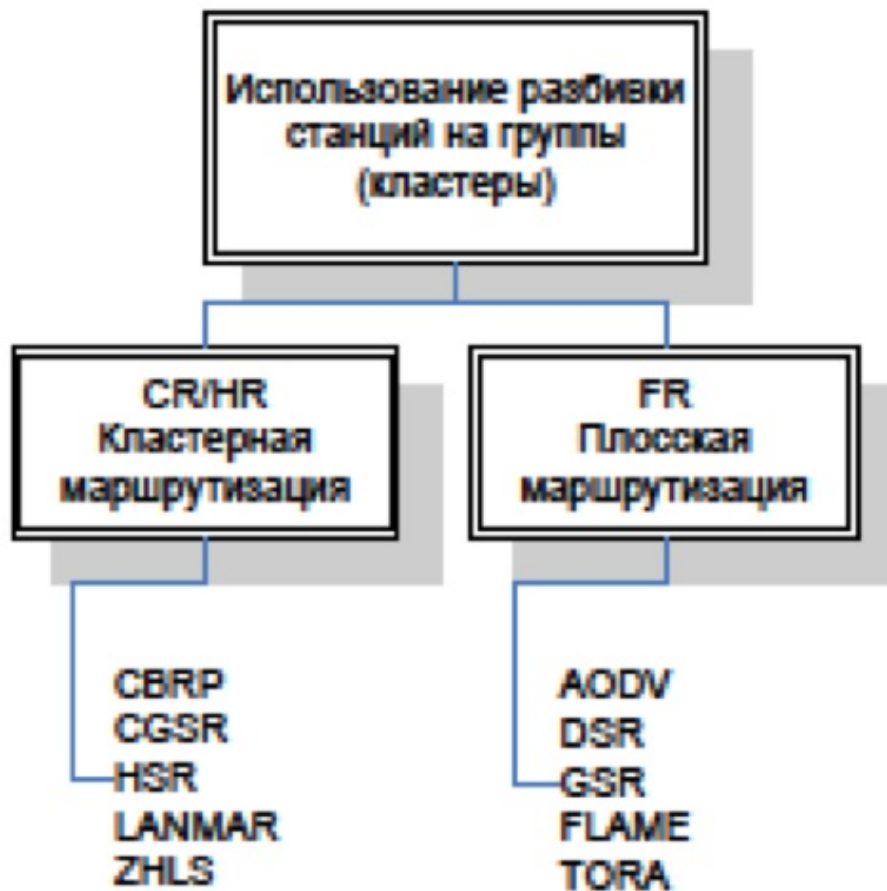
Тип маршрутной информации



- LS=Link State — хранит информацию о подмножестве соединений (графе) на котором можно в любой момент запустить поиск кратчайшего пути (чаще проактивные)
 - Next-Hop Routing — могут содержаться кольца
 - Source Routing — (источник строит маршрут) плохо для динамичных сред
- DV=Distance Vector — маршруты строятся распределенно (чаще реактивные)

- Алгоритм Дейкстры работает распределено (?)
- счетчик до бесконечности
- Еще один пункт, но это неважно

Возможность кластеризации



IEEE 802.11s WiFi Mesh

WiFi Mesh интегрировано в канальный уровень

- нужно чтобы использовать эксклюзивную информацию с канального уровня

MBSS

Новая архитектура в дополнение к инфраструктурной и AdHoc сетям.

Mesh BSS — сеть из равноправных узлов, при котором возможна передача через произвольные (?) промежуточные узлы.

Итого:

- В отличие от BSS можно передавать даже если станции не слышат друг друга
- нет точек доступа, но могут быть шлюзы
- Можно использовать MBSS, чтобы объединить несколько BSS в одну ESS

Особенность адресации кадров

Сколько нужно адресов? От 4 до 6.

1. RA — адрес следующего хопа (следующей меш станции)
2. TA — Адрес передающей меш станции
3. DA (dst) — Адрес назначения меш станции (итоговый)
4. SA (src) — исходная меш станция
5. DEA — Исходный получатель (нужен, если например адрес внешний)
6. SEA — Исходный отправитель

Биконы

- Mesh ID вместо SSID
- Генерирует и рассылает каждая станция
- Биконы **генерируются** периодически (В BSS генерируются одной, в AdHoc все пытаются послать Beacon, но посылает в итоге одна)

Beacons

Data Frames, Control Frames, Management Frames — кадры Wi-Fi

Control: ACK, RTS, CTS, CF-Pol (нужно послать очень-очень быстро, обрабатываются аппаратно, генерируются строго в определенный момент времени)

Management: меняют режим, сообщают дополнительную служебную информацию (генерируются программно, могут посылаться не сразу) .

Так вот, биконы это Management frame.

