

---

# Table of Contents

Introduction	1.1
WiFi	1.2
Лекция 1	1.2.1
Лекция 2	1.2.2
Решение ДЗ	1.2.2.1
Лекция 3	1.2.3
Лекция 4	1.2.4

## **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

```

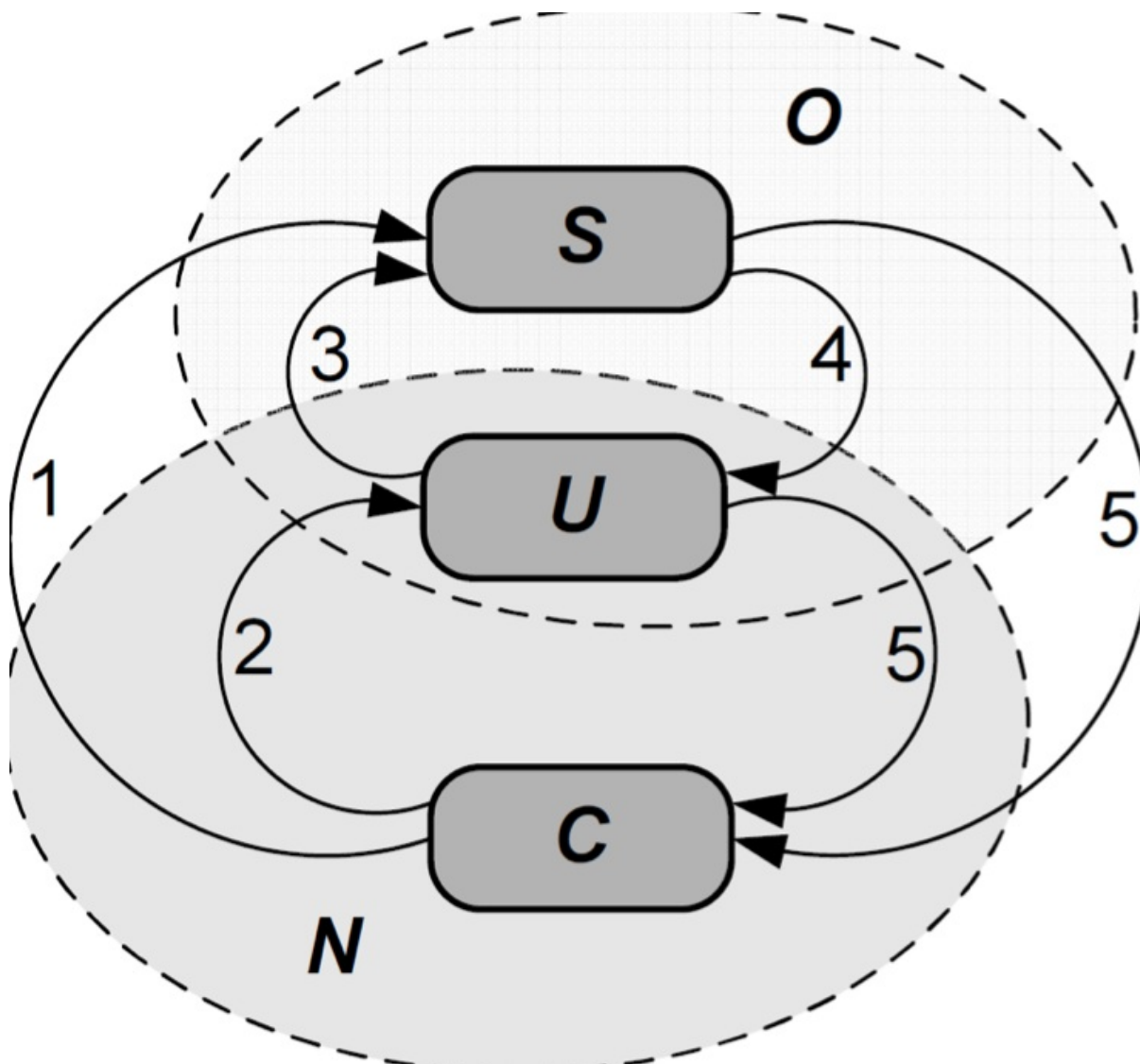
      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Reserved           |      Htime      | Willingness |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Link Code   |   Reserved   |   Link Message Size   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Neighbor Interface Address           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Neighbor Interface Address           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
:                                     :
:                                     :
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Link Code   |   Reserved   |   Link Message Size   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Neighbor Interface Address           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Neighbor Interface Address           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
:                                     :
:                                     :
(etc.)

