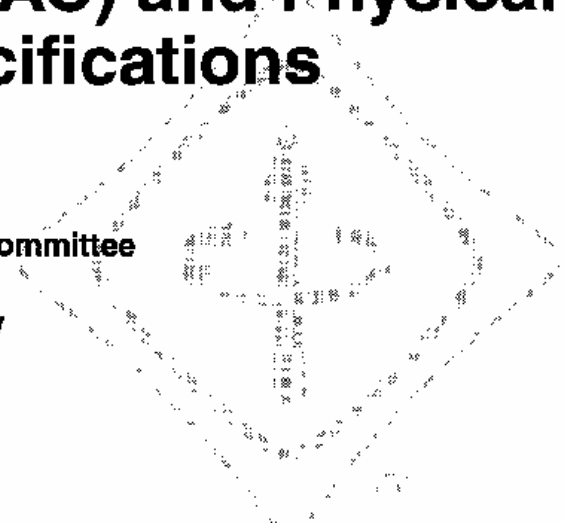
Русское изложение документа

Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks—
Specific requirements—

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications



Sponsor

LAN MAN Standards Committee of the IEEE Computer Society

Approved 26 June 1997

IEEE Standards Board

Перевод: рер

Содержание

Figures

Tables

1 Обзор

1 # 2 Нормативные документы

3 Определения

4 Сокращения

ACK	Подтверждение
AID	Идентификатор соответствия
AP	Точка доступа
ATIM	Сообщение индикации об объявлении трафика
BSA	Базовая область обслуживания
BSS	Базовый набор обслуживания
BSSID	Идентификация базового набора обслуживания
CCA	
CF	
CFP	
CID	
CP	
CRC	
CS	
CTS	
CW	
DA	
DBPSK	
DCE	
DCF	
DCLA	
DIFS	
DLL	
Dp	
DQPSK	
DS	
DSAP	
DSM	
DSS	
DSSS	
DTIM	
ED	
EIFS	
EIRP	
ERS	
ESA	
ESS	
FC	
FCS	
FER	
FH	
FHSS	
GFSK	
IBSS	
ICV	
IDU	
IFS	
IMp	
IR	
ISM	
IV	

LLC
LME
LRS
MAC
MDF
MIB
MLME
MMPDU
MPDU
MSDU
NAV
PC
PCF
PDU
PHY
PHY-SAP
PIFS
PLCP
PLME
PMD
PMD-SAP
PN
PPDU
PPM
PRNG
PS
PSDU
RA
RF
RSSI
RTS
RX
SA
SAP
SDU
SFD
SIFS
SLRC
SME
SMT
SQ
SRC
SS
SSAP
SSID
SSRC
STA
TA
TBTT
TIM
TSF
TU
TX

TXE
WAN
WDM
WDS
WEP
WM

5 Общее описание

5.1 *Общее описание архитектуры*

В данном разделе представлены концепции и терминология, используемые в стандарте IEEE Std 802.11-1997 (который в дальнейшем будет именоваться IEEE 802.11). Специфические термины определены в разделе 3. Иллюстрации, приведенные в тексте, описывают ключевые взаимосвязи и концепцию архитектурных компонентов IEEE 802.11. Архитектура используется для описания функциональных компонентов IEEE 802.11 LAN и не подразумевает какого-либо конкретного физического исполнения.

5.1.1 Чем отличаются беспроводные системы LAN

Беспроводные сети обладают такими фундаментальными характеристиками, которые в значительной степени отличают их от традиционных проводных LAN.

5.1.1.1 Адрес назначения не совпадает с местом назначения

В проводных LAN адрес назначения эквивалентен физическому местоположению. При проектировании проводных LAN это предполагается вполне очевидно. В IEEE 802.11 адресуемой единицей является станция (STA). STA является местом назначения сообщения, но в общем случае может не иметь фиксированного местоположения.

5.1.1.2 Учет воздействия среды распространения

Физические уровни, используемые в IEEE 802.11, кардинально отличаются от проводных сред распространения. Особенности физических уровней в IEEE 802.11 состоят в следующем:

a) Использование среды, которая не имеет абсолютных видимых границ распространения, накладывает ограничения на трансиверы физического уровня, которые при соответствующих условиях не могут принять фреймы.
b) Незащищенность от внешних сигналов.
c) Связь через подобную среду значительно уменьшает надежность по сравнению с проводными системами.
d) Динамическая топология сети.
e) Отсутствие полной соединенности в сети, что делает невозможным постоянную слышимость каждой STA (т.е. STA могут быть «спрятаны» друг от друга).
f) Распространение сигналов является асимметричным и зависит от времени.

Из-за перечисленных ограничений беспроводные LAN предназначены для покрытия вполне определенного географического расстояния и могут быть построены из базовых блоков, охватывающих определенную территорию.

5.1.1.3 Обслуживание мобильных станций

Одним из требований IEEE 802.11 является обслуживание мобильных так же, как и портативных станций. Портативной станцией является та, которая перемещается с места на место, но работает только в фиксированном месте расположения. Мобильные станции имеют реальный доступ к LAN во время движения.

По ряду технических причин обслуживание только портативных станций является недостаточным. Эффекты распространения размывают различия между портативными и мобильными станциями; стационарные устройства часто становятся мобильными из-за влияния эффектов распространения.

Другой особенностью мобильных станций является то, что они чаще всего питаются от батарей. Таким образом, важное значение приобретает управление питанием. Например, нельзя точно утверждать, что приемник мобильной станции всегда должен быть включен.

5.2 *Компоненты архитектуры IEEE 802.11*

Архитектура IEEE 802.11 состоит из различных компонентов, которые взаимодействую между собой для обеспечения функционирования беспроводной LAN, поддерживающей обслуживание мобильности станций для верхних уровней.

Базовый набор служб (BSS) является основным блоком IEEE 802.11 LAN. На Рис. 1 показаны два BSS, каждый из которых имеет две станции.

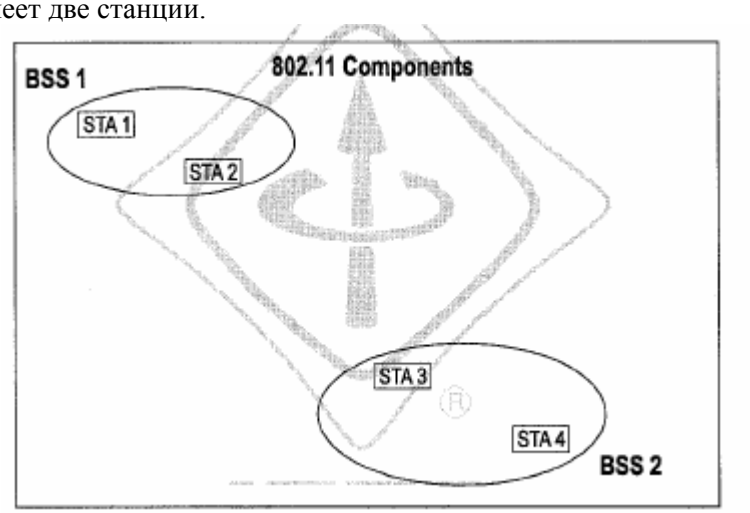


Рис. 1. Базовые наборы служб

5.2.1 Независимый BSS как отдельная сеть

Независимый BSS (IBSS) является наиболее часто используемым типом IEEE 802.11 LAN. Минимально IEEE 802.11 LAN может состоять из двух станций.

На Рис. 1 показаны два IBSS. Данный режим работы возможен в том случае, когда станции IEEE 802.11 могут взаимодействовать непосредственно друг с другом. Поскольку такой тип IEEE 802.11 LAN чаще всего формируется без предварительного планирования и только на время, пока LAN необходима, он называется отдельной сетью (ad hoc network).

5.2.1.1 Ассоциация STA – BSS является динамической

Ассоциация между STA и BSS является динамической (STA включается, выключается, входит и выходит из диапазона видимости). Для того чтобы стать членом инфраструктуры BSS, станция должна быть «ассоциирована». Такие ассоциации являются динамическими и требуют участия службы распределения системы (DSS), которая будет описана позже.

5.2.2 Концепция распределения в системе

Физические ограничения определяют расстояние станция-станция, которое может быть поддержано. Для некоторых сетей это расстояние является достаточным; в других сетях требуется увеличение покрытия.

Кроме независимой конфигурации BSS также могут формировать расширенную сеть, состоящую из нескольких BSS. Архитектурный компонент, используемый для взаимосвязи разных BSS, называется распределительной системой (DS).

IEEE 802.11 логически разделяет беспроводную среду (WM) от среды распределительной системы (DSM). Каждая логическая среда используется для определенных целей различными компонентами архитектуры. Определения IEEE 802.11 не требуют, но и не препятствуют тому, чтобы несколько сред распространения были одинаковыми или разными.

Утверждение о том, что несколько сред распространения логически отличаются друг от друга, является ключом к пониманию гибкости архитектуры. Архитектура IEEE 802.11 LAN определяется независимо от физических характеристик какого-либо конкретного исполнения.

DS разрешает поддержку мобильных устройств путем предоставления логических служб, необходимых для обслуживания адресной доставки и интеграции нескольких BSS.
Точкой доступа (AP) является STA, которая обеспечивает доступ к DS, предоставляя службы DS в дополнение к своей собственной работе как STA.
На Рис. 2 к архитектуре IEEE 802.11 добавлены компоненты DS и AP.

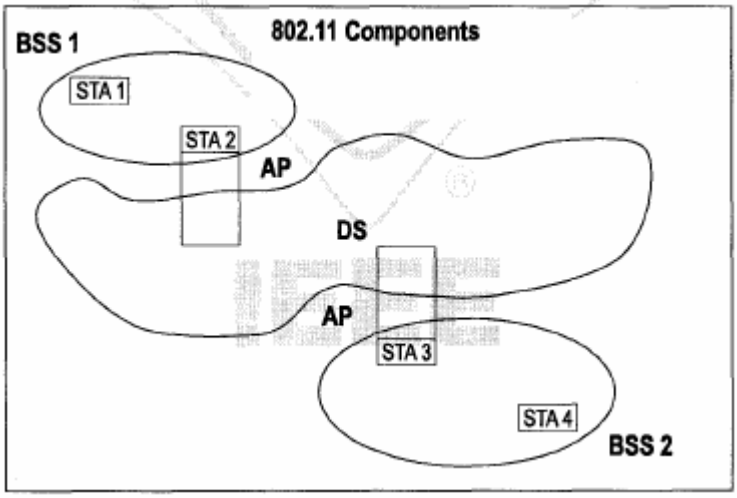


Рис. 2. Распределительные системы и точки доступа

Данные переносятся между BSS и DS через AP. Заметим, что все AP также являются и STA, т.е. адресуемыми устройствами. Адреса, используемые AP для связи через WM и DSM, необязательно являются одинаковыми.

5.2.2.1 Расширенный набор служб (ESS): большое покрытие сети

DS и BSS позволяют создавать в IEEE 802.11 беспроводные сети различного размера и сложности. IEEE 802.11 называет такой тип сетей сетями с расширенным набором служб (ESS).
Ключевым моментом является то, что сеть ESS выглядит для уровня LLC так же, как сеть IBSS. Станции внутри ESS могут взаимодействовать, а мобильные станции могут перемещаться от одного BSS к другому (внутри того же ESS) и все это совершенно прозрачно для LLC.
На Рис. 3 представлена схема расширенного набора служб, причем безотносительно конкретного физического положения BSS.

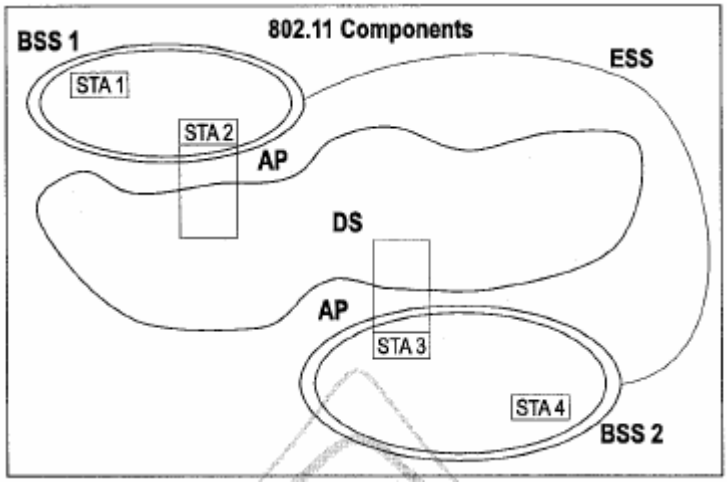


Рис. 3. Расширенный набор служб

Данная схема предоставляет следующие возможности:

a) BSS могут частично перекрываться. Это часто используется для достижения непрерывного покрытия внутри определенного физического объема.

b) BSS могут быть физически разъединены. Логически не существует каких-либо ограничений по расстоянию между BSS.
c) BSS могут быть физически соединены. Это может делаться для получения избыточности.
d) Одна (или более) сетей IBSS или ESS могут быть физически представлены на одном и том же пространстве как одна (или более) сетей ESS. Это может быть сделано по ряду причин. Две наиболее главные из них: когда отдельная сеть работает в пространстве, на котором уже есть сеть ESS; и когда физически перекрывающиеся сети IEEE 802.11 устанавливаются различными организациями.

5.2.3 Общая концепция сети

При использовании беспроводных сетей полное покрытие невозможно. Характеристики распространения являются динамическими и непредсказуемыми. Небольшие изменения в положении или направлении могут привести к огромным изменениям в силе и качестве сигнала. Аналогичные эффекты происходят независимо от того, является станция мобильной или стационарной (поскольку другие движущиеся объекты могут влиять на распространение от станции к станции).

На Рис. 4 показана карта распределения сигнала для простой квадратной комнаты с обычным металлическим столом и открытой дверью. На данном рисунке приведен статический снимок; картина распространения сигнала изменяется динамически по мере передвижения станций и объектов внутри среды. Темные (сплошные) блоки в левом нижнем углу рисунка показывают металлический стол, а в правом верхнем углу находится дверной проем. На данном рисунке можно видеть, как сильно изменяется поле сигнала даже в статической среде.

Хотя на архитектурных диаграммах границы BSS показаны четко, на самом деле это просто красивая картинка, не имеющая ничего общего с физической реальностью. Тем не менее, подобное изображение используется в IEEE 802.11, так как трехмерные изображения интенсивности сигнала являются очень сложными для рисования.