- Mesh ID вместо SSID

Beacon Interval — когда системные часы кратны Beacon Interval — Beacon отправляются, 100 Time Units,

1 Time Unit = 1024 microsecs

С помощью beacon устройства обнаруживают друг друга, узнают о параметрах сети, распространяется служебная информация.

Синхронизация происходит с помощью биконов: поле TSF — в нем указывается значение системных часов в микросекундах по модулю 2^{64} .

В Adhoc сетях часы обновляются если присланное время больше чем свое.

Beacon кладется в очередь...

Синхронизация

В mesh сетях синхронизация проходит также как в adhoc, только вместо коррекции времени запоминается offset для каждой станции.

Когда другая станция говорит о событии, которое произойдет в момент T (по другим часам), по собственным оно произойдет во время $T - offset$.

Избегание конфликтов биконов

(Mesh beacon collision avoidance (MBCA) — 14.13.4 в стандарте)

В основном направлено на уменьшение коллизий со скрытыми станциями (что часто для mesh сетей).

Глобально состоит из:

- Beacon timing advertisements
- TBTT selection
- TBTT adjustment

Каждая станция отслеживает TBTT (Target Beacon Transmission Time) своих соседей

- Как узнать когда именно TBTT станции?

$$T_{TBTT} = T_r - (T_t \bmod (T_{BeaconInterval} \times 1024))$$

где

T_r — время получения фрейма в терминах TSF

T_t — значение поля Timestamp время в полученном фрейме

$T_{BeaconInterval}$ — значение поля Beacon Interval в полученном фрейме

Рассылается информация (соседям) обо всех ТВТТ в единицах измерения 32 us, бекон-период в 1024us

=> с помощью беконов узнаем о соседях

Полученная информация позволяет настраивать ТВТТ на станциях так, чтобы минимизировать коллизии, алгоритм подробно не рассматривался.

Управление соединениями

Mesh Peering Management

Открытие соединений

- ->Peerig Open frame
- <- Peering Confirm frame
- <-Peering Open frame
- ->Peering Confirm frame

Закрытие

- ->Peering Close Frame
- <-Peering Close Frame

Все указанные кадры являются управляющими кадрами Management Action Frames.

(На каждый из пакетов еще посылаются аки)

Также дополнительно возможно секьюрное соединение (+обмен ключами)

После установки соединение будет считаться симметричным (отличие от OLSR)

Маршрутизация

- HWMP (Hybrid Wireless Mesh Protocol) — гибридный (проактивный + реактивный)
 - ранее рассматривался RA-OLSR
 - Является развитием AODV
 - Два режима работы
 - реактивный
 - Проактивный (...)
 - Могут работать вместе

Метрики маршрутизации

- HopCount
- ETX = Expected transmission count
 - Каждому линку приписывается переменная — среднее число попыток до успешной отправки
- ETT = Expected Transmission time
 - учитываются выбираемые СКК (сигнально кодовые конструкции)
- Airtime Link Metric
 - Используется в 802.11
 - Более точная ETT, учитывает сколько времени занят канал
 - $c_a = [O + \frac{B_t}{r}]_{1-e_f}^{-1}$
 - B_t — длина тестового пакета в битах (8192)
 - O — overhead не зависящий от скорости (преамбула + SIFS + DIFS)
 - r — data rate (Mb/s) represents the data rate at which the mesh STA would transmit a frame of standard size B_t based on current conditions, and its estimation is dependent on local implementation of rate adaptation
 - e_f — is the probability that when a frame of standard size B_t is transmitted at the current transmission bit rate r , the frame is corrupted due to transmission error; its estimation is a local implementation choice. Frame failures due to

exceeding Mesh TTL should not be included in this estimate as they are not correlated with link performance.

- $1/(1 - e_f)$ — среднее число попыток передачи
- Еще можно добавить backoff + время нахождения в очереди
- Другие метрики

Требования к метрикам

Дейкстра и Беллман-Форд требуют, чтобы метрика обладала свойствами:

Изотоничность: метрика будет изотоничная если при добавлении к каким то маршрутам звена или маршрута их качество по отношению к друг другу не меняется

ДЗ

Привести пример метрик которые обладают / не обладают таким свойством

Метод доступа МССА

- MCF controlled channel access
- Аналог НССА

MCF controlled channel access (MCCA) is an optional access method that allows mesh STAs to access the WM at selected times with lower contention than would otherwise be possible. This standard does not require all mesh STAs to use MCCA. MCCA might be used by a subset of mesh STAs in an MBSS.

Резервирование

Обычный DCF плохо работает в Mesh сетях — в основном из за скрытых станции

Две станции договариваются о временных интервалах передачи и рассылают это время бродкастом.