```

- 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 достучаться 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).

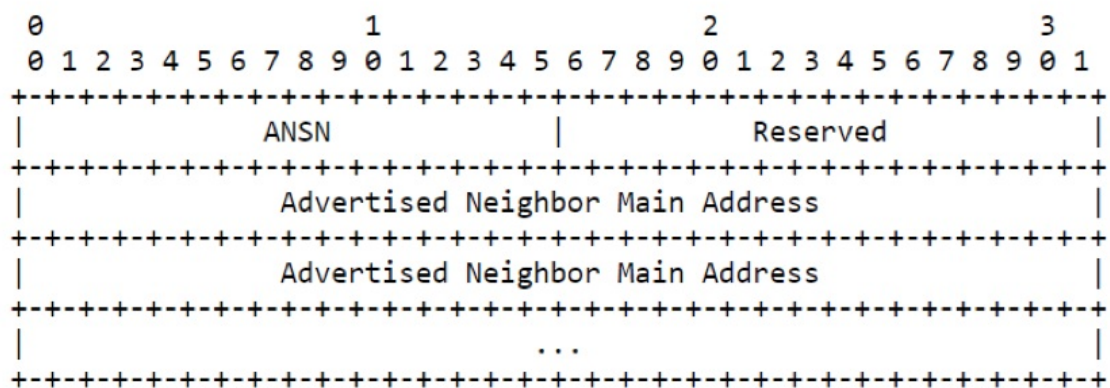
Each node maintains information about the set of neighbors that have selected it as MPR. This set is called the "Multipoint Relay Selector set" (MPR selector set) of a node. A node obtains this information from periodic HELLO messages received from the neighbors.

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.