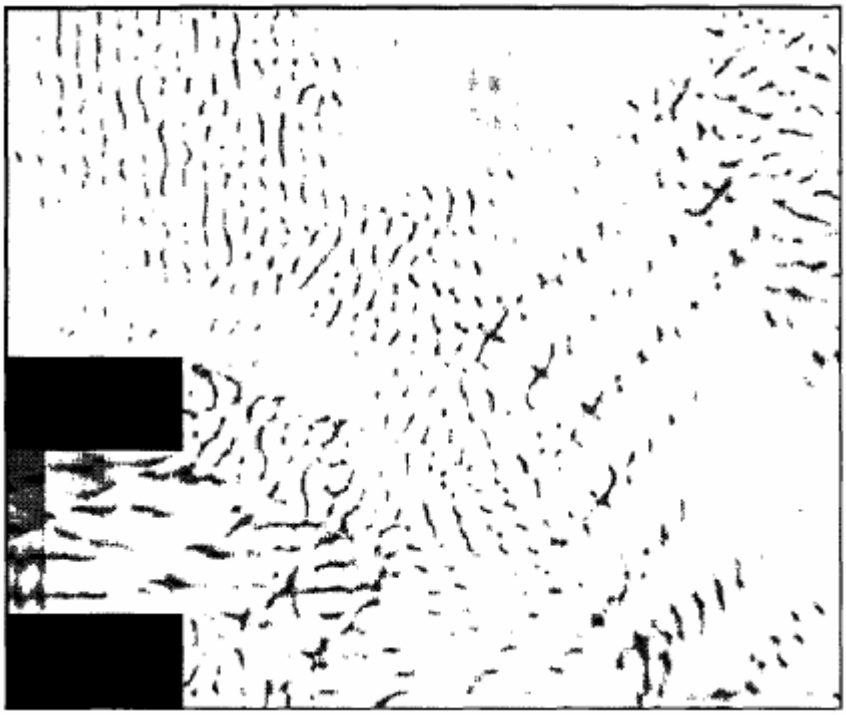


Рис. 4. Пример распределения интенсивности сигнала

Еще одна трудность возникает при попытке описать расположенные рядом и перекрывающиеся области. Обсудим Рис. 5, на котором STA 6 может принадлежать как BSS 2, так и BSS 3.

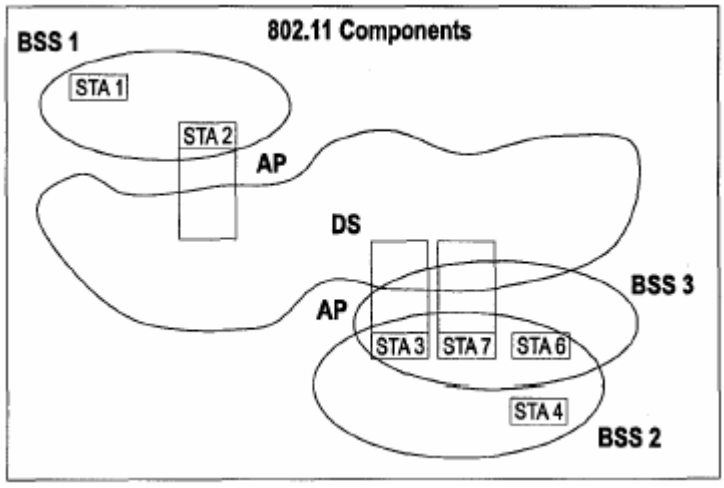


Рис. 5. Перекрывающиеся области

Несмотря на то, что понятие набора станций является более корректным, принято говорить об областях. Во многих разделах понятие области является достаточным. Объем (volume) является более точным термином, чем область (area), хотя и не совсем технически корректным. По ряду исторических причин в настоящем стандарте используется более общий термин область.

5.2.4 Интеграция с проводными LAN

Для интеграции архитектуры IEEE 802.11 с традиционными проводными LAN вводится последний логический архитектурный компонент – портал.

Портал – это логическая точка, в которой MSDU из сети LAN стандарта, отличного от IEEE 802.11, поступают на IEEE 802.11 DS. Например, на Рис. 6 показан портал, соединенный с проводной LAN 802.

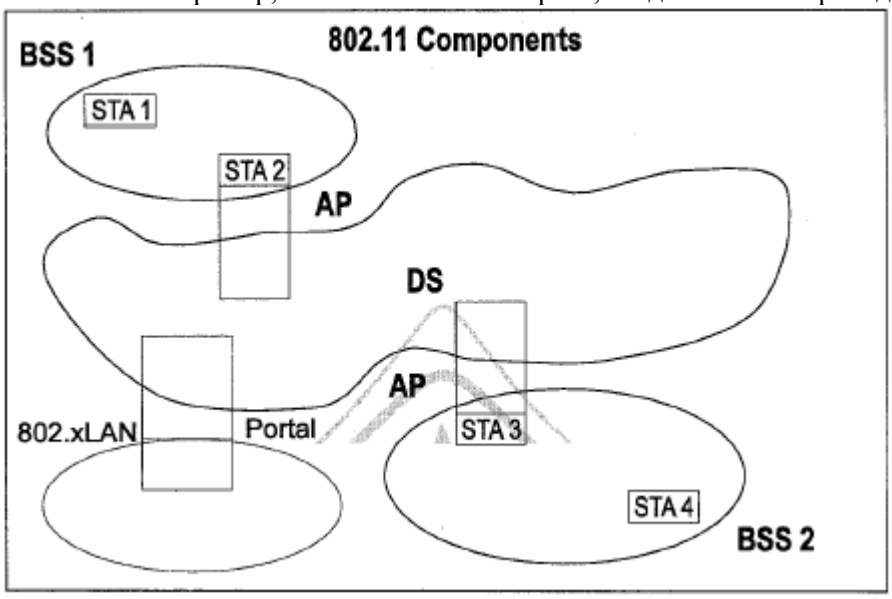


Рис. 6. Соединение с другими IEEE 802 LAN

Все данные из сети LAN стандарта, отличного от IEEE 802.11, поступают на IEEE 802.11 через портал. Портал обеспечивает логическую интеграцию между архитектурой IEEE 802.11 и существующими проводными LAN. Один прибор может обеспечивать обе функции – AP и портала; при этом возможен случай, когда DS выполняется из компонентов 802 LAN.

В IEEE 802.11 архитектура ESS (AP и DS) обеспечивает сегментацию трафика и расширение диапазона. Логическое соединение между IEEE 802.11 и другими LAN осуществляется через портал. Портал осуществляет соединение между средами DSM и LAN, поэтому является интегрированным.

5.3 *Интерфейсы логических служб*

Архитектура IEEE 802.11 допускает возможность того, что DS может отличаться от существующих проводных LAN. DS может создаваться по различным технологиям, включая имеющиеся проводные LAN 802. IEEE 802.11 не ограничивает построение DS сетевым уровнем или уровнем передачи данных. Также нет ограничений по вопросу, является DS централизованным или распределенным по своей структуре.

IEEE 802.11 явно не описывает детали построения DS. Вместо этого IEEE 802.11 определяет службы. Службы соответствуют различным компонентам архитектуры. Существует две категории служб в IEEE 802.11 – станционные службы (SS) и службы распределенной системы (DSS). Обе категории служб используются подуровнем MAC IEEE 802.11.

Полный набор служб IEEE 802.11 включает следующее:

a) Аутентификация
b) Ассоциация
c) Деаутентификация
d) Дисассоциация
e) Распределение
f) Интеграция
g) Секретность
h) Реассоциация
i) Доставка MSDU

Этот набор служб подразделяется на две группы: те, которые являются частью каждой станции, и те, которые являются частью DS.

5.3.1 Станционная служба (SS)

Служба, обеспечиваемая станцией, называется станционной службой.

SS предоставляются на каждой станции IEEE 802.11 (включая AP). SS определены для использования устройствами подуровня MAC. Все соответствующие станции предоставляют DS.

SS включают следующее:

a) Аутентификация
b) Деаутентификация
c) Секретность
d) Доставка MSDU

5.3.2 Служба распределенной системы (DSS)

Служба, обеспечиваемая DS, называется службой распределенной системы.

Такие службы представлены в архитектуре IEEE 802.11 в виде стрелок внутри AP; это указывает на то, что данные службы используются на всем протяжении среды и логических границ адресного пространства. Это наиболее удобное место изображения служб на рисунке. Физическая реализация различных служб может находиться внутри или снаружи физической AP.

DSS обеспечиваются DS. Доступ к ним осуществляется через STA, которые также обеспечиваются DSS. STA, которая обеспечивает доступ к DSS, является AP.

DSS включают следующее:

a) Ассоциация
b) Дисассоциация
c) Распределение
d) Интеграция
e) Реассоциация

DSS определены для использования устройствами подуровня MAC.

На Рис. 7 объединены компоненты с предыдущих рисунков с обоими типами служб, чтобы показать полную архитектуру IEEE 802.11.

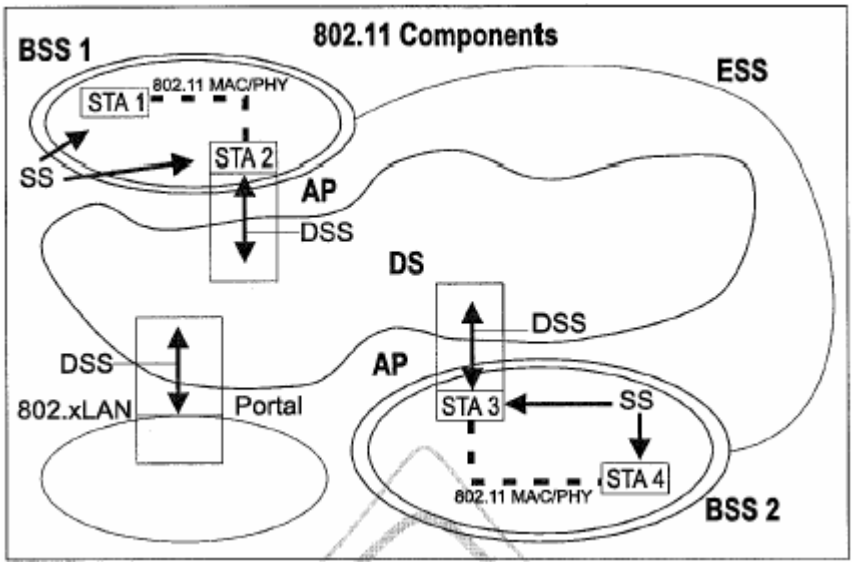


Рис. 7. Полная архитектура IEEE 802.11

5.3.3 Множественные логические адресные пространства

Наряду с тем, что архитектура IEEE 802.11 предоставляет возможность существования WM, DSM и интегрированных проводных LAN в различных физических средах, она также разрешает каждому из этих компонентов работать внутри различных адресных пространств. IEEE 802.11 использует и определяет только адресное пространство WM.

Каждый физический уровень IEEE 802.11 работает в одной среде – WM. MAC IEEE 802.11 работает в одном адресном пространстве. Таким образом, нет необходимости явно определять эти адреса как «WM адреса». Это подразумевается на протяжении всего стандарта IEEE 802.11.

В IEEE 802.11 выбрано 48-битовое адресное пространство IEEE 802 (см. 7.1.2.3.1). То есть адреса IEEE 802.11 совместимы с адресным пространством, используемым семейством LAN 802.

Такой выбор адресного пространства приводит к тому, что адресное пространство MAC проводного LAN и адресное пространство MAC IEEE 802.11 могут быть одинаковыми. В тех случаях, когда DS использует соответствующую адресацию уровня MAC 802, все три логических адресных пространства, используемых в системе, могут быть идентичными. Несмотря на то, что это общий случай, он не единственный, разрешаемый архитектурой. Архитектура IEEE 802.11 предоставляет возможность всем трем логическим адресным пространствам быть различными.

Пример множественного адресного пространства – когда реализация DS использует адресацию сетевого уровня. В этом случае адресное пространство WM и адресное пространство DS буду отличаться.

Возможность обрабатывать несколько логических сред и адресных пространств является ключевой в IEEE 802.11, что делает его независимым от реализации DS и позволяет легко разрабатывать интерфейсы для мобильных приложений сетевого уровня. Реализация DS не обсуждается и лежит за пределами данного стандарта.

5.4 *Обзор служб*

Существует девять служб, определяемых IEEE 802.11. шесть из них используются для поддержки доставки MSDU между STA. Три оставшихся используются для управления доступом и обеспечения конфиденциальности.

В данном разделе представлены службы, обзор их использования и описание взаимодействия с другими службами и архитектурой IEEE 802.11. Службы представлены в таком порядке, чтобы легче было понять работу сети ESS IEEE 802.11. Поэтому SS и DSS перемешаны друг с другом (а не сгруппированы по категориям).

Каждая из служб поддерживается одним или более типами фреймов MAC. Некоторые службы обеспечиваются управляющими сообщениями MAC, а некоторые – сообщениями данных. Все сообщения по-

лучают доступ к WM через подуровень MAC IEEE 802.11 с помощью метода, описанного в разделе 9 данного стандарта.
Подуровень MAC IEEE 802.11 использует три типа сообщений – данные, менеджмент и управление (см. раздел 7). Сообщения данных доставляются по служебному пути данных MAC.
Сообщения менеджмента используются для поддержки служб IEEE 802.11 и доставляются по служебному пути менеджмента данных MAC.
Управляющие сообщения используются для поддержки доставки сообщений менеджмента и данных.

5.4.1 Распределение сообщений внутри DS

5.4.1.1 Распределение

Это основная служба, используемая станциями IEEE 802.11. STA, работающая в ESS, обязательно вызывает эту службу для каждого сообщения данных (когда фрейм посылается через DS). Распределение – это DSS.
Рассмотрим сеть ESS на Рис. 7 и предположим, что сообщение данных посылается от STA1 на STA4. Сообщение посылается с STA1 и принимается на STA2 («вход» AP). AP предоставляет сообщению службу распределения DS.
Основной задачей службы распределения является доставка сообщения внутри DS таким образом, чтобы оно прибывало в соответствующее место DS назначенному приемнику.
В данном примере сообщение распределяется на STA3 («выход» AP), а STA3 получает доступ к WM для посылки сообщения на STA4 (назначенный приемник).
То, каким образом сообщение распределяется внутри DS, не описано в стандарте IEEE 802.11. Все, что требуется в стандарте, это обеспечить DS достаточной информацией, чтобы можно было установить «выходную» точку, соответствующую назначенному приемнику. Необходимая информация предоставляется DS с помощью трех служб (ассоциация, реассоциация и дисассоциация).
В предыдущем примере рассмотрен случай, когда AP, которая вызывает службу распределения, отличается от AP, которая принимает распределенное сообщение. Если сообщение назначено станции, которая является членом BSS, то «вход» и «выход» AP для сообщения являются одинаковыми.
В любом случае служба распределения логически вызывается. При этом не важно, должно ли сообщение действительно пройти через физический DSM или нет, это определяется способом исполнения DS и не описывается в стандарте IEEE 802.11.
Поскольку IEEE 802.11 не описывает исполнение DS, он также не распознает и поддержку использования WM в качестве DSM. Такая поддержка заложена в форматах фреймов IEEE 802.11 (см. раздел 7).

5.4.1.2 Интеграция

Если служба распределения устанавливает, что место назначения сообщения является членом интегрированной LAN, то «выходной» точкой DS может быть портал, а не AP.
Сообщения, которые распределяются в портал, заставляют DS вызывать функцию интеграции (концептуально сразу после службы распределения). Функция интеграции отвечает за выполнение необходимых действий в том случае, когда требуется доставка сообщения из DSM в интегрированную среду LAN (включая все необходимые ретрансляции для типа среды и адресного пространства). Интеграция – это DSS.
При приеме сообщений на IDS из интегрированной LAN (через портал) STA вызывает функцию интеграции до того, как сообщение будет распределено службой распределения.
Детальное описание функции интеграции зависит от конкретного исполнения DS и находится за пределами обзора данного стандарта.

5.4.2 Службы, которые поддерживают службу распределения

Основной целью подуровня MAC является передача MSDU между устройствами подуровней MAC. Информация, необходимая для работы службы распределения, обеспечивается службами ассоциации. Перед тем, как сообщение может быть обработано службой распределения, STA должна быть «ассоциирована».

Для уяснения понятия ассоциации сначала необходимо определить понятие мобильности.