Все эти резервирование происходят периодически (метод периодических резервирований (много где используются)).

Три числа чтобы описать такие периоды

- длительность
- период
- смещения

При ненадежной передаче (вероятность успешной передачи $p < 1$) резервирование устанавливаются чаще чем период поступления пакетов.

Аналитическая модель

1. <https://www.researchgate.net/publication/220654266> (по сути то, что ниже)
2. https://www.researchgate.net/profile/A_Lyakhov/publication/254037355_Mathematical_model_of_MCCA-based_streaming_process_in_mesh_networks_in_the_presence_of_noise/links/0deec5209e9cb33078000000/Mathematical-model-of-MCCA-based-streaming-process-in-mesh-networks-in-the-presence-of-noise.pdf (финальная версия статьи с усложненной моделью)

Постановка задачи

Вход:

- максимальная доля потерянных пакетов L_{QoS}
- максимальное время доставки пакетов D_{QoS}

Найти:

с каким интервалом следует устанавливать периодические интервалы отправки? (по сути посчитать PLR, исходя из него можно подобрать оптимальный выбор таких интервалов)

Всю ось разбиваем на временные слоты τ — НОД (интервал между пакетами, период резервирования) — не обязательно целый

Интервал между пакетами = $t_p \tau$

Период резервирования = $t_r \tau$

Шкала времени разбивается так, чтобы период резервирования совпадал с началом слота. Моменты поступления будут сдвинуты относительно начала слота на ξ .

Описание процесса

- марковская цепь с единицей времени: период резервирования
- Состояние: $h(t)$:
 - $h(t) \geq 0$ — возраст (в слотах) самого старшего пакета в очереди
 - $h(t) < 0$ — время прибытия (в слотах) следующего пакета
 - Минимальное значение $h(t) = -t_p + t_r$
 - Максимальное значение $h(t) \leq d = \lfloor \frac{D-\xi}{\tau} \rfloor$, $D = D_{QOS} - TransmissionTime$

Расчеты:

$h \geq 0$:

1. $h \rightarrow h - t_p + t_r$ — Когда самый старший пакет или успешно передается или отвергается ($h + t_r > d$)
2. $h \rightarrow h + t_r$ — Ошибка передачи и $h + t_r \leq d$

$h < 0$

1. $h \rightarrow h + t_r$

Итого, т.к. пакет сбрасывается с вероятностью $(1 - p)$ из любого состояния, где $h + t_r > d$, при этом кол-во принятых пакетов = t_r / t_p