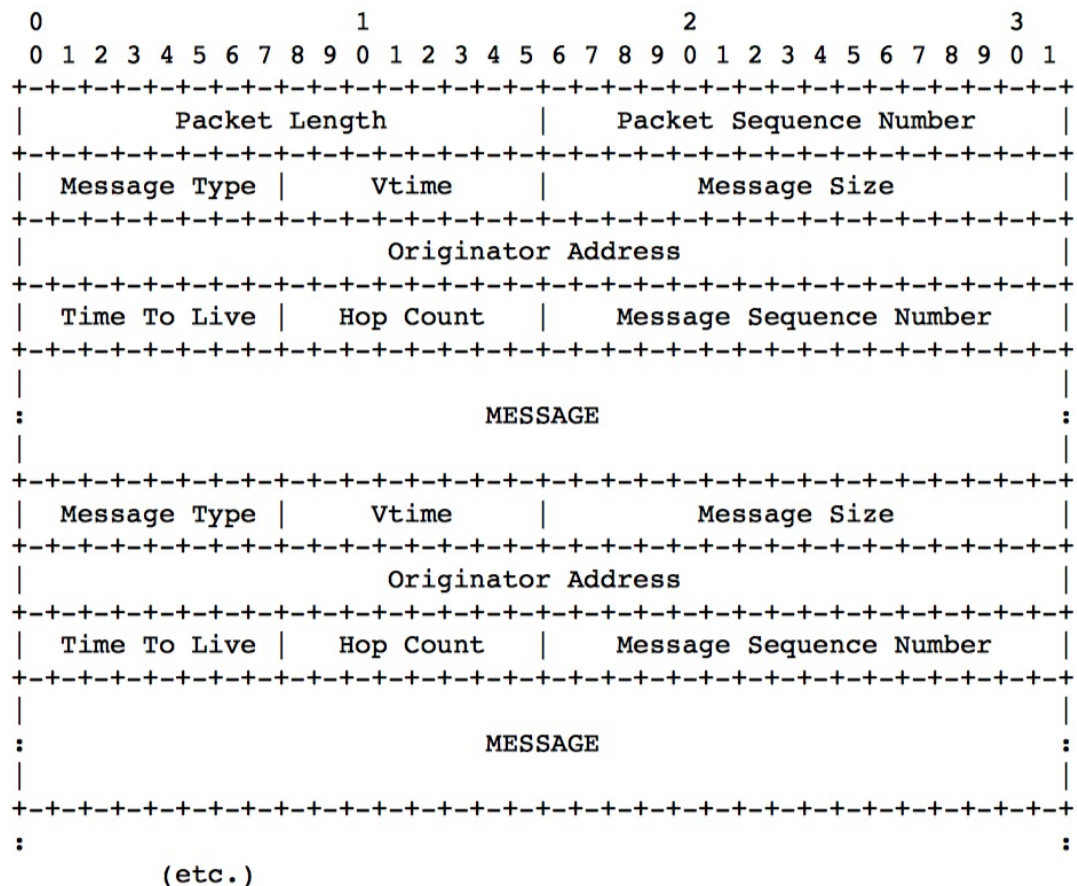
### Формат пакета

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:

$\pi_s$  — вероятность того, что соединение симметрично

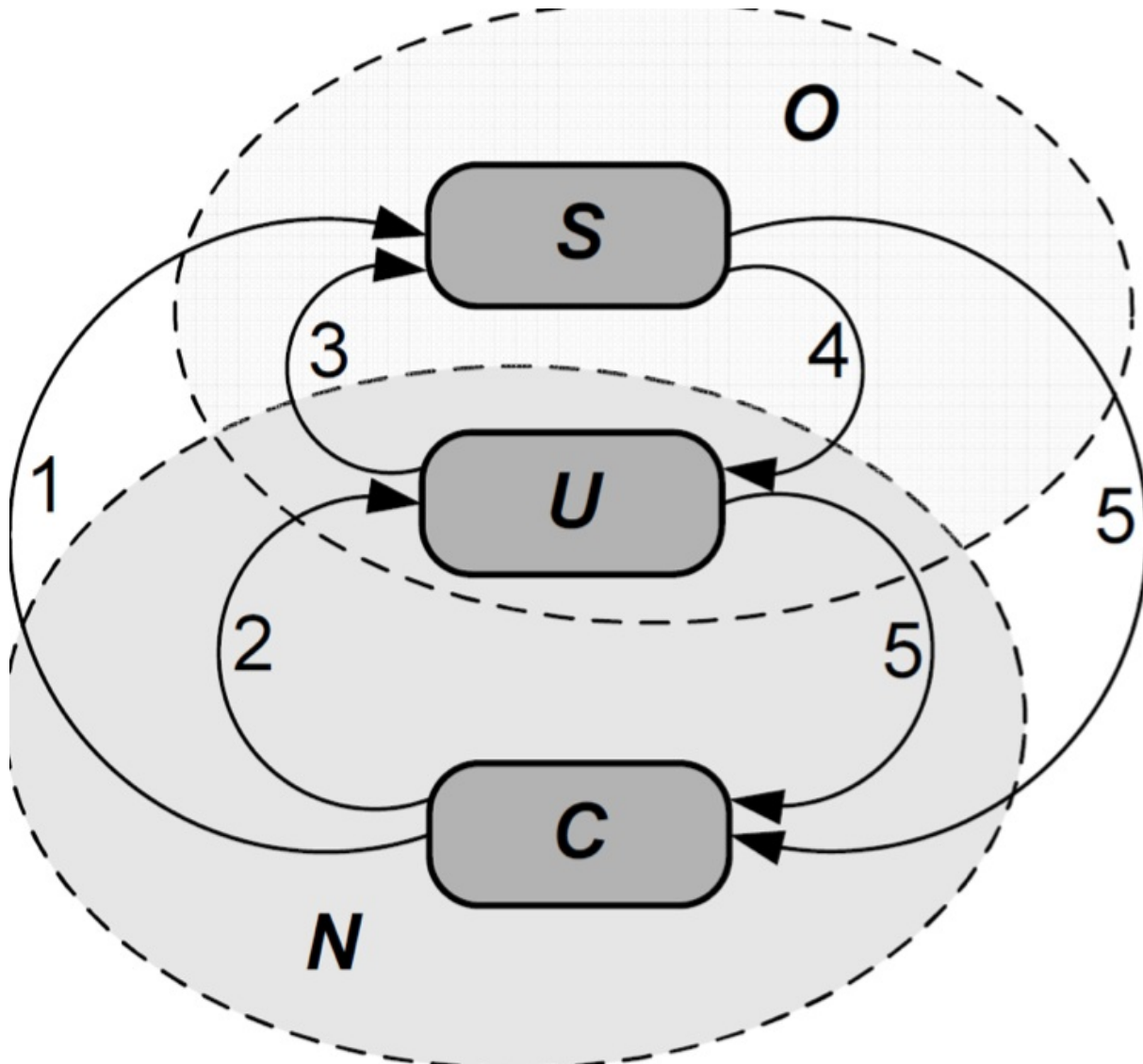
$T_S$  — среднее время когда соединение симметрично