5.4.2.1 Типы мобильности

Существует три типа переходов, которые являются значительными для данного стандарта и описывают мобильность станции внутри сети:

a) **Нет перехода**: Для данного типа существует два подкласса, которые обычно очень трудно различить:
   1) Статический – отсутствие движение.
   2) Локальное перемещение – перемещение внутри физического диапазона соединенных STA (т.е., перемещение внутри базовой области обслуживания (BSA)).
b) **BSS-переход**: Данный тип определяется как перемещение станции от одной BSS к другой внутри одного и того же ESS.
c) **ESS-переход**: Данный тип определяется как перемещение станции от BSS в одной ESS к BSS в другой ESS. Данный случай поддерживается только в том случае, когда STA может двигаться. Поддержка соединений верхнего уровня в данном случае не может быть гарантирована стандартом IEEE 802.11; на самом деле вполне возможно разрушение службы.

Различные службы ассоциации поддерживают различные типы мобильности.

5.4.2.2 Ассоциация

Для доставки сообщения внутри DS службе распределения необходимо знать, какая AP имеет доступ к нужной STA IEEE 802.11. Данная информация обеспечивается на DS понятием «ассоциации». Ассоциация является необходимым, но не достаточным условием для поддержки BSS-перехода. Ассоциация является достаточным условием для поддержки мобильности «без перехода». Ассоциация – это DSS.

До того, как STA будет разрешено послать сообщение данных через AP, она должна стать ассоциированной внутри AP. С этой целью вызывается функция ассоциации, которая обеспечивается STA на соответствующей AP. DS использует данную информацию для обеспечения своей службы распределения сообщений. Данный стандарт не определяет, каким образом информация, поставляемая службой ассоциации, хранится и обрабатывается внутри DS.

В любой данный момент времени STA может быть ассоциирована не более чем с одной AP. При этом можно получить однозначный ответ на вопрос «какая AP обслуживает STA X?». Как только ассоциация завершена, STA может полностью использовать DS для связи (через AP). Ассоциация всегда инициируется мобильной STA, но не AP.

AP может быть одновременно ассоциирована с несколькими STA.

STA распознает присутствие AP и запрашивает создание ассоциации путем вызова службы ассоциации. Более подробное описание того, как станция узнает о присутствии AP, приведено в разделе 11.1.3.

5.4.2.3 Реассоциация

Ассоциация является достаточной для доставки сообщений без переходов между станциями IEEE 802.11. Для поддержки BSS-переходов необходима дополнительная функциональная возможность, обеспечиваемая функцией реассоциации. Реассоциация – это DSS.

Функция реассоциации вызывается для «перемещения» текущей ассоциации от одной AP к другой. При этом на DS сохраняется информация о текущем соответствии между AP и STA, как если бы станция перемещалась от BSS к BSS внутри ESS. Реассоциация также разрешает изменение атрибутов уже созданной ассоциации, в то время как STA остается ассоциированной с той же самой AP. Реассоциация всегда инициируется мобильной станцией.

5.4.2.4 Дисассоциация

Служба дисассоциации вызывается всякий раз, когда нужно завершить существующую ассоциацию. Дисассоциация – это DSS.

В ESS это означает, что DS должна стереть существующую информацию ассоциации. Попытка послать сообщение через DS на дисассоциированную STA будет безуспешной.

Служба ассоциации может быть вызвана любым из членов ассоциации (STA, не являющейся AP, или AP). Дисассоциация является не запросом, а уведомлением. Дисассоциация не может быть проигнорирована ни одним из членов ассоциации.
AP может потребоваться дисассоциировать STA, чтобы позволить удаление AP из сети для обслуживания или по какой-либо другой причине.
STA должны осуществлять попытку дисассоциации всякий раз, когда они покидают сеть. Тем не менее, протокол MAC не зависит от того, вызывают ли STA функцию дисассоциации. (Управление MAC спроектировано таким образом, чтобы приспосабливаться к потере ассоциированных STA).

5.4.3 Службы управления доступом и конфиденциальности

В IEEE 802.11 необходимо наличие двух служб, обеспечивающих функциональную эквивалентность проводным LAN. Проектирование проводных LAN учитывает физические атрибуты линии. В частности, проводные LAN проектируются в предположении, что проводная среда полностью закрыта и управляема. Физически открытая среда IEEE 802.11 LAN отвергает такие предположения.
Таким образом, вводится две службы, обеспечивающие функциональную эквивалентность проводным LAN, аутентификация и секретность. Аутентификация используется вместо физического соединения проводной среды. Секретность используется для обеспечения условий конфиденциальности проводной среды.

5.4.3.1 Аутентификация

В проводных LAN могут быть использованы физические методы безопасности для предотвращения несанкционированного доступа. Сие является непрактичным для беспроводных LAN, поскольку они работают в среде, не имеющей четких границ.
IEEE 802.11 обеспечивает возможность управления доступом к LAN через службу аутентификации. Данная служба используется всеми станциями для создания их тождественности со станциями, с которыми они будут связываться. Это является истинным как для сетей ESS, так и IBSS. Если между двумя станциями не был создан обоюдно приемлемый уровень аутентификации, ассоциации не будет. Аутентификация – это SS.
IEEE 802.11 поддерживает различные процессы аутентификации. Кроме того, механизм аутентификации позволяет расширить поддерживаемые схемы аутентификации. IEEE 802.11 не предписывает использование каких-либо отдельных схем аутентификации.
IEEE 802.11 обеспечивает аутентификацию связного уровня (link) между станциями. IEEE 802.11 не обеспечивает ни сквозную аутентификацию (источник сообщения – приемник сообщения), ни аутентификацию пользователь-пользователь. Аутентификация IEEE 802.11 используется просто для переноса на беспроводное соединение условий, накладываемых физическими стандартами проводной связи. (Такое использование аутентификации абстрагируется от любых конкретных процессов аутентификации, которые могут использоваться на верхних уровнях сетевого протокольного стека). Если необходимо наличие аутентификации, отличной от описанной здесь, рекомендуется применение стандарта IEEE Std 802.10-1992 [B3].
Если необходимо, сеть IEEE 802.11 может работать с использованием Открытой Системной аутентификации (см. 8.1.1). При этом могут наблюдаться отклонения от предположений, делаемых по умолчанию на верхних уровнях. В Открытой Системе любая станция может стать аутентифицированной.
IEEE 802.11 также поддерживает аутентификацию Shared Key. Использование этого механизма аутентификации требует выполнения опции WEP (см. 8.2). В системе с аутентификацией Shared Key тождественность демонстрируется путем знания общего секретного ключа шифрования WEP.
Для поддержки стандартизированных схем аутентификации предназначены функции информационной базы управления (MIB).
IEEE 802.11 требует наличия взаимно приемлемой и успешной аутентификации.
В любой данный момент времени STA может быть аутентифицирована с несколькими другими STA.

5.4.3.1.1 Преаутентификация

Поскольку процесс аутентификации может занять длительное время (в зависимости от используемого протокола), служба аутентификации может вызываться независимо от службы ассоциации.
Преаутентификация обычно осуществляется STA, уже ассоциированной с AP (с которой она первоначально аутенцифицировалась). IEEE 802.11 не требует, чтобы STA преаутенцифицировалась с AP. Однако, аутентификация необходима до того, как может быть создана ассоциация.
Если аутентификация была потеряна во время реассоциации, это может значительно уменьшить скорость, с которой STA может реассоциироваться между AP, ограничивая качество мобильности BSS-перехода. Использование преаутентификации позволяет службе аутентификации перекрыть критичный ко времени процесс реассоциации.

5.4.3.2 Деаутентификация

Служба деаутентификации вызывается всякий раз, когда нужно завершить существующую аутентификацию. Деаутентификация – это SS.
Поскольку в ESS аутентификация является предварительно необходимой для ассоциации, вызов деаутентификации должен приводить к дисассоциации станции. Служба деаутентификации может быть вызвана любым из аутентифицированных членов (STA, не являющейся AP, или AP). Деаутентификация является не запросом, а уведомлением. Деаутентификация не может быть проигнорирована ни одним из аутентифицированных членов. Когда AP посылает уведомление о деаутентификации ассоциированной STA, ассоциация также должна быть разорвана.

5.4.3.3 Секретность

В проводных LAN «услышать» трафик могут только те станции, которые физически подсоединены к линии. В беспроводной среде сие является неверным. Любая станция, соответствующая стандарту IEEE 802.11, может слышать весь трафик, находящийся в переделах физической видимости. Таким образом, соединение любой беспроводной станции (без обеспечения секретности) с существующей проводной сетью LAN может значительно уменьшить уровень секретности этой сети.
Для приведения в соответствие функционального уровня беспроводных LAN к уровню, предполагаемому по умолчанию в проводных сетях, IEEE 802.11 предоставляет возможность шифрования содержимого сообщений. Данная функциональная возможность предоставляется службой секретности. Секретность – это SS.
IEEE 802.11 определяет необязательный алгоритм шифрования (WEP), который разработан с целью достичь секретности, «эквивалентной» проводным LAN. Алгоритм не рассчитан на обеспечение суперсекретности, однако, считается достаточным для обеспечения секретности «как минимум как у проводной линии». Более подробно см. Главу 8.
IEEE 802.11 использует алгоритм WEP для шифрования сообщений. Функции MIB обеспечиваются для поддержки WEP.
Заметим, что функция секретности может быть вызвана только для фреймов данных и некоторых фреймов управления аутентификацией. Все станции начинают работать «в явном виде» для того, чтобы установить службы аутентификации и секретности.
По умолчанию для всех станций IEEE 802.11 состояние секретности является выключенным (in the clear). Если служба секретности не вызвана, то все сообщения должны посылаться незашифрованными. Если такая установка по умолчанию не приемлема для какого-либо из участников связи, то фреймы данных не будут успешно пересылаться между устройствами LLC. Нешифрованные фреймы данных, принимаемые станцией, сконфигурированной в секретном режиме, так же, как и шифрованные фреймы, принимаемые станцией, не имеющей ключа шифрования, опускаются без индикации на LLC (либо без индикации на службу распределения в случае, когда фреймы «для DS» принимаются на AP). Эти фреймы подтверждаются на WM (если они приняты без ошибки во фреймовой проверочной последовательности – FCS) во избежание потери ширины полосы WM при повторной передаче.

5.5 *Связь между службами*

STA хранит две переменных состояния для каждой STA, с которой необходимо прямое соединение через WM:

- Состояние аутентификации: возможные значения – неаутентифицирована и аутентифицирована.
- Состояние ассоциации: возможные значения – неассоциирована и ассоциирована.

С помощью этих переменных создается три локальных состояния для каждой удаленной STA:

- Состояние 1: начальное состояние, неаутентифицирована, неассоциирована.
- Состояние 2: аутентифицирована, неассоциирована.
- Состояние 3: аутентифицирована, ассоциирована.

Связи между переменными состояния станции и службами представлены на Рис. 8.

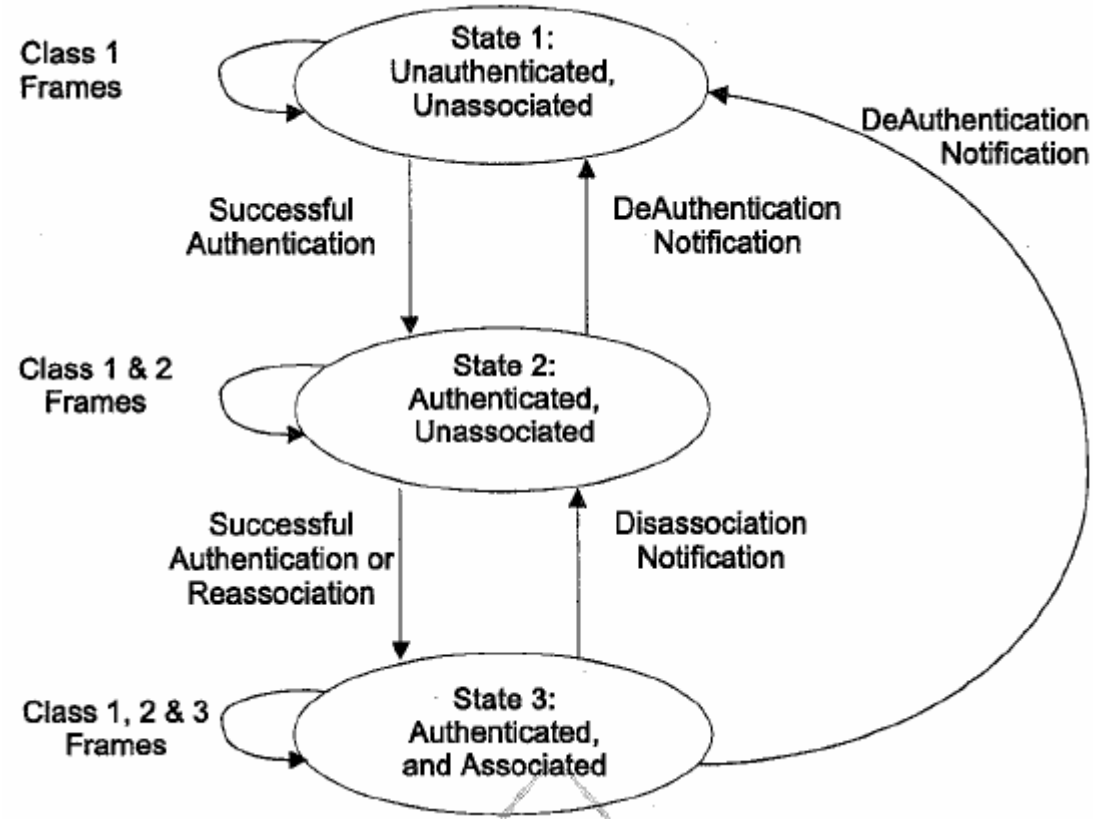


Рис. 8. Связь между переменными состояния и службами

Текущее состояние, существующее между передающей и приемной станциями, определяет типы фреймов IEEE 802.11, которыми они могут обмениваться (см. Главу 7). Состояние передающей STA, представленное на Рис. 8, приведено относительно соответствующей приемной STA. Разрешенные типы фреймов сгруппированы по классам, а классы соответствуют состоянию станции. В Состоянии 1 разрешены только фреймы Класса 1. В Состоянии 2 разрешены фреймы Классов 1 и 2. В Состоянии 3 разрешены все фреймы (Классов 1, 2 и 3). Классы фреймов определяются следующим образом:

a) Фреймы Класса 1 (разрешены в Состояниях 1, 2 и 3):
   1) Фреймы управления
      i) Запрос передачи (RTS)
      ii) Сброс передачи (CTS)
      iii) Подтверждение (ACK)
      iv) Пустой фрейм (CF) – окончание + ACK
      v) CF-окончание
   2) Фреймы менеджмента
      i) Пробный запрос/ответ
      ii) Маяк
      iii) Аутентификация: успешная аутентификация разрешает станции обмениваться фреймами Класса 2. Неуспешная аутентификация оставляет станцию в Состоянии 1.

iv) Деаутентификация: уведомление о деаутентификации, когда станция переходит из Состояния 2 или 3 в Состояние 1. Станция должна вновь стать аутентифицированной до начала посылки фреймов Класса 2.
v) Индикация об объявлении трафика (ATIM)

3) Фреймы данных
i) Данные: фреймы данных с управляющими битами «для DS» и «от DS», которые установлены в ноль.

b) Фреймы Класса 2 (разрешены только в Состояниях 2 и 3, когда станция аутентифицирована)
1) Фреймы менеджмента
i) Запрос/ответ ассоциации
– Успешная ассоциация разрешает фреймы Класса 3.
– Неуспешная ассоциация оставляет станцию в Состоянии 2.
ii) Запрос/ответ реассоциации
– Успешная реассоциация разрешает фреймы Класса 3.
– Неуспешная реассоциация оставляет станцию в Состоянии 2 (относительно STA, которой посылалось сообщение реассоциации). Фреймы реассоциации должны посылаться только в том случае, если приемная STA уже ассоциирована в той же ESS.
iii) Дисассоциация
– Уведомление о дисассоциации, когда станция переходит из Состояния 3 в Состояние 2. Станция должна вновь стать ассоциированной, если она хочет пользоваться DS.