$$PLR = \frac{(1-p) \sum_{i=d-t_r+1}^d \pi_i}{t_r / t_p}$$

Результаты

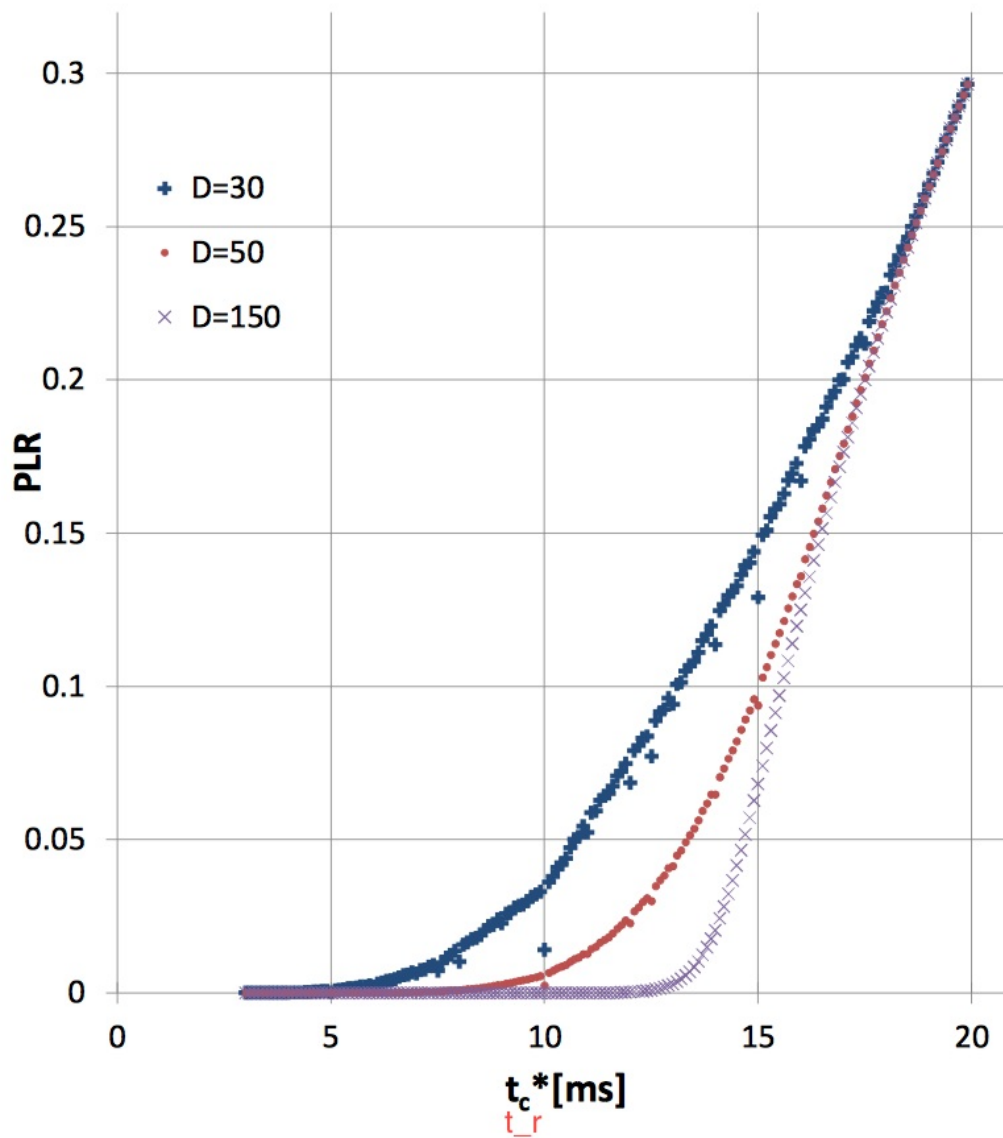


Figure 2: PLR versus t_c^* : $t_\lambda^* = 20$ ms, $q = 0.3$

Забавный или не очень факт в том, что $\text{PLR}(t_r)$ не является монотонной в любой точке. Подробнее про это можно почитать в статье.

Высокоскоростной WiFi

- Сначала скорость WiFi 1-2 Мб/с
- потом 802.11a и b до 11 и 54 мб/с
- OFDMA
- и тд до 30 Гб/с

Пути увеличения скорости

- Увеличение ширины полосы
- MIMO
 - две антенны на передатчике и 2 на приемнике
 - 4 пути
 - Сумма двух сигналов $X_1(t - \delta t_1) + X_2(t - \delta t_2) \rightarrow h_1 X_1(t) + h_2 X_2(t)$
 - имеем систему из двух уравнений
 - знаем затухание (коэффициенты)
 - решаем систему, получаем искомое
 - минус в том, что знаем все это с какой-то точностью (h меняются)
 - пути не всегда независимы
- Single User MIMO (802.11 ac, ax)
 - Пути (стримы) принадлежат одному устройству (на каждом устройстве больше одной антенны, в Multi User мб одна)
 - Например, 8 антенн => 4 устройства по 2 стрима
- Новые СКК
 - QAM 16, QAM 64, QAM 256, QAM 1024
 - QAM 4096 (думают делать или нет)
 - Номер соответствует количеству значений одного символа (на одной несущей)
 - 64 => 6 бит (можно закодировать)
 - 1024 => 10 бит
 - Упирается в формулу Шеннона (?)
 - Каждый раз нужен в 2 раза меньший SNR

Номинальная скорость передачи данных — скорость передачи payload для пакетов

Пакет еще содержит преамбулу (передается на некоторой фиксированной СКК), состоит из:

- training fields ~ 16мкс (чтобы обозначить факт передачу)
- OFDM символ ~ 4мкс (задает параметры: длительность пакета и алгоритм СКК для payload)

При увеличении номинальной скорости, сильно играют роль накладные расходы на преамбулу и АСК-и.

Пути решения:

- агрегирование
- блочное подтверждение

Агрегирование

Объяснение отходит от стандарта

TXOP и OFDMA

ТХОР

Неясно при чем здесь ТХОР

Избавляемся от слотов отсрочки.

К этому можно добавить блочное подтверждение.

A-MPDU и A-MSDU

По сути и есть агрегирование, появилось в 802.11n

A-MPDU — protocol

A-MSDU — service

A-MSDU

есть пакеты с равными src и dst mac, их можно склеить.