$T_N$  — 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

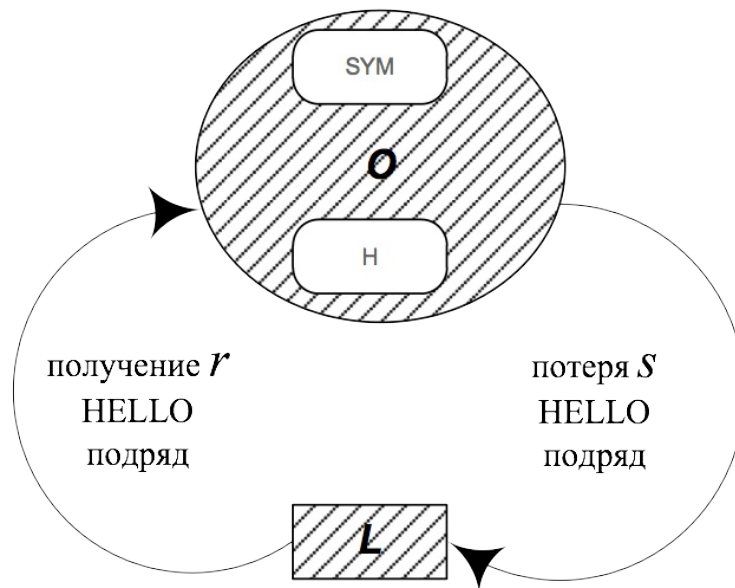
$$E(\sum_{i=1}^N X_i) = E(N)E(X)$$

**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

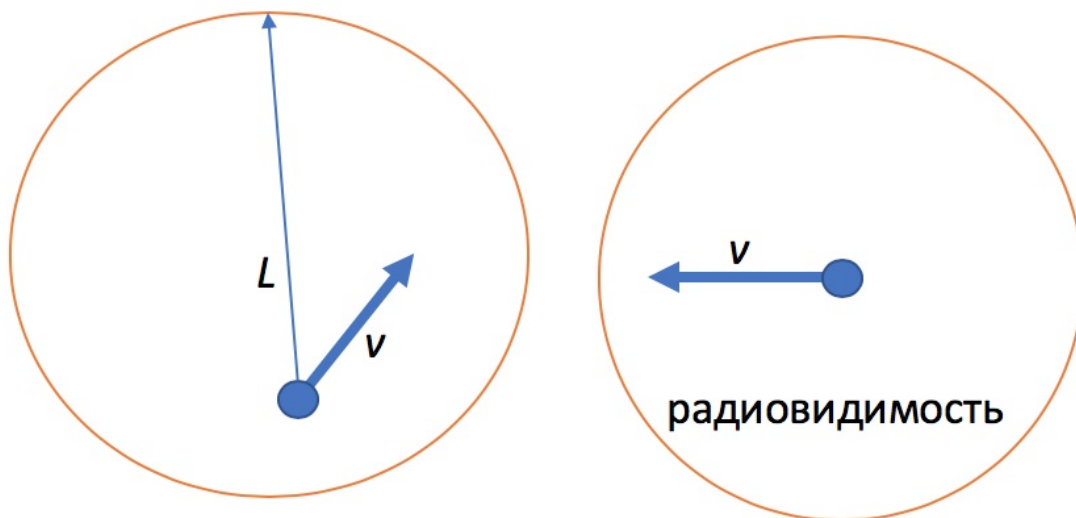
Найти  $T_N$  среднее время в нахождении  $C + U$

Итого: 4 ДЗ

ДЗ 5\*

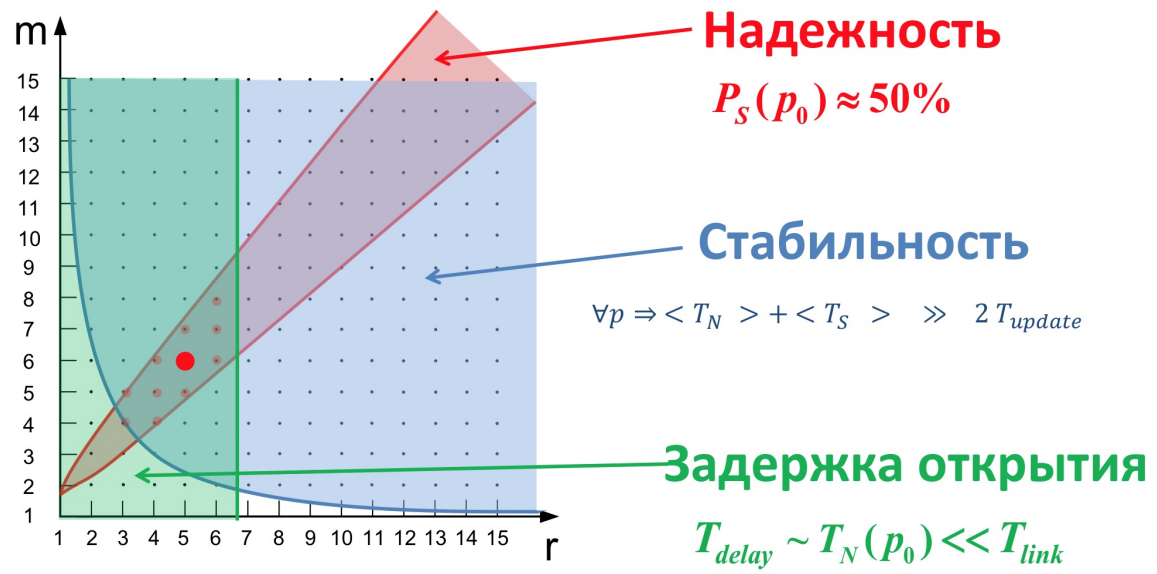
- 5-е дз со \* (Среднее время жизни физического соединения)