Если STA A принимает фрейм Класса 2 с однонаправленным адресом в поле Address 1 от STA B, которая не аутентифицирована с STA A, STA A должна послать фрейм деаутентификации на STA B.

(Использование слова «принимает» в данном разделе относится к фреймам, которые удовлетворяют всем критериям фильтрации, определенным в Главе 9).

c) Фреймы Класса 3 (разрешены только в Состоянии 3, когда станция ассоциирована)
1) Фреймы данных
– Подтипы данных: фреймы данных разрешены. То есть, управляющие биты «для DS» и «от DS» могут устанавливаться в единицу для пользования DSS.
2) Фреймы менеджмента
– Деаутентификация: уведомление о деаутентификации, когда станция осуществляет дисассоциацию в Состоянии 3, то есть переходит в Состояние 1. Станция должна вновь стать аутентифицированной перед следующей ассоциацией.
3) Фреймы управления
– PS-опрос

Если STA A принимает фрейм Класса 3 с однонаправленным адресом в поле Address 1 от STA B, которая аутентифицирована, но не ассоциирована с STA A, STA A должна послать фрейм дисассоциации на STA B.

Если STA A принимает фрейм Класса 3 с однонаправленным адресом в поле Address 1 от STA B, которая не аутентифицирована с STA A, STA A должна послать фрейм деаутентификации на STA B.

(Использование слова «принимает» в данном разделе относится к фреймам, которые удовлетворяют всем критериям фильтрации, определенным в Главах 8 и 9).

5.6 *Различия между ESS и IBSS LAN'ами*

В разделе 5.2.1 было введено понятие IBSS LAN. Было также замечено, что IBSS часто используется для поддержки отдельной сети. STA в сети IBSS непосредственно связаны с одной или несколькими другими STA.

Рассмотрим полную архитектуру IEEE 802.11, показанную на Рис. 9.

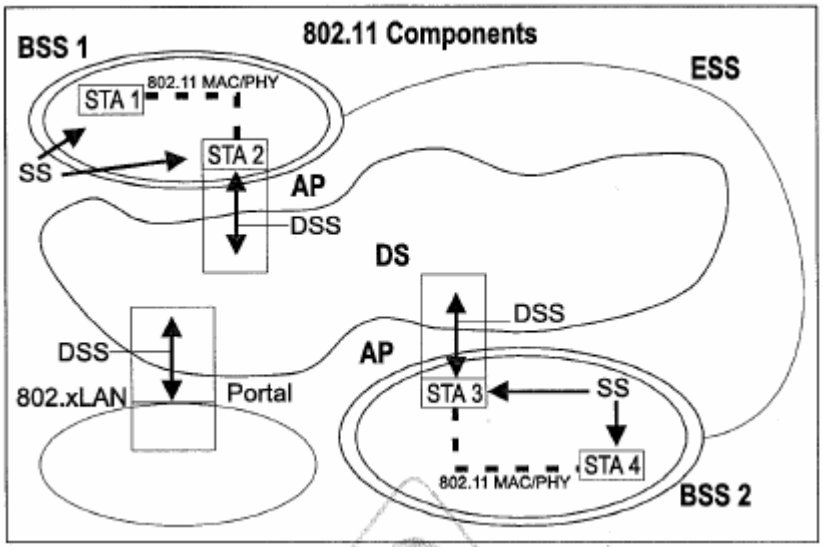


Рис. 9. Архитектура IEEE 802.11 (повтор)

IBSS состоит из STA, которые соединены непосредственно. Таким образом, здесь (по определению) есть только один BSS. Более того, поскольку здесь нет физической DS, то не может быть и портала, интегрированной проводной LAN или DSS. Логическая структура уменьшается до той, что представлена на Рис. 10.

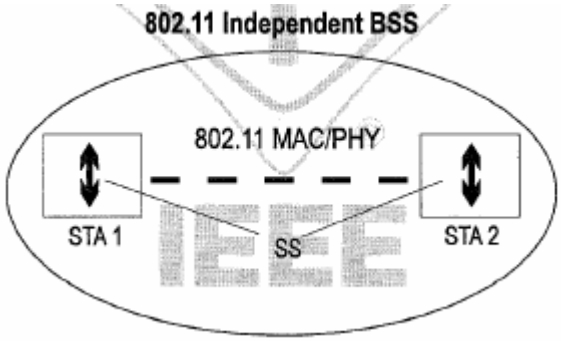


Рис. 10. Логическая архитектура IBSS

На Рис. 10 представлены только две станции (минимально возможный случай). IBSS может иметь произвольное количество членов. В IBSS разрешены только фреймы Классов 1 и 2, поскольку здесь нет DS. Службами, которые прилагаются к IBSS, являются SS.

5.7 *Содержимое информационных сообщений, которые поддерживают службы*

Каждая служба поддерживается одним или более сообщениями IEEE 802.11. Значения информационных пунктов приведены в Главе 7.

5.7.1 Данные

Для посылки данных от одной STA к другой используется сообщение данных следующего вида:

Сообщения данных

- Тип сообщения: данные
- Подтип сообщения: данные
- Информационные пункты:
  - IEEE адрес источника сообщения

- IEEE адрес назначения сообщения
- BSS ID

– Направление сообщения: от STA к STA

5.7.2 Ассоциация

Для ассоциации STA служба ассоциации использует следующие сообщения:

Запрос ассоциации

– Тип сообщения: менеджмент
– Подтип сообщения: запрос ассоциации
– Информационные пункты:
  - IEEE адрес STA, инициирующей ассоциацию
  - IEEE адрес AP, с которой станция будет ассоциирована
  - ESS ID
– Направление сообщения: от STA к AP

Ответ ассоциации

– Тип сообщения: менеджмент
– Подтип сообщения: ответ ассоциации
– Информационные пункты:
  - Результат запрошенной ассоциации. Это пункт со значениями «успешно» или «неуспешно».
  - Если ассоциация является успешной, ответ должен включать идентификатор ассоциации (AID).
– Направление сообщения: от AP к STA

5.7.3 Реассоциация

Для реассоциации STA служба реассоциации использует следующие сообщения:

Запрос реассоциации

– Тип сообщения: менеджмент
– Подтип сообщения: запрос реассоциации
– Информационные пункты:
  - IEEE адрес STA, инициирующей реассоциацию
  - IEEE адрес AP, с которой станция будет реассоциирована
  - IEEE адрес AP, с которой станция ассоциирована в настоящий момент
  - ESS ID
– Направление сообщения: от STA к AP (AP, с которой STA запрашивает реассоциацию)

Адрес текущей AP включен для избыточности. Включение текущего адреса AP позволяет выполнять реассоциацию MAC независимо от исполнения DS.

Ответ реассоциации

– Тип сообщения: менеджмент
– Подтип сообщения: ответ реассоциации
– Информационные пункты:
  - Результат запрошенной реассоциации. Это пункт со значениями «успешно» или «неуспешно».

- Если реассоциация является успешной, ответ должен включать AID.
- Направление сообщения: от AP к STA

5.7.4 Дисассоциация

Для завершения активной ассоциации STA служба дисассоциации использует следующее сообщение:

Дисассоциация

- Тип сообщения: менеджмент
- Подтип сообщения: дисассоциация
- Информационные пункты:
  - IEEE адрес станции, которая будет дисассоциирована. Это должен быть broadcast адрес в том случае, когда AP дисассоциируется со всеми ассоциированными станциями.
  - IEEE адрес AP, с которой станция ассоциирована в настоящий момент
  - ESS ID
- Направление сообщения: от STA к STA (например, от STA к AP или от AP к STA)

5.7.5 Секретность

Для того чтобы STA могла вызвать алгоритм секретности WEP (как указывается соответствующими атрибутами MIB, см. Главу 11), служба секретности осуществляет шифрование MPDU и устанавливает соответствующим образом бит заголовка WEP фрейма (см. Главу 7).

5.7.6 Аутентификация

Для аутентификации одной STA с другой служба аутентификации осуществляет обмен одним или более фреймами менеджмента аутентификации. Точная последовательность фреймов и их содержимое зависит от вызванной схемы аутентификации. Для всех схем аутентификации идентификация алгоритма определяется содержанием тела фрейма менеджмента.
В среде IBSS любая станция может быть инициирующей STA (STA 1). В среде ESS STA 1 – это мобильная STA, а STA 2 – это AP.

Аутентификация (первый фрейм последовательности)

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации
  - Идентификатор станции
  - Последовательный номер процедуры аутентификации
  - Информация, зависящая от алгоритма аутентификации
- Направление сообщения: первый фрейм последовательности всегда направляется от STA 1 к STA 2

Первый фрейм аутентификационной последовательности всегда должен быть нешифрованным.

Аутентификация (промежуточные фреймы последовательности)

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации

    - Последовательный номер процедуры аутентификации
    - Информация, зависящая от алгоритма аутентификации
- Направление сообщения:
    - Четные фреймы последовательности от STA 2 к STA 1
    - Нечетные фреймы последовательности от STA 1 к STA 2

Аутентификация (последний фрейм последовательности)

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
    - Идентификация алгоритма аутентификации
    - Идентификатор станции
    - Последовательный номер процедуры аутентификации
    - Информация, зависящая от алгоритма аутентификации
    - Результат запрошенной аутентификации. Это пункт со значениями «успешно» или «не-успешно».
- Направление сообщения: от STA 2 к STA 1

5.7.7 Деаутентификация

Чтобы прервать активную аутентификацию, STA посылает следующее сообщение:

Деаутентификация

- Тип сообщения: менеджмент
- Подтип сообщения: деаутентификация
- Информационные пункты:
    - IEEE адрес STA, которая будет деаутентифицирована.
    - IEEE адрес STA, с которой STA аутентифицирована в настоящий момент
    - Это должен быть broadcast адрес в том случае, когда STA деаутентифицируется со всеми аутентифицированными станциями.
- Направление сообщения: от STA к STA

5.8 *Модель системы*

Архитектура, приведенная в стандарте, особое внимание уделяет разделению системы на две основные части: связной уровень данных MAC и PHY. Эти уровни введены по аналогии с низшими уровнями из базовой модели ISO/IEC Open System Interconnection (OSI) (ISO/IEC 7498-1: 1994). Уровни и подуровни, описанные в настоящем стандарте, приведены на Рис. 11.

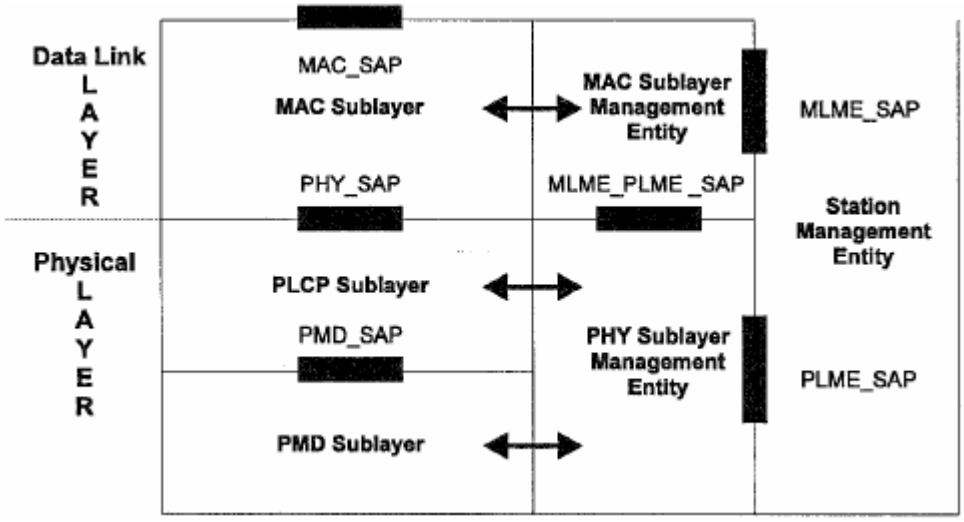


Рис. 11. Часть базовой модели ISO/IEC, представленная в данном стандарте

6 Определение служб MAC

6.1 *Обзор служб MAC*

6.2 *Список и описание примитивов служб MAC*

6.2.1 MA-UNITDATA.request

Функция	Примитив запроса передачи MSDU от местного подуровня LLC к отдельному объекту подуровня LLC, или многим объектам подуровня LLC, в случае групповой адресации.	
Параметры	source address	Адрес источника (SA) определяет индивидуальный MAC адрес объекта подуровня, которому передается MSDU.
	destination address	Адрес приемника (DA) определяет индивидуальный адрес или адрес группы объектов подуровня MAC.
	routing information	Параметр информации направления определяет, маршрут, желательный для передачи данных (пустое значение указывает, что исходное направление не должно использоваться). Для IEEE 802.11, параметр информации направления должен быть пустой.
	data	Специфические особенности параметра данных MSDU, который будет передан объектом подуровня MAC. Для IEEE 802.11, длина MSDU должна быть меньше или равняться 2304 octets.
	priority	Параметр приоритета определяет приоритет, желательный для передачи блока данных. IEEE 802.11 допускает два значения: Contention или ContentionFree.
	service class	Параметр класса обслуживания определяет класс обслуживания, желательный для передачи блока данных. IEEE 802.11 допускает два значения: Reorder-ableMulticast или StrictlyOrdered.
Назначение	Примитив вырабатывается объектом подуровня LLC всякий раз, когда MSDU должен быть передан равноправному объекту или объектам подуровня LLC.	
Действия при получении	Получение этого примитива заставляет объект подуровня MAC добавлять в конец все специфические поля MAC, включая DA, SA, и все поля, которые являются уникальными в IEEE 802.11, и передают должным образом сформированный фрейм более низким слоям для передачи объекту или объектам подуровня MAC.	