Вместо data в payload помещается AMSDU, состоящий из сабфреймов, каждый состоит из:

- 2 адреса (могут быть уникальными для каждого пакета (случаи 3-х и 4-х адресов))
- Длина
- MSDU (оригинальная data)
- Padding

Общая контрольная сумма на все сабфреймы

Минусы:

- одна контрольная сумма

A-MPDU

Все похоже, но еще есть delimiter'ы

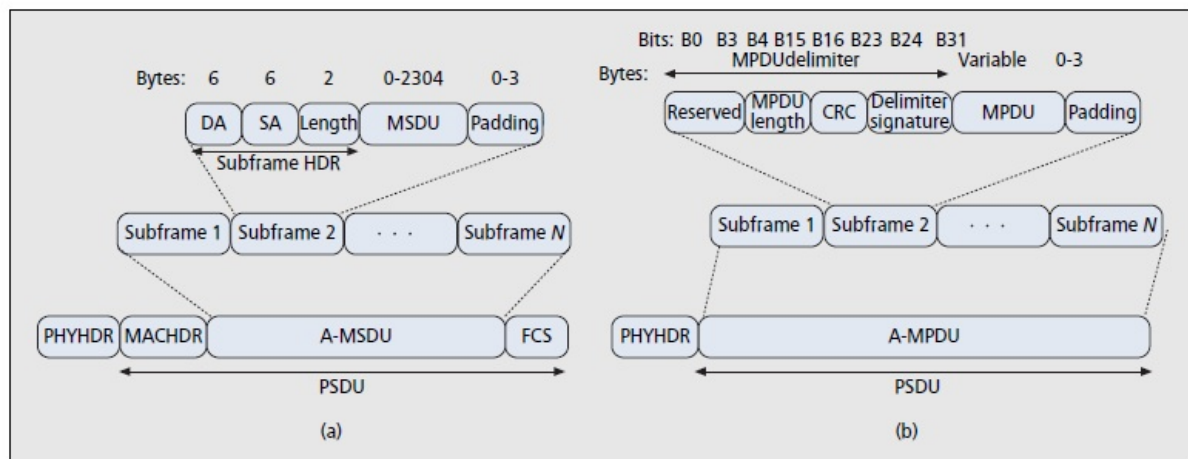
Состав сабфрейма:

- delimiter
 - резерв
 - длина
 - контрольная сумма (для поля length)
 - подпись (ascii код буквы "N" (78))
 - Нужно, чтобы восстановиться при нарушении контрольной суммы length
- MPDU (кадр канального уровня со всеми заголовками)
- Padding (чтобы выравнивать длины фреймов для железок)

MPDU длиннее чем MSDU

Зато есть контрольная сумма для каждого кадра.

Btw, использование A-MPDU, предполагает наличие block ack.



■ Figure 2. One-level frame aggregation: a) A-MSDU; b) A-MPDU.

OFDMA

Посылаем одну преамбулу физического уровня

Дальнейшая передача продолжается независимо в каждом из resource юнитов (поднесущие). Таким образом преамбула передается один раз для всех уровней. Таким образом увеличиваем SNR, идеально для uplink-a.

Передающая станция сообщает получателю ресурс юнит, который нужно слушать.

Блочное подтверждение

Устройства договариваются, что подтвердить будут пачками

Появилось в 802.11e. (будет в тесте)

Вопрос: В каком стандарте появился BlockAck

Ответ: Тогда, когда появился txor (видимо в 802.11e)

Посылаем много пакетов.

Потом block-ack request, указывая стартовый номер пакета

В ответ получаем битовую маску.

Нужно договориться (AddBARequest, AddBAResponse, DelBA — Management кадры, подтверждаются аком):

- Размер окна (в штуках пакетов), ограничивает память на сохранение пакетов
- Таймаут (если не использовали block-ack в течение этого времени, то забываем о соглашениях)
- Начальный seqNumber

Пакеты нумеруются по mod 4096.

Реальное окно не более чем 64 пакета. (зафиксировано в стандарте).

Останавливаем передачу не по retry limit, а по packet lifetime, иначе сложно отслеживать.

Неясно, что за packet lifetime

Размер окна также отслеживается.

- Если получили старый SN, то пакет отбрасывается
- если в пределах окна, то принимаем
- если слишком новый, то сдвигаем окно, а те пакеты, которые выпали мы передаем наверх (тк все равно все)

остальные мы отбросим).

- Иначе, пакеты не передаются наверх, пока не будут закрыты все дырки.
- Когда приходит Block ACK Request, также сдвигаем окно до указанного в запросе номера.

Есть два режима отправки блочных подтверждений (Immediate, Delayed).

Immediate

Сейчас все стараются использовать его.

Block Ack Response следует за Block Ack Request через SIFS

Delayed

BlockAckRequest → Ack → ... ChannelAccess ... → BlockAckResponse

Кроме Immediate и delayed существует:

- Постоянное отслеживание состояний
- Временное (храним не всю битовую маску, отвечаем наугад, хз зачем)

BTW

Sequence control состоит из:

- Sequence number
- Fragment number (для фрагментации)

802.11 ax

Заканчивается разработка, поступит в продажу (массовую) через год (2020 год).