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



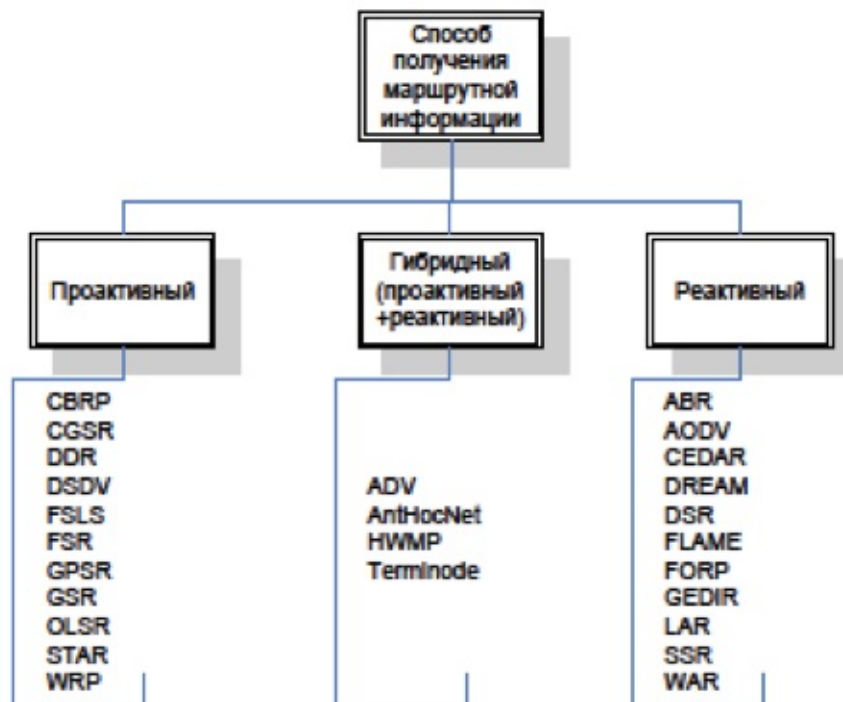
### Настройка параметров

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



Что такое здесь  $m$  — неясно (видимо  $s$ )

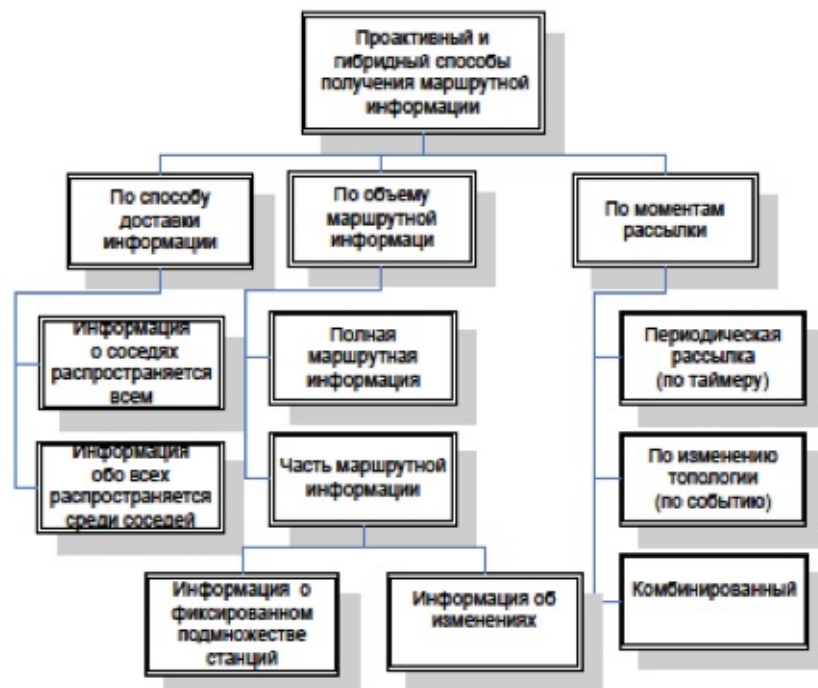
## Способ получения сетевой (маршрутной) информации



- 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.