6.2.2 MA-UNITDATA.indication

Функция	Примитив определяет передачу MSDU от объекта подуровня MAC до объекта или объектов, в случае адресов группы, подуровня LLC. При отсутствии ошибок, содержание параметра данных логически полно и неизменно относительно параметра данных в связанном примитиве MA-UNTTDATA.request.	
Параметры	source address	SA параметр - индивидуальный адрес, согласно полю SA поступающего фрейма.
	destination address	DA параметр является индивидуальным адресом или адресом группы, согласно полю DA поступающего фрейма.
	routing information	Параметр информации направления определяет маршрут, который использовался для передачи данных. IEEE 802.11 всегда устанавливает это поле в *null*.
	data	Параметр данных определяет MSDU как полученный локальным объектом MAC.
	reception staus	Параметр статуса приема указывает успех или неудачу получения фрейма сообщения IEEE 802.11 через MA-UNTTDATA.indication. MAC сообщает только "об успехе", поскольку все неудачи приема отвергаются без генерации MA-UNTTDATA.indication.
	priority	Параметр приоритета определяет полученный приоритет обработки, который использовался для передачи блока данных. IEEE 802.11 допускает два значения: Contention или ContentionFree.

	service class	Параметр класса обслуживания определяет полученный класс обслуживания, который использовался для передачи блока данных. IEEE 802.11 допускает два значения: ReorderableMulticast или StrictlyOrdered.
Назначение		Примитив MA-UNITDATA.indication передается от объекта подуровня MAC до объекта или объектов подуровня LLC, для сигнализации о получении фрейма в локальном объекте подуровня MAC. Индикация происходит только, если фрейм правильно сформирован на подуровне MAC, получен без ошибок, получен с правильным (или пустым) значением WEP шифрования, и адрес приемника определяет локальный объект подуровня MAC.
Действия при получении		Действия при получении этого примитива подуровнем LLC зависят от правильности и содержания фрейма.

6.2.3 MA-UNITDATA-STATUS.indication

Функция Примитив имеет местное значение и обеспечивает подуровень LLC информацией статуса передачи предшествующего примитива MA-UNITDATA.request.

Параметры

source address — SA параметр - индивидуальный адрес объекта подуровня MAC, как определено в связанном примитиве MA-UNITDATA.request.

destination address — DA параметр является индивидуальным адресом или адресом группы объектов подуровня MAC, как определено в связанном примитиве MA-UNITDATA.request.

transmission status — Параметр статуса передачи будет использоваться, для передачи информации статуса назад к затребовавшему локальному объекту подуровня LLC. IEEE 802.11 определяет следующие значения для статуса передачи:

a) Successful (успешно),
b) Undeliverable (не может быть доставлено) (для неподтвержденных направленных MSDU, когда были превышены лимиты повторения aShortRetryMax или aLongRetryMax),
c) Excessive data length (чрезмерная длина данных),
d) Non-null source routing (непустой исходный маршрут),
e) Unsupported priority (не поддержанный приоритет) (для приоритетов, отличных от Contention или ContentionFree),
f) Unsupported service class (неподдерживаемый класс обслуживания) (для классов обслуживания, отличных от ReorderableMulticast или StrictlyOrdered),
g) Unavailable priority (недопустимый приоритет) (для ContentionFree, когда нет доступных координаторов, когда MSDU передан с обеспеченным приоритетом Contention), и
h) Unavailable service class (недоступный класс обслуживания) (для обслуживания StrictlyOrdered, когда способ управления мощностью станции отличается от "активного"),
i) Undeliverable (не может быть доставлено) (TransmitMSDUTimer достиг aMaxTransmitMSDULifetime прежде, чем произошла успешная
j) доставка),
k) Undeliverable (не может быть доставлено) (нет доступных BSS),
l) Undeliverable (не может быть доставлено) (ключ, упомянутый как aWEPDefaultKeylD или определенный картой ключ пустой).

provided priority — Обеспеченный параметр приоритета определяет приоритет, который использовался для связанной передачи блока данных (Contention или ContentionFree).

provided service class — The provided service class parameter specifies the class of service used for the associated data unit transfer (ReorderableMulticast or StrictlyOrdered). Обеспеченный параметр класса обслуживания определяет класс обслуживания, используемый для связанной передачи блока данных (ReorderableMulticast или StrictlyOrdered).

Назначение	Примитив MA-UNITDATA-STATUS.indication передается от объекта подуровня MAC до объекта подуровня LLC, чтобы указать, что службы выполнил соответствующий примитив MA-UNITDATA.request.
Действия при получении	Действия при получении этого примитива подуровнем LLC зависим от типа действия, используемого объектом подуровня LLC.

1

7 Форматы фреймов

7.1 *Форматы фреймов MAC*

Каждый фрейм состоит из следующих основных компонентов:

a) *Заголовок MAC (MAC header),* который включает управление фреймом, продолжительность, адрес, и информация контроля последовательности.
b) *Тело фрейма (frame body)* переменной длины, которое содержит информацию, определенную *типом* фрейма.
c) *Последовательность проверки фрейма (frame check sequence)* (FCS), которая содержит IEEE 32-битовый циклический кодекс избыточности (CRC).

7.1.1 Общий формат фрейма

Формат фрейма MAC включает набор полей, которые следуют в установленном порядке во всех фреймах:

Таблица 1. Общий формат фрейма

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
Duration / ID	2	
Address 1	6	
Address 2	6	
Address 3	6	
Seqence Control	2	
Address 4	6	
Frame Body	0 – 2312	
FCS	4	

7.1.2 Поля фрейма

7.1.2.1 Поле Frame Control

Поле Frame Control содержит следующие подполя:

Таблица 2. Поле Frame Control

Поле	Длина (в битах)	Биты
Protocol Version	2	0-1
Type	2	2-3
Subtype	4	4-7
To DS	1	8
From DS	1	9
More Frag	1	10
Retry	1	11
Pwr Mgt	1	12
More Data	1	13
WEP	1	14
Order	1	15

7.1.2.1.1 Поле Protocol Version

Поле Protocol Version – длина 2 бита и постоянная для всех версий IEEE Std 802.11. Для этого стандарта, значение версии протокола - 0. Все другие значения сохранены. Уровень версии будет увеличен только, когда обнаружится фундаментальная несовместимость между новой версией и предшествующим изданием стандарта. Устройство, которое получает фрейм с более высоким уровнем версии чем оно поддерживает, откажется от фрейма без индикации передающей станции или LLC.

7.1.2.1.2 Поля Type и Subtype

Поле Type – длина 2 бита и поля Subtype – длина 4 бита. Поля Type и Subtype вместе идентифицируют функцию фрейма. Есть три типа фрейма: контроль, данные, и управление. Каждый из типов фрейма имеет несколько определенных подтипов. Таблица 3 определяет действующие комбинации типа и подтипа.

Таблица 3. Правильные комбинации Type / Subtype

Значение Type биты 3-2	Описание типа	Значение Subtype биты 7-4	Описание подтипа
00	Management	0000	Запрос ассоциации
00	Management	0001	Ответ ассоциации
00	Management	0010	Запрос реассоциации
00	Management	0011	Ответ реассоциации
00	Management	0100	Запрос пробы
00	Management	0101	Ответ пробы
00	Management	0110-0111	Резерв
00	Management	1000	Маяк
00	Management	1001	Объявление сообщения индикации трафика (ATIM)
00	Management	1010	Дисассоциация
00	Management	1011	Аутентификация
00	Management	1100	Деаутентификация
00	Management	1101-1111	Резерв
01	Control	0000-1001	Резерв
01	Control	1010	Энергосбережение (PS)-Poll
01	Control	1011	Запрос передачи (RTS)
01	Control	1100	Готов к передаче (CTS)
01	Control	1101	Подтверждение (ACK)
01	Control	1110	Неконкурентен (CF)-End
01	Control	1111	CF-End + CF-Ack
10	Data	0000	Данные
10	Data	0001	Данные + CF-Ack
10	Data	0010	Данные + CF-Poll
10	Data	0011	Данные + CF-Ack + CF-Poll
10	Data	0100	Пустая функция (нет данных)
10	Data	0101	CF-Ack (нет данных)
10	Data	0110	CF-Poll (нет данных)
10	Data	0111	CF-Ack + CF-Poll (нет данных)
10	Data	1000-1111	Резерв
11	Reserved	0000-1111	Резерв

7.1.2.1.3 Поле To DS

Поле To DS – длина 1 бит, устанавливается в «1» в фреймах данных, предназначенных для DS. Оно включается во все фреймы данных, посланные STA, ассоциированными с AP. Поле To DS устанавливается в «0» во всех других фреймах.

7.1.2.1.4 Поле From DS

Поля From DS – длинна 1 бит, устанавливается в «1» в фреймах данных, выходящих из DS. Во всех остальных фреймах оно устанавливается в «0».

Разрешенные комбинации полей To / From DS и их назначения показаны в Таблица 4.

Таблица 4. Комбинации полей To/From DS в фреймах типа данных

Значение To/From DS	Назначение
To DS = 0 From DS = 0	Данные направляются от одного STA до другого STA в пределах одного IBSS, также как все фреймы управления и контроля.
To DS = 1 FromDS=0	Фрейм данных, предназначенный для DS.
To DS = 0 From DS = 1	Фрейм данных, выходящий из DS.
To DS = 1 From DS = 1	Фрейм беспроводной система распределения (WDS), распределяемый от одной AP к другой.

7.1.2.1.5 Поле More Fragments

Поле More Fragments – длина 1 бит, установлено в «1» во всех фреймах данных или управления, которые имеют другой следующий фрагмент потока MSDU или потока MMPDU. Оно устанавливается в «0» во всех других фреймах.

7.1.2.1.6 Поле Retry

Поле Retry – длина 1 бит, установлено в «1» в любых фреймах данных или управления, которые является повтором более раннего фрейма. Оно устанавливается в «0» во всех других фреймах. Станция назначения использует этот признак для обнаружения и устранения дубликатов фреймов.

7.1.2.1.7 Поле Power Management

Поле Power Management – длина 1 бит, используется для индикации режима управления питанием STA. Значение этого поля остается постоянным в каждом фрейме от отдельного STA в пределах последовательности обмена фреймами, определенной в 9.7. Значение указывает режим, в котором станция будет после успешного завершения последовательности обмена фреймами.

Значение «1» указывает, что STA будет в режиме энергосбережения. Значение «0» указывает, что STA будет в активном режиме. Это поле всегда устанавливается в «0» в фреймах, переданных AP.

7.1.2.1.8 Поле More Data

Поле More Data – длина 1 бит, используется для индикации, что STA находится в режиме энергосбережения, чтобы большее количество MSDUs или MAC PDU управления (MMPDUs) для этой STA буферировалось в AP. Поле More Data имеет силу в направленных данных или фреймах управления, переданных AP к STA в энергосберегающем режиме. Значение «1» указывает, что для того же самого STA присутствует по крайней мере один дополнительный буферированный MSDU или MMPDU.

Поле More Data может быть установлено в «1» в направленных фреймах данных, переданных неконкурентной (CF)-Pollable STA точечному координатор (PC) в ответ на CF-Poll, чтобы указать, что STA имеет по крайней мере один дополнительный буферированный MSDU, доступный для передачи в ответ на последующий CF-Poll.

Поле More Data устанавливается в «0» во всех других направленных фреймах.

Поле More Data устанавливается в «1» в broadcast/multicast фреймах, переданных AP, когда дополнительный broadcast/multicast MSDU или MMPDU остался для передачи AP в течение этого интервала маяка. Поле More Data устанавливается в «0» в broadcast/multicast фреймах, переданных AP когда нет более broadcast/multicast MSDU или MMPDU для передачи AP в течение этого интервала маяка и во всех broadcast/multicast фреймах, переданных non-AP станциями.

7.1.2.1.9 Поле WEP

Поле WEP – длина 1 бит. Устанавливается в «1», если поле Frame Body содержит информацию, которая была обработана алгоритмом WEP. Поле WEP устанавливается в «1» только в пределах фреймов данных управления, подтип аутентификация.

Поле WEP устанавливается в «0» во всех других фреймах. Когда бит WEP установлен в «1», поле Frame Body расширено как определено в 8.2.5.

7.1.2.1.10 Поле Order

Поле Order – длина 1 бит и устанавливается в «1» в любом фрейме данных, который содержит MSDU или фрагментировать его, который передается с использованием класса обслуживания StrictlyOrdered. Это поле устанавливается в «0» во всех других фреймах.

7.1.2.2 Поле Duration/ID

Поле Duration/ID – длина 16 бит. Содержание этого поля следующее:

a) В фреймах контроля энергосбережения (PS)-Poll, поле Duration/ID несет идентичность ассоциации (AID) станции, которая передала фрейм в 14 младших битах, 2-х старших бита в «1». Значение AID находится в диапазоне 1-2007.

b) Во всех других фреймах, поле Duration/ID содержит значение продолжительности каждого фрейма (см. 7.2). Для фреймов, переданных в течение неконкурентного периода (CFP), поле продолжительности установлено в 32 768.

Всякий раз, когда содержание поля Duration/ID меньше чем 32 768, значение длины используется, чтобы модернизировать вектор распределения сети (NAV) согласно процедурам, определенным в 9.

Декодирование поля Duration/ID показано в Таблица 5.

Таблица 5. Декодирование поля Duration/ID

Бит 15	Бит 14	Биты 13-0	Использование
0	0-32767		Продолжительность
1	0	0	Фиксированное значение в пределах фреймов, переданных в течение CFP
1	0	1-16383	Резерв
1	1	0	Резерв
1	1	1-2007	AID в PS-Poll фреймах
1	1	2008-16383	Резерв

7.1.2.3 Поля адреса

Есть четыре поля адреса в формате фрейма MAC. Эти поля используются, чтобы указать BSSID, исходный адрес, адрес приемника, адрес передающей станции и адрес получающей станции. Согласно назначению четыре поля адреса в каждом фрейме обозначены сокращениями BSSID, DA, SA, RA, и TA, указывая BSS идентификатор (BSSID), Адрес получателя, Адрес отправителя, Адрес приемника, и Адрес передатчика, соответственно. Некоторые фреймы могут не содержать некоторые из полей адреса.

Использование некоторых полей адреса определено относительным положением полей Address (1-4) в пределах заголовка MAC, независимо от типа адреса, присутствующего в этом поле. Например, сравнение адреса приемника всегда выполняется с содержимым поля Address 1 в полученных фреймах, а адреса приемника фреймов CTS и ACK всегда с полем Address 2 в соответствующем RTS фрейме, или из подтвержденного фрейма.

7.1.2.3.1 Представление адреса

Каждое поле Address содержит 48-битовый адрес как определено в 5.2 IEEE Std 802-1990.

7.1.2.3.2 Обозначение адреса

Адрес подуровня MAC - один из двух типов:

a) Индивидуальный адрес. Адрес, связанный со специфической станцией в сети.
b) Адрес Группы. Адрес мультиназначения, связанный с одной или более станциями в данной сети. Есть два вида адресов группы:
   1) Multicast - адрес Группы. Адрес, связанный в соответствии с соглашением более высокого уровня с группой логически связанных станций.
   2) Broadcast адрес. Известный, предопределенный адрес multicast, который всегда обозначает набор всех станций на данном LAN. Все «1» в поле Destination address интерпретируются как broadcast адрес. Эта группа предопределена для каждой среды связи, состоящей из всех станций, активно связанных с той средой; это используется, чтобы передать всем активным станциям в этой среде. Все станции способны распознать broadcast адрес. Нет необходимости, чтобы станция были способны формировать broadcast адрес.

   См. также IEEE Std 802-1990.

7.1.2.3.3 Поле BSSID

BSSID - 48-битовое поле того же самого формата как IEEE 802 MAC адрес. Это поле уникально идентифицирует каждый BSS. Значение этого поля, в инфраструктуре BSS, является адресом MAC, используемым в настоящее время STA в AP BSS.

Значение этого поля в IBSS – локально управляемый IEEE адрес MAC, сформированный из 46-битового случайного числа, произведенного согласно процедуре, определенной в 11.1.3. Бит индивидуального/группового адреса установлен в «0». Бит универсального/местного адреса установлен в «1». Этот механизм используется, чтобы обеспечить высокую вероятность отбора уникального BSSID.