Эволюция стандартов:

1. 802.11a и g используют OFDM (а для 5ГГц, g для 2.4 ГГц, в остальных стандарты похожи)
2. 802.11n — MIMO + расширение полосы [5/2.4]
3. 802.11ac — higher order channel bonding & MIMO + MU MIMO [только 5ГГц, т.к. для 2.4 нет полосы 60МГц]
4. 802.11 ax — OFDMA + UL MU MIMO

802.11 ax не для увеличения номинальной скорости передачи пакетов, он для увеличения реальной скорости.

Например, 11ac может в теории передавать 7Gbps, но условия почти нереальные:

- насыщенная передача
- нет интерференции
- большая полоса
- по 8 антенн на отправителе

В жизни все не так.

Основные цели

High Efficiency

- Улучшение перфоманса в плотных сетях
- улучших Quality of Experience

Главные плюшки

- Modified PHY
- Uplink Multi-user Multiple Input Multiple Output (UL MU MIMO), (802.11ac supports multiuser MIMO, but only in downlink mode. In contrast, 802.11ax adds uplink capability, so multiple users can upload video simultaneously)
- OFDMA (киллер фиша) — появилось только тут
- Channel access modifications
- Features for enabling Spatial Reuse (SR) (две точки доступа работают рядом, но при этом разделяют канал без негативных последствий)
- Enhanced Power Efficiency, например TWT.

Номинальная скорость увеличилась всего на 37%

Зато реальная конкретно увеличилась

То, что появился full-duplex — это миф.

Но появился MU Transmission в виде OFDMA.

В даунлинке просто посылаем всем одну преамбулу.

В аплинке пользователи отправляет одну и ту же преамбулу, синхронизируются базовой станцией.

PHY Features

- Унаследовали с 11ac
- Changed numerology: 1024 QAM
- Растянули OFDM символ 3.2 us (+ GI of 0.8us) -> 12.8 us (+GI of 3.2, 1.6 or 0.8 us) (omg, что это вообще значит,

спасите).

- GI это Guard Interval
- Нужно, чтобы передавать на большие расстояния (?)
- и уменьшать overhead от GI

To increase the number of tones per band and, thus, to provide better granularity for OFDMA, the OFDM symbols used for PHY payload are 4 times longer, i.e. 12.8 μ s instead of 3.2 μ s [5]. Long OFDM symbols also improve robustness especially for the UL MU transmission in outdoor scenarios prone to large timing jitter across the users.

Преамбула состоит из двух частей:

- Legacy
- HE (новая для ax)
 - HE-SIG-A
 - HE-SIG-B

Legacy

Нужна для обратной совместимости (или ее воркэраунда с L-SIG Protection). Также содержит training sequences needed for the receiver to synchronize on the received signal.

HE

Начинается с повторения LSIG, это важно для плотных сетей. Затем идут SIG-A, SIG-B.

HE-SIG-A

Длинной 2 OFDM символа. Дублируется каждые 20 МГц, содержит информацию важную для принятия пакета: СКК, число spatial стримов, bw. Также в нее записали часть информации с MAC, например TXOP duration (это нужно для экономии энергии). Тк важно принять эту часть надежно возможно хитрое дублирование данного заголовка / кадра / чего? По сути содержит всю необходимую информацию для SU передачи.

HE-SIG-B

Используется, когда происходит MU передача (OFDMA или MU MIMO). Содержит информацию для всех станций, затем идет информация для конкретной станции. Для увеличения надежности дублируется каждые нечетные 20МГц.

MultiUser

- Появился UL MU MIMO, это сложнее чем DL.
- Появился OFDMA

To simplify resource management, in 802.11ax network each transmission occupies a particular set of OFDM tones called RU. A RU can contain 26, 52, 106, 242, 484 or 996 tones (including service tones). The whole 20 MHz band, 40 MHz band, 80 MHz band and 160 MHz band correspond to a 242-tone RU, two 242-tone RUs, two 484-tone RUs and two 996-tone RUs, respectively. Each wide RU can be split into two approximately twice-narrower RUs. In turn, each of them can be split again, separately from another one. The only exception is that in a 20 MHz band a 242-tone RU can be replaced by 2 106-tone RUs and one 26-tone RU.

MU MIMO

Может использоваться совместно с OFDMA, но только в RU ≥ 106 .

OFDMA

В случае даунлинка все +- просто, информация о распределении RU хранится в пакете. Посылается общая преамбула, а затем разные payload.