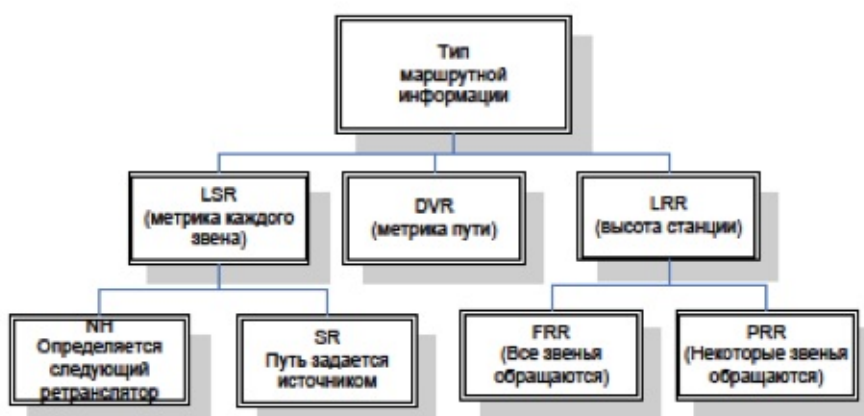
### Классификация проактивных протоколов





- 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

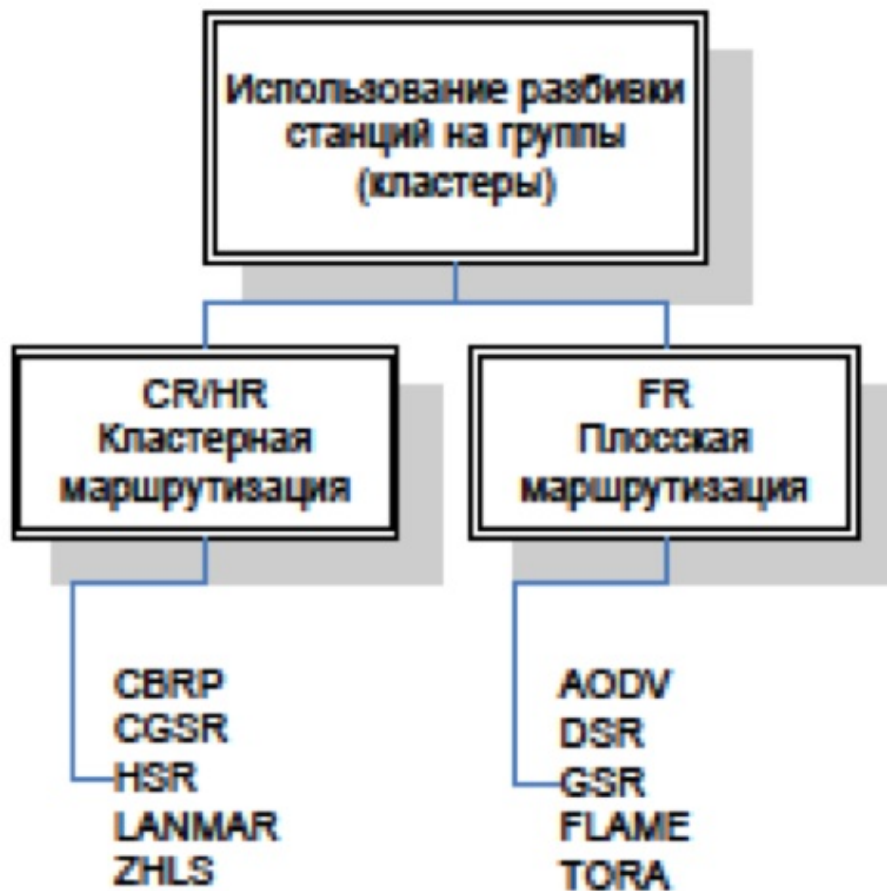
#### Тип маршрутной информации



- 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
  - $c_a = [O + \frac{B_t}{r}]_{1-ef}^{-1}$
  - $B_t$  — 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
  - $ef$  — 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](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  $\tau$  — 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  $\xi$ .