Значение все «1» используется, чтобы указать broadcast BSSID. Broadcast BSSID может использоваться только в поле BSSID фрейма управления, подтип запрос пробы.

7.1.2.3.4 Поле «Адрес получателя» (DA)

Поле Адрес получателя (DA) содержит IEEE MAC индивидуальный или групповой адрес, который идентифицирует объект или объекты MAC, определенный как конечный получатель(и) MSDU (или фрагмента его), содержащегося в поле тела фрейма.

7.1.2.3.5 Поле «Адрес источника» (SA)

Поле Адрес источника (SA) содержит IEEE MAC индивидуальный адрес, который идентифицирует объект MAC, от которого передан MSDU (или фрагмент его), содержащейся в поле тела фрейма. Бит индивидуального/группового адреса всегда передается как ноль.

7.1.2.3.6 Поле «Адрес приемника» (RA)

Поле Адрес приемника (RA) содержит IEEE MAC индивидуальный или групповой адрес, который идентифицирует непосредственно STA, получающие информацию содержащуюся в поле тела фрейма, беспроводной среды (WM).

7.1.2.3.7 Поле «Адрес передатчика» (TA)

Поле Адрес передатчика (TA) содержит IEEE MAC индивидуальный адрес, который идентифицирует STA, которая передала в WM MPDU, содержащийся в поле тела фрейма. Бит индивидуального/группового адреса всегда передается как ноль.

7.1.2.4 Поле «Sequence Control»

Поле Sequence Control – длина 16 бит состоит из двух подполей, Номера Последовательности и Номера Фрагмента. Формат поля Sequence Control показан в Таблица 6.

Таблица 6. Поле Sequence Control

Поле	Длина (в битах)	Биты

Поле	Длина (в битах)	Биты
Fragment Number	4	0-3
Sequence Number	12	4-15

7.1.2.4.1 Поле «Sequence Number»

Номер Последовательности - 12-битное поле, содержащее номер последовательности MSDU или MMPDU. Каждому MSDU или MMPDU, переданному STA, назначается номер последовательности. Номера Последовательности назначаются по модулю 4096, начиная с 0 (ноль) и увеличивая на 1 (один) для каждого MSDU или MMPDU. Каждый фрагмент MSDU или MMPDU содержит назначенный номер последовательности. Номер последовательности остается постоянным во всех повторных передачах MSDU, MMPDU, или его фрагментов.

7.1.2.4.2 Поле «Fragment Number»

Номер Фрагмента - 4-битовое поле, содержащее номер каждого фрагмента MSDU или MMPDU. Номер фрагмента устанавливается в ноль в первом фрагменте MSDU или MMPDU и увеличивается на один для каждого последующего фрагмента этого MSDU или MMPDU. Номер фрагмента остается постоянным во всех повторных передачах фрагмента.

7.1.2.5 Поле «Frame Body» (Тело фрейма)

Тело фрейма - поле переменной длины и содержит информацию, определенную к индивидуально типом и подтипом фрейма. Минимальная длина тела фрейма - ноль байтов. Максимальная длина тела фрейма определена максимальной длиной (MSDU + ICV + IV); где ICV и IV - поля WEP, определенные в 8.2.5.

7.1.2.6 Поле FCS

Поле FCS - 32-битовое поле, содержащая 32-битовую CRC. FCS рассчитывается по всем полям заголовка MAC и поля Frame Body. Они упомянуты как поля вычисления.

FCS рассчитывается с использованием следующего стандартного полинома степени 32:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

FCS - дополнение суммы (по модулю 2) следующего:

a) Остаток от xk · (x31 + x30 + x29 +... + x2 + x + 1) разделенный (по модулю 2) на G (x), где k – число битов в полях вычисления, и
b) Остаток после умножения содержимого (трактуемый как полином) полей вычисления на x32 и, затем, деления на G (x).

Поле FCS передается, начиная с коэффициента самого высокого порядка.

Как типовое исполнение, в передатчике, начальный остаток от деления устанавливается в значение «все «1» и изменяется делением полей вычисления на полином G(x). Дополнение этого остатка передается начиная со старшего бита частицей как поле FCS.

В приемнике, начальный остаток устанавливается в значение «все «1», последовательным поступающие биты полей вычисления и FCS, делятся на G(x), и, при отсутствии ошибок передачи, результатом является уникальное, отличное от нуля, значении остатка. Уникальное значение остатка - полином:

$$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1.$$

7.2 *Форматы индивидуальных типов фреймов*

7.2.1 Фреймы контроля

Далее, описание «немедленно предыдущий» фрейм означает фрейм, чей прием закончился в пределах предшествующего короткого межфреймового интервала (SIFS) интервала.

Подполя в пределах поля Frame Control фрейма управления установлены как показано в Таблица 7.

Таблица 7. Значения подполей в пределах поля Frame Control фрейма управления

Поле	Значение	Длина (в битах)	Биты
Protocol Version	Protocol Version	2	0-1
Type	Control	2	2-3
Subtype	Subtype	4	4-7
To DS	0	1	8
From DS	0	1	9
More Frag	0	1	10
Retry	0	1	11
Pwr Mgt	Pwr Mgt	1	12
More Data	0	1	13
WEP	0	1	14
Order	0	1	15

7.2.1.1 Формат фрейма «Запрос передачи» (RTS)

Формат фрейма RTS показан в Таблица 8.

Таблица 8. Фрейм RTS

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
Duration	2	
RA	6	
TA	6	
FCS	4	

RA фрейма RTS - адрес STA, на WM, который является назначенным непосредственным получателем ожидаемых направленных данных или фрейма управления.

TA - адрес STA, передающей фрейм RTS.

Значение duration - время, в микросекундах, требуемых, чтобы передать ожидаемые данные или фрейм управления, плюс один фрейм CTS, плюс один фрейм ACK, плюс три SIFS интервала. Если расчетная продолжительность включает неполную микросекунду, то значение округляется до большего целого числа.

7.2.1.2 Формат фрейма «Готов к передаче» (CTS)

Формат фрейма CTS показан в Таблица 9.

Таблица 9. Фрейм CTS

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
Duration	2	
RA	6	
FCS	4	

RA фрейма CTS скопирован из поля TA немедленно предыдущего фрейма RTS, для которому CTS является ответом.

Значение duration - значение, полученное из поля Duration немедленно предыдущего фрейма RTS, минус время, в микросекундах, требуемых, чтобы передать фрейм CTS и его SIFS интервал. Если расчетная продолжительность включает фракционную неполную, то значение округляется до большего целого числа.

7.2.1.3 Формат фрейма «Подтверждение» (ACK)

Формат фрейма ACK показан в Таблица 10.

Таблица 10. Фрейм ACK

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
Duration	2	
RA	6	
FCS	4	

RA фрейма ACK скопирован из поля Address 2 немедленно предыдущих направленных данных, управления, или PS-Poll фрейма контроля.

Если бит More Fragment поля Frame Control немедленно предыдущих направленных данных или фрейма управления установлен в «0» в, значение duration установлено в «0». Если бит More Fragment поля Frame Control немедленно предыдущих направленных данных или фрейма управления установлен в «1», значение duration - значение, полученное из поля Duration немедленно предыдущих данных или фрейма управления, минус время, в микросекундах, требуемых, чтобы передать фрейм ACK и его SIFS интервал. Если расчетная продолжительность включает неполную микросекунду, то значение округляется до большего целого числа.

7.2.1.4 Формат фрейма «Опрос энергосбережения» (PS-Poll)

Формат фрейма PS-Poll показан в Таблица 11.

Таблица 11. Фрейм PS-Poll

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
AID	2	
BSS ID	6	
TA	6	
FCS	4	

BSSID - адрес STA, содержащейся в AP. TA - адрес STA передающей фрейм. AID - значение, назначенное на передачу фрейма STA AP во фрейме ответа ассоциации, который установил текущую ассоциацию STA.

В значении AID всегда 2 старших бита установлены в «1». Все STA, после получения PS-Poll фрейма, модифицируют свои установки NAV, поскольку в соответствии с правилами функции координации, использующими значение продолжительности равное времени, в микросекундах, требуемых, чтобы передать один фрейм ACK плюс один SIFS интервал.

7.2.1.5 Формат фрейма CF-End

Формат фрейма CF-End показан в Таблица 12.

Таблица 12. Фрейм CF-End

Поле	Длина (в байтах)	
Frame Control	2	Заголо-вок MAC
Duration	2	
RA	6	
BSS ID	6	
FCS	4	

BSSID - адрес STA, содержащейся в AP. RA - broadcast адрес группы.
Поле Duration установлено в «0».

7.2.1.6 Формат фрейма CF-End+CF-Ack

Формат фрейма CF-End+CF-Ack показан в Таблица 13.

Таблица 13. Фрейм CF-End+CF-Ack

Поле	Длина (в байтах)	
Frame Control	2	Заголо-вок MAC
Duration	2	
RA	6	
BSS ID	6	
FCS	4	

BSSID - адрес STA, содержащейся в AP. RA - broadcast адрес группы.
Поле Duration установлено в «0».

7.2.2 Фреймы данных

Формат фрейма данных, независимо от подтипа, представлен в Таблица 14.

Таблица 14. Фрейм данных

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
Duration / ID	2	
Address 1	6	
Address 2	6	
Address 3	6	
Sequence Control	2	
Address 4	6	
Frame Body	0 – 2312	
FCS	4	

Содержание полей Address фрейма данных зависит от значений битов To DS и From DS частиц и показано в Таблица 15. Если содержание поля показывается как N/A, поле опущено. Обратите внимание, что Address 1 всегда содержит адрес приемника назначенного адресата (или, в случае фреймов multicast, приемники), а Address 2 всегда содержит адрес станции, которая передает фрейм.

Таблица 15. Содержимое полей Address

ToDS	FromDS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSS ID	N/A
0	1	DA	BSS ID	SA	N/A
1	0	BSS ID	SA	DA	N/A
1	1	RA	TA	DA	SA

Станция использует значение поля Address 1 для сравнения адреса. Если поле Address 1 содержит адрес группы, используется BSSID, чтобы гарантировать что broadcast или multicast, находится в той же BSS.

Станция использует значение поля Address 2, чтобы отправить подтверждение, если оно необходимо.

DA - адресат MSDU (или его фрагмента) в поле тела фрейма.

SA - адрес объекта MAC, который ввел MSDU (или его фрагмент) в поле тела фрейма.

RA - адрес STA, содержащейся в AP беспроводной системы распределения, которая является следующим непосредственным назначенным получателем фрейма.

TA - адрес STA, содержащейся в AP беспроводной системы распределения, которая передает фрейм.

BSSID фрейма Данных определен следующим образом:

a) Если станция - AP или связана с AP, BSSID - адрес, используемый в настоящее время STA, содержащейся в AP.
b) Если станция - член IBSS, BSSID - BSSID IBSS.

Тело фрейма состоит из MSDU, или его фрагмента, и WEP IV и ICV (только если подполе WEP в поле управления фрейма, установлено в «1»). Тело фрейма пустой (длина ноль байтов) во фреймах данных подтипа Пустой функции (Null function (no data)), CF-ACK (no data), CF-Poll (no data), и CF-ACK+CF-Poll (no data).

В пределах всех фреймов данных, посланных в течение CFP, поле Duration установлено в значение 32768. В пределах всех фреймов типа данных, посланных в течение периода конкуренции, поле Duration установлено согласно следующим правилам:

- Если поле Address 1 содержит адрес группы, значение duration, установлено в «0».
- Если бит More Fragments в поле Frame Control установлен в «0» и поле Address 1 содержит индивидуальный адрес, значение duration установлено на время, в микросекундах, необходимое для передачи одного фрейма ACK, плюс один SIFS интервал.
- Если бит More Fragments в поле Frame Control установлен в «1» и поле Address 1 содержит индивидуальный адрес, значение duration установлено на время, в микросекундах, необходимое для передачи следующий фрагмента этого фрейма данных, плюс два фрейма ACK, плюс три SIFS интервала.

Вычисление значения duration для фрейма данных основано на правилах в 9.6, которые определяют скорость передачи данных, с которой передаются фреймы контроля последовательности обмена фреймами. Если расчетная продолжительность включает неполную микросекунду, то значение округляется до большего целого числа. Все станции обрабатывают поле Duration правильно полученных фреймов, оценивают меньшие или равные 32767 значения duration, для модификации их NAV в соответствии с правилами функции координации.

7.2.3 Фреймы управления

Формат фрейма управления, независимо от подтипа, представлен в Таблица 16.

Таблица 16. Фрейм управления

Поле	Длина (в байтах)	
Frame Control	2	Заголовок MAC
Duration	2	
DA	6	
SA	6	
BSS ID	6	
Sequence Control	2	
Frame Body	0 – 2312	
FCS	4	

STA использует содержимое поля Address 1 для сравнения адреса. Если поле Address 1 содержит адрес группы, и тип фрейма отличается от Beacon, используется BSSID, чтобы гарантировать что broadcast или multicast, находится в той же BSS. Если тип фрейма - Beacon, используется другой адрес, как определено в 11.1.2.3.

Поля адреса для фреймов управления не зависят от подтипа фрейма. BSSID фрейма управления определяется следующим образом:

a) Если станция - AP или связана с AP, BSSID - адрес, используемый в настоящее время STA, содержащейся в AP.
b) Если станция - член IBSS, BSSID - BSSID IBSS.
c) Во фреймах Управления подтип Probe Request, BSSID является или определенным BSSID, или broadcast BSSID согласно процедурам, описанным в Пункте 10.

DA - получатель фрейма.

SA - адрес станции, передающей фрейм.

В пределах всех фреймов управления, посланных в течение CFP, поле Duration установлено в значение 32768. В пределах всех фреймов управления, посланных в течение периода конкуренции, поле Duration устанавливается согласно следующим правилам:

- Если поле DA содержит адрес группы, значение duration, установлено в «0».
- Если бит More Fragments в поле Frame Control установлен в «0» и поле DA содержит индивидуальный адрес, значение duration установлено на время, в микросекундах, необходимое для передачи одного фрейма ACK, плюс один SIFS интервал.
- Если бит More Fragments в поле Frame Control установлен в «1» и поле DA содержит индивидуальный адрес, значение duration установлено на время, в микросекундах, необходимое для передачи следующий фрагмента этого фрейма данных, плюс два фрейма ACK, плюс три SIFS интервала.

Вычисление значения duration для фрейма управления основано на правилах в 9.6, которые определяют скорость передачи данных, с которой передаются фреймы контроля последовательности обмена фреймами. Если расчетная продолжительность включает неполную микросекунду, то значение округляется до большего целого числа. Все станции обрабатывают поле Duration правильно полученных фреймов, оценивают меньшие или равные 32767 значения duration, для модификации их NAV в соответствии с правилами функции координации.

Тело фрейма состоит из фиксированных полей и информационных элементов, определенных для каждого подтипа фрейма управления. Все фиксированных поля и информационные элементы обязательны если не заявлено иначе, и они могут появляться только в указанном порядке. Если станция находит типом элемента, который она не понимает, то элемент игнорируется. Коды типа элемента, явно не определенные в стандарте, зарезервированы, и не появляются ни в каких фреймах.

7.2.3.1 Формат фрейма «Маяк»

Тело фрейма управления, подтип «Маяк» содержит информацию, показанную в Таблица 17.