С аплинком все сложнее:

- Станции сообщают о размере своего буфера
- Учитывая эту информацию точка доступа выделяет RUs и рассылает эту информацию с помощью нового Trigger frame, станции должны произвести отправку сразу при получении Trigger frame. (Но есть нюанс — In contrast to legacy STAs, which ignore their Network Allocation Vectors (NAVs), if any, when transmit an immediate response, a .11ax STA shall cancel its UL transmission if its NAV is nonzero, unless it transmits an ACK or BlockAck, or the NAV was set by the AP, which is the originator of the Trigger frame. Apart from that, the STA shall not transmit if its transmission exceeds the UL transmission duration indicated in the Trigger frame.)
- Круто, что все это можно делать добавляя информацию из Trigger Frame в DL MU фрейм — и, например моментально доставлять ACK пакеты.

Доступ к каналу

Помимо описанного выше MU доступа существует Aloha-подобный механизм доступа к каналу. Предполагается, что базовая станция не знает об очередях. Для этого используется OFDMA Back-off процедура. Сначала взводится рандомный бэкофф (OFDMA contention window). При получении Trigger frame он уменьшается на количество RU в этом фрейме. Когда счетчик достигает нуля — выбирается рандомный RU и производится передача.

Также предполагается (для отказа от всяких RTS/CTS и ненужных IFS) использовать механизм roster (список). По сути это реализация идеи критического доступа с помощью токена. Так, составляется список станций которые могут передавать в данный временной промежуток. В этом списке в определенной последовательности станции передают, при этом если станции нечего передать — токен передается следующей станции в списке. Это работает лучше чем MCCA, так как канал выделяется для группы станций.

Spatial Reuse

Данное предложение предлагает различать кадры из своего BSS и чужого BSS. Для этого каждая станция случайно выбирает себе 8-ми битный номер — BSS color. Данное значение добавляется в HE-SIG-A. Такой подход позволяет эффективней различать занятость канала на физическом уровне и канальном уровне (не убивать свой NAV по CF-END с другого BSS).

802.11n

BlockACK

802.11e

Некоторые дополнения к Лекции 5.

Блочное подтверждение улучшает использование ресурсов

2 метода подтверждения:

Immediate

Delayed

BlockACK появился как механизм, который может быть реализован в драйвере, а не в железе.

Становится очевидным почему может быть Delayed — можно медленно обработать в драйвере.

"Больше задержка, но зато хорошо"

Картинки работы см на слайде

Мы отправили BlockACKReq.

Станция, которая отправила BlockACKReq — знает дошло ли или нет (по BlockACK) — если нет — повтор.

В immediate не нужен ACK на BlockACK, но в delayed нужен .

BlockACKReq — имеет 2 интересных поля: BARControl, в нем указан TID (тип трафика)

BlockACKStartingSequenceControl — состоит из 2 подполей: номер пакета, номер фрагмента.

В ответ приходит BlockACKResponse.

BTW, помимо BlockAck в 802.11e появилось (задается в поле QosControl):

- NoAck
- NormalAck
- No explicit ACK (?)
- Сам blockAck

802.11n

в 11n было улучшено.

BlockACK был один — Basic, добавили сжатый (compressed) и multi-tid.

В BARControl ключевыми полями являются Multi-tid и compressed.

Multi-tid blockACKReq появилось дополнительное поле переменной длины, указываем для каждого TID может быть несколько SequenceControl.

Как я понял, MultiTID может просто подтверждать пакеты из разных TID.

В Compressed мы учитываем только MSDU, а не фрагменты.

Не путать агрегацию и фрагментацию, агрегация появилась в 802.11n, фрагментация +- deprecated, начиная с 802.11n (это неточно).

Compressed используется, если не используется фрагментация. При compressed размер bitmap 8 байт (64 MPDU), без — 128. (16 фрагментов * 64 MPDU)

ADDBA Request — starting sequence number, BlockACK timeout, BufferSize

ADDBA Response — buffer size, Block ACK timeout.

ADDBA:

- Dialog Token
- Block Ack Parameter Set
- Block Ack Timeout Value

Повторные попытки передач:

Делаем попытки передачи не по retry-limit, а по времени жизни MSDU.

Делим пространство номеров пополам, все пространство номеров больше выбранного — новые, остальные — старые.

Зачем?

См. пикчу со слайдов (с кругом)

Есть буфер мы передаем пакеты (MSDU) наверх до первой дырки.

Пакет передаем наверх по lifetime.

Окно сдвигается, если пришел кадр за пределами окна, либо если вручную это сделал BlockACK.

11e — поверхностно, 11n — подробнее.

BlockAck Modifications

Для более простых устройств:

Full state — always maintain the bitmap.

Partial-state — can reset the bitmap if frames from another transmitter are received.

MIMO

SU MIMO появился в 802.11n

802.11n поддерживает 4 x 4 : 4 SU MIMO.

Благодаря MIMO мы увеличиваем скорость передачи данных, можно увеличить payload (число бит передаваемых в единицу времени).