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  $h + t_r \leq d$

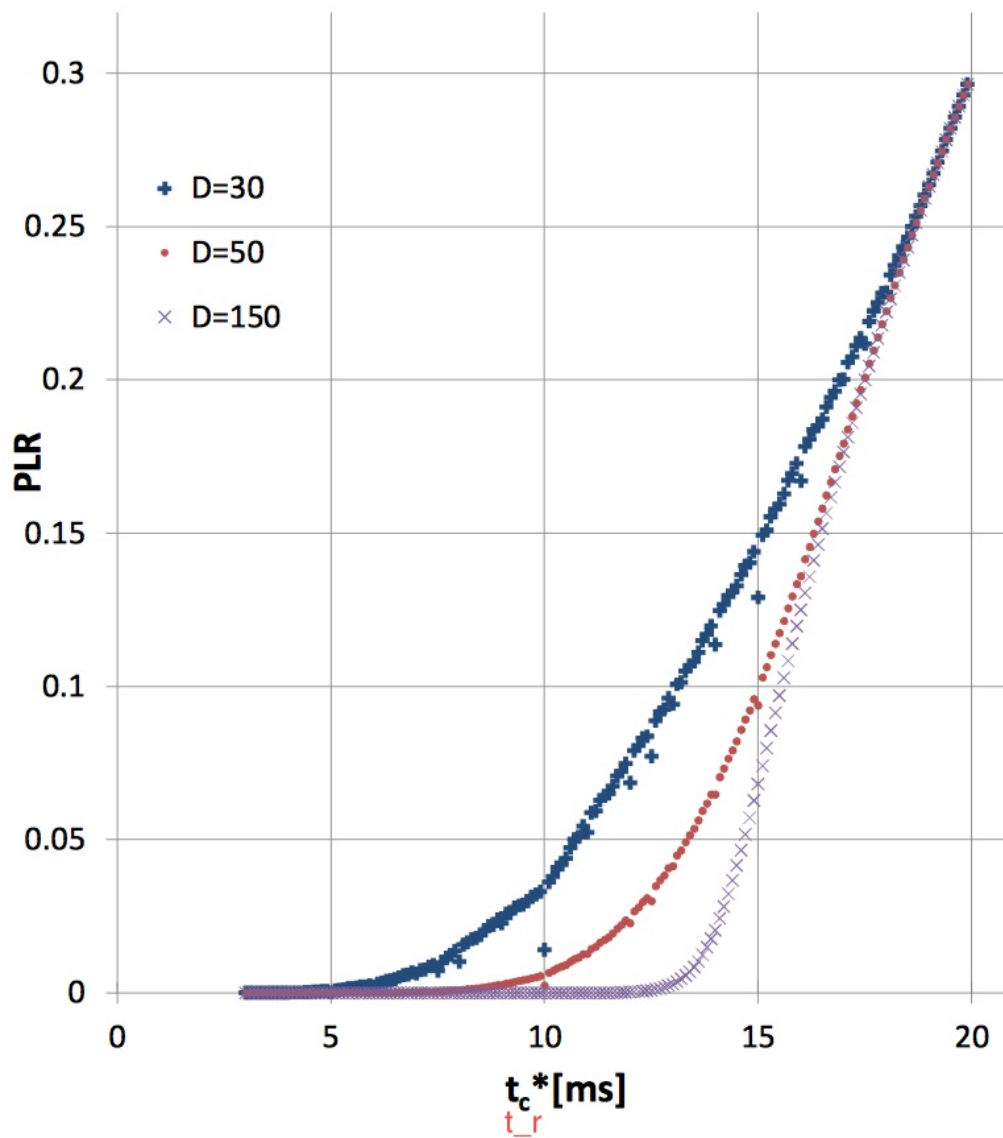
$h < 0$

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.