Таблица 17. Формат фрейма «Маяк»

Порядок	Информация	Примечание
1	Timestamp	—
2	Интервал Маяка	—
3	Информация способности	—
4	SSID	—
5	Поддерживаемые скорости	—
6	Набор параметра FH	1
7	Набор параметра DS	2
8	Набор параметра CF	3
9	Набор параметра IBSS	4
10	TIM	5

ПРИМЕЧАНИЯ:

1. Информационный элемент набора параметра FH присутствует только в пределах фреймов Маяка, произведенных STA с использованием PHY с прыгающей частотой.
2. Информационный элемент набора параметра DS присутствует только в пределах фреймов Маяка, произведенных STA с использованием прямой последовательности PHY.
3. Информационный элемент набора параметра CF присутствует только в пределах фреймов Маяка, произведенных AP, поддерживающими PCF.
4. Информационный элемент набора параметра IBSS присутствует только в пределах фреймов Маяка, произведенных STA в IBSS.
5. Элемент информации TIM присутствует только в пределах фреймов Маяка, произведенных AP.

7.2.3.2 Формат фрейма «IBSS Сообщение объявления признака трафика (ATIM)»

Тело фрейма управления подтипа «ATIM» пустое.

7.2.3.3 Формат фрейма «Дисассоциация»

Тело фрейма управления, подтип «Дисассоциация» содержит информацию, показанную в Таблица 18.

Таблица 18. Формат фрейма «Дисассоциация»

Порядок	Информация
1	Код причины

7.2.3.4 Формат фрейма «Запрос Ассоциации»

Тело фрейма управления, подтип «Запрос Ассоциации» содержит информацию, показанную в Таблица 19.

Таблица 19. Формат фрейма «Запрос Ассоциации»

Порядок	Информация
1	Информация способности
2	Интервал прослушивания
3	SSID
4	Поддерживаемые скорости

7.2.3.5 Формат фрейма «Ответ Ассоциации»

Тело фрейма управления, подтип «Ответ Ассоциации» содержит информацию, показанную в Таблица 20.

Таблица 20. Формат фрейма «Ответ Ассоциации»

Порядок	Информация
1	Информация способности
2	Код статуса
3	ID Ассоциации (AID)
4	Поддерживаемые скорости

7.2.3.6 Формат фрейма «Запрос Реассоциации»

Тело фрейма управления, подтип «Запроса Реассоциации» содержит информацию, показанную в Таблица 21.

Таблица 21. Формат фрейма «Запрос Реассоциации»

Порядок	Информация
1	Информация способности
2	Интервал прослушивания
3	Текущий адрес AP
4	SSID
5	Поддерживаемые скорости

7.2.3.7 Формат фрейма «Ответ Реассоциации»

Тело фрейма управления, подтип «Ответа Реассоциации» содержит информацию, показанную в Таблица 22.

Таблица 22. Формат фрейма «Ответ Реассоциации»

Порядок	Информация
1	Информация способности
2	Код статуса
3	ID Ассоциации (AID)
4	Поддерживаемые скорости

7.2.3.8 Формат фрейма «Запрос Доступа»

Тело фрейма управления, подтип «Запроса Доступа» содержит информацию, показанную в Таблица 23.

Таблица 23. Формат фрейма «Запрос Доступа»

Порядок	Информация
1	SSID
2	Поддерживаемые скорости

7.2.3.9 Формат фрейма «Ответ Доступа»

Тело фрейма управления, подтип «Ответ Доступа» содержит информацию, показанную в Таблица 24.

Таблица 24. Формат фрейма «Ответ Доступа»

Порядок	Информация	Примечание
1	Timestamp	—
2	Интервал Маяка	—
3	Информация способности	—
4	SSID	—
5	Поддерживаемые скорости	—
6	Набор параметра FH	1
7	Набор параметра DS	2
8	Набор параметра CF	3
9	Набор параметра IBSS	4

ПРИМЕЧАНИЯ:
1. Информационный элемент набора параметра FH присутствует только в пределах фреймов Маяка, произведенных STA с использованием PHY с прыгающей частотой.
2. Информационный элемент набора параметра DS присутствует только в пределах фреймов Маяка, произведенных STA с использованием прямой последовательности PHY.
3. Информационный элемент набора параметра CF присутствует только в пределах фреймов Маяка, произведенных AP, поддерживающими PCF.
4. Информационный элемент набора параметра IBSS присутствует только в пределах фреймов Маяка, произведенных STA в IBSS.

7.2.3.10 Формат фрейма «Аутентификация»

Тело фрейма управления, подтип «Аутентификация» содержит информацию, показанную в Таблица 25.

Таблица 25. Формат фрейма «Аутентификация»

Порядок	Информация	Примечание
1	Номер алгоритма аутентификации	
2	Номер транзакции аутентификационной последовательности	
3	Код статуса	1
4	Текст вызова	2

ПРИМЕЧАНИЯ:
1. Информация кода статуса зарезервирована и установлена в 0 в некоторых фреймах аутентификации как определено в Таблица

Порядок	Информация	Примечание
	26.	
2.	Информация текста вызова присутствует только в определенных фреймах аутентификации как определено в Таблица 26.	

Таблица 26. Наличие информации текста вызова

Номер алгоритма аутентификации	Номер транзакции аутентификационной последовательности	Код статуса	Текст вызова
Открытая система	1	Резерв	Отсутствует
Открытая система	2	Статус	Отсутствует
Общий ключ	1	Резерв	Отсутствует
Общий ключ	2	Статус	Присутствует
Общий ключ	3	Резерв	Присутствует
Общий ключ	4	Статус	Отсутствует

7.2.3.11 Формат фрейма «Деаутентификация»

Тело фрейма управления, подтип «Деаутентификация» содержит информацию, показанную в Таблица 27.

Таблица 27. Формат фрейма «Деаутентификация»

Порядок	Информация	Примечание
1	Код причины	

7.3 *Компоненты тела фрейма управления*

В пределах фреймов управления имеются обязательные компоненты фиксированной длины, определенные как поля фиксированной длины, и все дополнительные компоненты тела фрейма, определенные как информационные элементы.

7.3.1 Фиксированные поля

7.3.1.1 Поле «Номер алгоритма аутентификации»

Поле номера алгоритма аутентификации указывает отдельный алгоритм аутентификации. Длина поля номера алгоритма аутентификации - два октета. Поле номера алгоритма аутентификации иллюстрировано в Таблица 28. Для номера алгоритма аутентификации определены следующие значения:

Номер алгоритма аутентификации = 0: Открытая система;
Номер алгоритма аутентификации = 1: Общий ключ;
Все другие значения номера алгоритма аутентификации зарезервированы.

Таблица 28. Фиксированное поле номера алгоритма аутентификации

Поле	Длина (в байтах)	Биты
Номер алгоритма аутентификации	2	0-15

7.3.1.2 Поле «Номер транзакции аутентификационной последовательности»

Поле номера транзакции аутентификационной последовательности указывает текущее состояние процесса через многошаговую транзакцию. Длина поля номера транзакции аутентификационной после-

довательности - два октета. Поле номера транзакции аутентификационной последовательности показано в Таблица 29.

Таблица 29. Фиксированное поле номера алгоритма аутентификации

Поле	Длина (в байтах)	Биты
Номер транзакции аутентификационной последовательности	2	0-15

7.3.1.3 Поле «Интервал Маяка»

Поле «Интервал Маяка» представляет число единиц времени (TU) между целевыми временами передачи маяка (TBTT). Длина поля «Интервал Маяка» - два октета. Поле «Интервал Маяка» показано в Таблица 30.

Таблица 30. Фиксированное поле «Интервал Маяка»

Поле	Длина (в байтах)	Биты
Интервал Маяка	2	0-15

7.3.1.4 Поле «Информация возможностей»

Поле «Информация возможностей» содержит множество подполей, которые используются, чтобы указать требуемые или предоставляемые возможности.

Длина поля «Информация возможностей» - 2 октета. Поле «Информация возможностей» состоит из следующих подполей: ESS, IBSS, CF-Pollable, CF-Poll Request, и Privacy, Short Preamble, PBCC, и Channel Agility. Формат поля «Информация возможностей» показан в Таблица 31.

Таблица 31. Фиксированное поле «Информация возможностей»

Поле	Длина (в битах)	Биты
ESS	1	0
IBSS	1	1
CF-Pollable	1	2
CF-Poll Request	1	3
Privacy	1	4
Short Preamble	1	5
PBCC	1	6
Channel Agility	1	7
Резерв	8	8-15

Каждое подполе информации возможностей интерпретируется только в подтипах фрейма управления, для которых определены правила передачи.

AP устанавливают подполе ESS в 1 и подполе IBSS в 0 в пределах переданного фрейма управления подтипа «Маяк» или «Ответ пробы». STA в пределах IBSS устанавливают подполе ESS в 0 и подполе IBSS в 1 в переданном фрейме управления подтипа «Маяк» или «Ответ пробы».

STA устанавливают подполя CF-Pollable и CF-Poll Request в фреймах управления подтипа «Ассоциация» и «Запрос Реассоциации», согласно Таблица 32.

Таблица 32. Использование STA подполей CF-Pollable и CF-Poll Request

CF-Pollable	CF-PoII request	Значение
0	0	STA не CF-Pollable

CF-Pollable	CF-PoII request	Значение
0	1	STA CF-Pollable, не требующий помещения в список CF-Polling
1	0	STA CF-Pollable, требующий помещения в список CF-Polling
1	1	STA CF-Pollable, требующий, чтобы никогда не голосовать

AP устанавливают подполя CF-Pollable и CF-Poll Request во фреймах управления «Маяк», «Ответ доступа», «Ответ Ассоциации» и «Ответ реассоциации» согласно Таблица 33. AP устанавливает значения подполей CF-Pollable и CF-Poll Reques во фреймах управления «Ответ Ассоциации» и «Ответ Реассоциации» в значения, полученные в последнем фрейме «Маяк» или «Ответ доступа».

Таблица 33. Использование AP подполей CF-Pollable и CF-Poll Request

CF-Pollable	CF-PoII request	Значение
0	0	Координатор точки не в AP
0	1	Координатор точки в AP только для доставки (нет опроса)
1	0	Координатор точки не в AP для доставки и опроса
1	1	Reserved

AP устанавливают подполе Privacy в 1 в пределах передаваемых фреймов управления подтипа «Маяк», «Ответ доступа», «Ответ Ассоциации» и «Ответ Реассоциации», если требуется шифрование WEP для всех фреймов типа данных, которыми обмениваются в пределах BSS. Если WEP шифрование не требуется, подполе Privacy установлено в 0.

STA в пределах IBSS устанавливают подполе Privacy в 1 в передаваемых фреймах управления подтипа «Маяк» или «Ответ доступа», если шифрование WEP требуется для всех фреймов типа данных, которыми обмениваются в пределах IBSS. Если шифрование WEP не требуется, подполе Privacy установлено в 0.

AP (также как STA в IBSS) должны установить подполе Short Preamble в 1 в передаваемых MMPDU управления подтипа «Маяк», «Ответ доступа», «Ответ Ассоциации» и «Ответ реассоциации» для указания что используется опция Короткой Преамбулы, как описано в 18.2.2.2, допустимая в пределах этого BSS. Чтобы указывать, что использование опции Короткой Преамбулы не позволяется, подполе Short Preamble должно быть установлено в 0 в переданных в пределах BSS MMPDU управления «Маяк», «Ответ доступа», «Ответ Ассоциации» и «Ответ Реассоциации».

STA устанавливают подполе Short Preamble в 1 в переданных MMPDU подтипа «Запрос Ассоциации» и «Запрос Реассоциации», когда MIB атрибут dot11ShortPreambleOptionImplemented, истинен. Иначе, STA должны установить подполе Short Preamble в 0 в переданных MMPDU «Запрос Ассоциации» и «Запрос Реассоциации».

AP (также как STA в IBSS) должны установить подполе PBCC в 1 в переданных MMPDU управлении подтипа «Маяк», «Ответ доступа», «Ответ Ассоциации» и «Ответ Реассоциации» указывая, что позволяется использование опции PBCC Модуляции, как описано в 18.4.6.6, в пределах этого BSS. Чтобы указывать, что использование опции PBCC Модуляции не позволяется, подполе PBCC должно быть установлено в 0 в передаваемых в пределах BSS MMPDU управления подтипа «Маяк», «Ответе доступа», «Ответ Ассоциации» и «Ответ Реассоциации».

STA устанавливает подполе PBCC в 1 в передаваемых MMPDU подтипа «Запрос Ассоциации» и «Запросе Реассоциации», когда MIB атрибут dot11PBCCOptionImplemented, истинен. Иначе, STA должны установить подполе PBCC в 0 в передаваемых MMPDU подтипа «Запрос Ассоциации» и «Запрос Реассоциации».

Бит 7 поля Capabilities Information используется, для указания использования бита Channel Agility HR/DSSS PHY. STA устанавливают бит Channel Agility в 1, если Channel Agility он используется, и в 0 иначе.

Биты 8–15 поля «Информация возможностей» зарезервированы.

7.3.1.5 Поле «Текущий адрес AP»

Поле «Текущий адрес AP» является MAC адресом AP с которой станция в настоящее время ассоциирована. Длина поля «Текущий адрес AP» - шесть октетов. Поле «Текущий адрес AP» показано в Таблица 34.

Таблица 34. Фиксированное поле «Интервал Маяка»

Поле	Длина (в байтах)	Биты
Текущий адрес AP	6	0-47

7.3.1.6 Поле «Интервал прослушивания»

Поле Listen Interval используется, чтобы указать AP, как часто STA пробуждается, чтобы слушать фреймы управления подтипа «Маяк». Значение этого параметра - aListenlnterval MIB атрибут STA, выраженный в единицах Интервала Маяка. Длина поля Listen Interval - два октета. Поле Listen Interval показано в Таблица 35.

Таблица 35. Фиксированное поле «Интервал прослушивания»

Поле	Длина (в байтах)	Биты
Интервал прослушивания	2	0-15

AP может использовать информацию Интервала прослушивания при определении продолжительности жизни фреймов, которые буферированы в STA.

7.3.1.7 Поле «Код причины»

Поле Reason Code используется для указания причины появления незапрашиваемого фрейма управления уведомления типа «Дисассоциация» или «Деаутентификация». Длина поля Reason Code - два октета. Поле Reason Code показано в Таблица 36.

Таблица 36. Фиксированное поле «Код причины»

Поле	Длина (в байтах)	Биты
Код причины	2	0-15

Коды причины определены в Таблица 37.

Таблица 37. Коды причины

Код причины	Значение
0	Резерв
1	Неуказанная причина
2	Предыдущая аутентификация больше не имеет силы
3	Деаутентифицирована, т.к.посылающая станция покидает (покинула) IBSS или ESS
4	Дисассоциирована из-за неактивности
5	Дисассоциирована, т.к. AP неспособна обработать всеми в настоящее время ассоциированные станции
6	Фрейм класса 2 получен от неаутентифицированной станции
7	Фрейм класса 3 получен от неассоциированной станции
8	Дисассоциирована, т.к. посылающая станция покидает (покинула) BSH
9	Станция, требующая (ре)ассоциации не аутентифицирована с отвечающей станцией
10-65 535	Резерв

7.3.1.8 Поле «Association ID» (AID)

Поле AID - значение, назначенное AP в течение ассоциации и содержит 16 бит ID STA. Длина поля AID - два октета. Поле AID показано в Таблица 38.