STBC — специальный тип кодирования, мы можем увеличить дальность передачи сигналов за счет того, что по нескольким антеннам идет один поток.

Когда у нас есть несколько антенн, мы можем увеличивать количество spatial-stream — страдает дальность, но этот подход выгоднее на более близких расстояниях.

Суть spatial-stream: физический пакет есть, вот он может передаваться быстрее.

По сути MIMO ускоряет текущую СКК. (Так это моделирует NS3).

Фактически на приемнике нужно решать некоторые линейные уравнения, для этого нужно произвести оценку канала.

Для использования MIMO кадр усложняется: нужно передавать специальные символы, которые позволяют "натренировать" канал + добавляются специальные параметры.

HT-SIG — новые CCK, используется ли MIMO, сколько spatial стримов.

Далее идут специальные тренировочные поля по количеству spatial стримов + дополнительные поля для ресивера.

В 11a поле signal содержало CCK и длину пакета.

Кадры являются обратно совместимыми.

L-SIG — указывается минимальная скорость, длина пакета (фейковая)

HT-GREENFIELD-PPDU — "зеленое" поле, значит, что legacy-станций нет, и мы можем использовать сжатый кадр.

max HT length — 64Kб.

Длина кадра ограничивается A-MSDU.

LSIG TXOP Protection

Тк обратной совместимости нет, становится актуальной задача донести информацию о передаче (NAV) до legacy станций. В 802.11n появилась возможность делать это на физическом уровне.

In 802.11g, there was only one protection mechanism defined. Before transmitting in the newer style, a device had the responsibility to transmit in a backward compatible way to make sure older stations correctly deferred access to the medium. The most common way of achieving this is for a device to transmit a CTS frame to itself, using an older modulation that can correctly be processed. Before transmitting a frame using the "new" style, a station sends a frame in the "old" style that tells receiving MACs to defer access to the medium. Although an older station is unable to detect the busy medium, it will still defer access based on the medium reservation in the CTS.

In addition to the MAC-layer protections, 802.11n adds a new PHY-layer protection mechanism. The PLCP contains information on the length of a transmission, and 802.11n sets up the physical-layer header so that it includes information on the length of transmissions.

L-SIG TXOP Protection — защита TXOP с помощью того значения, которое мы указывали в поле L-SIG.

В L-SIG — некорректные данные (фейковая длина пакета). Будем выставлять это значение в большее, чем нам нужно для передачи нашего кадра + передача в обратную сторону.

Та станция, которая услышит L-SIG будет считать, что канал занят и не будет передавать.

NAV, который выставляется на физическом уровне.

Для durationid нужно декодировать RTS/CTS, если какая-то из станций его не слышит, но получив из поля L-SIG это значение, значит, ей не нужно что-то декодировать и все норм.

802.11 ac

Channel bonding

В 11ac стало возможным использование полосы до 160МГц.

Всегда определяется primary 20МГц канал.

Primary и secondary.

Primary половина — там где лежит primary20, вторая половина secondary.

2.4 нельзя — не хватит ширины полосы.

backoff считаем в primary20 (слушаем только его), мы можем вести передачу и принимаем решение в котором канале вести передачу. Если secondary канал был свободен в течение PIFS можем его задействовать, если secondary40 был свободен в течение PIFS, то передаем в 80МГц, если secondary80 был свободен то передаем в 160 МГц.

Кадр RTS/CTS должен передаваться во всей используемой полосе, чтобы покрыть всю будущую передачу.

CTS — должен передаваться в legacy формате. (в 20 МГц)

Внимание вопрос: воспользовавшись стандартом ответьте, что нужно делать с RTS, BlockAck, BlockAckRequest.

В 11ac MU-MIMO — данные передаются одновременно нескольким станциям.

Мы одновременно передаем разные пакеты, которые передаются разным станциям.

Backoff считаем в primary, а выбор primary / secondary происходит в момент передачи:

- 20
- 40
- 80

802.11 ad ay

Wifi в миллиметровом диапазоне

Зачем:

- можно взять больше ширину полосы
- частота больше
- больше спектра — больше пропускная способность (следует из формулы Шеннона)

Нужно для многих сценариев, где нужны высокие скорости.

802.11 ad — аналог a, но в mm

802.11 ay — аналог ac, но в mm

802.11 ad:

- ширина полосы 2 ГГц
- отказались от OFDM (из-за большого количества поднесущих) — используется что-то другое
- более высокие скорости в СКК
- Номинальная скорость до 7 Гб/с

802.11 ay:

- ширина полосы 8 ГГц
- До 8 spatial стримов

Проблемы с диапазоном выданным под нелегальным использованием — в этом диапазоне передается максимум на 10 метров из-за поглощения кислорода (?)

Архитектура сети