Таблица 38. Фиксированное поле AID

Поле	Длина (в байтах)	Биты
Association ID (AID)	2	0-15

Значение Association ID находится в диапазоне 1-2007 и помещено в 14 младших бит поля AID, а 2 старших бита поля AID установлены в 1 (см. 7.1.2.3.2).

Значение AID 0 используется, чтобы объявить broadcast и multicast фреймы в информационных элементах карты индикации трафика.

7.3.1.9 Поле «Код статуса»

Поле Status Code используется во фрейме управления «ответ» для указания успеха или неудачи требуемого действия. Длина поля Status Code - два октета. Поле Status Code показано в Таблица 39.

Таблица 39. Фиксированное поле «Код статуса»

Поле	Длина (в байтах)	Биты
Код статуса	2	0-15

Если действие выполнено, код статуса установлен в 0. Иначе, кодекс статуса указывает причину неудачи. Коды причины неудачи определены в Таблица 40.

Таблица 40. Коды статуса

Код статуса	Значение
0	Успешно
1	Неуказанная неудача
2-9	Резерв
10	Не может поддерживать все требуемые способности в поле Capability Information
11	Реассоциация отвергнута из-за неспособности подтвердить, что ассоциация существует
12	Ассоциация отвергнута из-за нахождения вне возможностей этого стандарта
13	Отвечающая станция не поддерживает указанный алгоритм аутентификации
14	Получен фрейм аутентификации с номером аутентификации не входящим в ожидаемую последовательность
15	Аутентификация отклонена из-за неудачи вызова
16	Аутентификация отклонена из-за перерыва для ожидания следующего фрейма в последовательности
17	Ассоциация отвергнута, т.к. AP неспособна обработать дополнительные связанные станции
18	Ассоциация отвергнута из-за отсутствия поддержки запрашивающей станцией всех скоростей передачи данных в параметре BSSBasicRateSet
19	Ассоциация отвергнута из-за отсутствия поддержки запрашивающей станцией опции Короткой Преамбулы.
20	Ассоциация отвергнута из-за отсутствия поддержки запрашивающей станцией опции PBCC Модуляции.

Код статуса	Значение
21	Ассоциация отвергнута из-за отсутствия поддержки запрашивающей станцией опции Channel Agility.
22-65535	Резерв

7.3.1.10 Timestamp

Это поле содержит значение TSFTIMER (см. 11.1) источника фрейма. Длина поля Timestamp - восемь октетов. Поле Timestamp показано в Таблица 41.

Таблица 41. Фиксированное поле «Timestamp»

Поле	Длина (в байтах)	Биты
Timestamp	8	0-63

7.3.2 Информационные элементы

Все элементы имеют определенный общий формат, состоящий из однобайтового поля Element ID, однобайтового поля длины, и элементно-зависимого информационного поля переменной длины. Каждому элементу назначен уникальный Element ID как определено в этой спецификации. Поле длины определяет число октетов в поле информации (См. Таблица 42).

Таблица 42. Формат элемента

Поле	Длина (в байтах)
Element ID	1
Длина	1
Информация	Длина

Набор имеющих силу элементов определен в Таблица 43.

Таблица 43. Значения поля Element ID

Информационный элемент	Element ID
SSID	0
Поддерживаемые скорости	1
Набор параметров FH	2
Набор параметров DS	3
Набор параметров CF	4
TIM	5
Набор параметров IBSS	6
Резерв	7-15
Текст вызова	16
Зарезервировано для расширения текста вызова	17-31
Резерв	32-255

7.3.2.1 Элемент службы установки идентичности (SSID)

Элемент службы установки идентичности (SSID) показывает идентичность расширенного набора служб (ESS) или IBSS. (См. Таблица 44).

Таблица 44. Формат элемента SSID

Поле	Длина (в байтах)

Поле	Длина (в байтах)
Element ID	1
Длина	1
SSID	0 - 32

Длина информационного поля SSID – от 0 до 32 октетов. Поле информации нулевой длины указывает broadcast SSID.

7.3.2.2 Элемент «Поддерживаемые скорости»

Элемент «Поддерживаемые скорости» определяет все скорости, которые эта STA способна получить. Информационное поле имеет длину от 1 до 8 октетов, где каждый октет описывает отдельную поддерживаемую скорость в единицах 500 kbit/s.

В пределах фреймов управления «Маяк», «Ответ Доступа», «Ответ Ассоциации» и «Ответ Реассоциации» каждая поддерживаемая скорость, принадлежащая к BSSBasicRateSet, как определенно в 10,3.10.1, кодируется как октет со старшим битом (бит 7) установленным в 1 (например, скорость 1 Mbit/s, принадлежащая к BSSBasicRateSet кодируется как X '82'}. Скорости, не принадлежащие к BSSBasicRateSet кодируются со старшим битом в 0 (например, скорость 2 Mbit/s, не принадлежащая BSSBasicRateSet кодируется как X '04'). Сташий бит каждого октета поддерживаемой скорости в других типах фрейма управления игнорируется при получении STA.

BSSBasicRateSet информация в фреймах управления «Маяк» и «Ответа Пробы» используется STA чтобы избегать связи с BSS, если они не поддерживают все данные скорости в BSSBasicRateSet. (См. Таблица 45).

Таблица 45. Формат элемента «Поддерживаемые скорости»

Поле	Длина (в байтах)
Element ID	1
Длина	1
Поддерживаемые скорости	1 – 8

7.3.2.3 Элемент «Набор параметров FH»

Элемент «Набор параметров FH», содержит набор параметров, необходимых для синхронизации STA использующих PHY с прыгающего частотой (FH). Информационное поле содержит параметры «Выдержка времени», «Набор прыжков», «Шаблон прыжков» и «Индекс прыжков». Полная длина информационного поля - 5 октетов. (См. Таблица 46).

Таблица 46. Формат элемента «Набор параметров FH»

Поле	Длина (в байтах)
Element ID	1
Длина	1
Выдержка времени (TU)	2
Набор прыжков	1
Шаблон прыжков	1
Индекс прыжков	1

Поле «Выдержка времени» - длина два октета и содержит выдержку времени в TU.

Поле «Набор прыжков» идентифицирует специфический набор шаблонов прыжков и имеет длину один октет.

Поле «Шаблон прыжков» идентифицирует матрицу индивидуальных шаблонов в пределах набора шаблонов прыжков и имеет длину один октет.

Поле «Индекс прыжков» выбирает текущий индекс канала в пределах шаблона и имеет длину один октет.

7.3.2.4 Элемент «Набор параметров DS»

Элемент «Набор параметров DS» содержит информацию для идентификации номера канала для STA с использованием PHY с направленной последовательностью спектра распространения (DSSS). Информационное поле содержит единственный параметр, содержащий текущий номер канала. Длина текущего параметра номера канала - один октет. (См. Таблица 47).

Таблица 47. Формат элемента «Набор параметров DS»

Поле	Длина (в байтах)
Element ID	1
Длина	1
Текущий канал	1

7.3.2.5 Элемент «Набор параметров CF»

Элемент «Набор параметров CF» содержит набор параметров, необходимых для поддержки PCF. Информационное поле содержит поля CFPCount, CFPPeriod, CFPMaxDuration и CFPDurRemaining. Полная длина информационного поля - 6 октетов. (См. Таблица 48).

Таблица 48. Формат элемента «Набор параметров CF»

Поле	Длина (в байтах)
Element ID	1
Длина	1
CFP Count	1
CFP Period	1
CFP MaxDuration (TU)	2
CFP DurRemaining (TU)	2

CFPCount указывает, сколько DTIM (включая текущий фрейм) появится перед следующим началом CFP. CFPCounl 0 указывает, что поток DTIM отмечает начало CFP.

CFPPeriod указывает число DTIM интервалов между началом CFP. Значение - целое число DTIM интервалов.

CFPMaxDuration указывает максимальную продолжительность, в TU, CFP, который может быть произведен этим PCF. Это значение используется STA, чтобы установить их NAV в TBTT маяков, которые начинают CFP.

CFPDurRemaining указывает максимальное время, в TU, остающемся в текущем CFP, и установлен в ноль в элементах «CFP Параметр» маяков, переданных в течение периода утверждения. Значение CFPDurRemaining ссылается на немедленно предыдущий TBTT. Это значение используется всем STA, чтобы модернизировать их NAV в течение CFP.

7.3.2.6 Элемент TIM

Элемент TIM содержит четыре поля: DTIM Count, DTIM Period, Bitmap Control и Partial Virtual Bitmap. (См. Таблица 49).

Таблица 49. Формат элемента TIM

Поле	Длина (в байтах)
Element ID	1

Поле	Длина (в байтах)
Длина	1
DTIM Count	1
DTIM Period	1
Bitmap Control	1
Partial Virtual Bitmap	1 - 251

Поле длины для этого элемента указывает длину информации - поля, которое ограничено как описано ниже.

Поле DTIM Count указывает, сколько маяков (включая текущий фрейм) появятся перед следующим DTIM. DTIM Count 0 указывает, что текущий TIM - DTIM. Длина поля DTIM Count - один октет.

Поле DTIM Period указывает число интервалов Маяка между последовательными DTIM. Если все TIM - DTIM, поле DTIM Period имеет значение 1. Значение поля DTIM Period 0 зарезервировано. Длина поля DTIM Period - один октет.

Поле Bitmap Control - один октет, младший бит содержит бит ассоциированный с Association ID 0. Этот бит установлен в 1 в элементах TIM со значением 0 в поле DTIM Count, когда один или более broadcast или multicast фреймы - буферированы в AP. Старший бит 7 формирует подполе Bitmap Offset. Подполе Bitmap Offset - число между 0 и 250, сформированное с использованием поля Bitmap Control с младшим битом, установленным в 0, и обсуждается ниже.

Traffic-indication virtual bitmap, поддержанный AP, который производит TIM, состоит из 2008 b, и организован в 251 октет такой, что номер бита H ($0 < N < 2007$) в bitmap соответствует номеру бита (N mod 8) в октете номер $\lfloor N / 8 \rfloor$, где младший бит каждого октета – бит с номером 0, и бит высокого порядка станции в пределах BSS, AP которой подготовлен, чтобы поставить во время, когда передается фрейм маяка. Номер бита N - 0, если нет никаких направленных буферированных фреймов для станции, чей Association ID - N. Если любые направленные фреймы для той станции - буферированы, и AP подготовлен, чтобы поставить им, номер бита N в traffic-indication virtual bitmap - 1. PC может отказаться устанавливать биты в TIM для CF-Pollable станций и не намеревается голосовать (см. 11.2.1.5).

Поле Partial Virtual bitmap состоит из октетов, с N1 по N2 traffic-indication virtual bitmap, где N1 - самый большой номер такой, что биты с 1 по (N1 x 8} = 1 в bitmap - все 0, и N2 - самый маленький номер такой, что биты (N2 + 1) x 8 по 2007 в bitmap = 0. В этом случае, значение подполя Bitmap Offset содержит номер N1, и поле длины будет установлено в (N2-N1) + 4.

Когда все биты, отличные от бита 0 в virtual bitmap является 0, поле Partial Virtual bitmap кодируется как единственный октет равняется 0, и подполе Bitmap Offset - 0.

7.3.2.7 Элемент «Набор параметров IBSS»

Элемент «Набор параметров IBSS» содержит набор параметров, необходимых для поддержки IBSS. Информационное поле содержит параметр «Окно ATIM». (См. Таблица 50).

Таблица 50. Формат элемента «Набор параметров IBSS»

Поле	Длина (в байтах)
Element ID	1
Длина	1
Окно ATIM	2

Поле «Окно ATIM» - длина 2 октета и содержит длину Окна ATIM в TU**.**

7.3.2.8 Элемент «Текст Вызова»

Элемент «Текст Вызова» содержит текст вызова в пределах обменов аутентификации. Длина поля информации элемента зависит от: алгоритма аутентификации и последовательного номера транзакции как определено в 8.1. (См. Таблица 51).

Таблица 51. Формат элемента «Текст Вызова»

Поле	Длина (в байтах)
Element ID	1
Длина	1
Текст Вызова	1 - 253

8 Аутентификация и секретность

8.1 *Службы аутентификации*

IEEE 802.11 определяет два подтипа служб аутентификации: Открытая Система и Общий Ключ. Вызываемый подтип указывается в теле фреймов управления аутентификацией. Таким образом, фреймы аутентификации являются самоидентифицирующимися относительно алгоритма аутентификации. Все фреймы управления с подтипом Authentication должны быть уникально направленными, так как аутентификация осуществляется между парой станций (т.е., multicast аутентификация не разрешена). Фреймы управления с подтипом Deauthentication являются уведомительными и могут посылаться с групповым адресом.

После успешного аутентификационного обмена, описанного ниже, между двумя станциями должна образоваться общая аутентификационная связь. Аутентификация должна использоваться между станциями и AP в инфраструктурных BSS. Аутентификация может использоваться между двумя STA в IBSS.

8.1.1 Аутентификация «Открытая Система»

Аутентификация «Открытая Система» является простейшим из доступных алгоритмов аутентификации. По существу это нулевой алгоритм аутентификации. Любая STA, которая запрашивает аутентификацию по этому алгоритму, может стать аутентифицированной, если aAuthenticationType приемной станции установлен в аутентификацию «Открытая Система». Аутентификация «Открытая Система» не обязательно должна быть успешной, так как STA может перейти к аутентификации с любой другой STA. Аутентификация «Открытая Система» используется в качестве алгоритма по умолчанию.

Аутентификация «Открытая Система» включает двухшаговую последовательность обмена. Первым шагом является заявление идентификатора и запрос аутентификации. Второй фрейм является результатом аутентификации. Если результат равен “successful”, STA должны быть взаимно аутентифицированы.

8.1.1.1 Аутентификация «Открытая Система» (первый фрейм)

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации = “Open System”
  - Идентификатор станции (в поле SA заголовка)
  - Последовательный номер процедуры аутентификации = 1
  - Информация, зависящая от алгоритма аутентификации (нет)
- Направление сообщения: от STA, инициирующей аутентификацию

8.1.1.2 Аутентификация «Открытая Система» (последний фрейм)

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации = “Open System”
  - Последовательный номер процедуры аутентификации = 2
  - Информация, зависящая от алгоритма аутентификации (нет)
  - Результат запрошенной аутентификации, как определено в 7.3.1.9
- Направление сообщения: от аутентифицирующей STA к инициирующей STA

Если aAuthenticationType не включает значение “Open System”, результирующий код не должен иметь значение “successful”.

8.1.2 Аутентификация «Общий Ключ»

Аутентификация «Общий Ключ» обеспечивает аутентификацию тех STA, которые либо знают общий секретный ключ, либо нет. Это достигается без необходимости передавать секретный ключ в явном ви-

де; требуется использование механизма секретности WEP. Таким образом, данная схема аутентификации доступна только в том случае, если реализована опция WEP. Кроме того, алгоритм аутентификации «Общий Ключ» должен быть реализован как один из aAuthenticationAlgoritms на любой STA, имеющей реализованный WEP.

Предполагается, что необходимый секретный общий ключ доставляется на STA по секретному каналу, не зависящему от IEEE 802.11. Этот общий ключ сохраняется в атрибуте MIB, предназначенном только для записи, по каналам менеджмента MAC. Атрибут может только записываться, поэтому значение ключа остается внутри MAC.

В течение обмена «Общий Ключ» передаются вызов и шифрованный вызов. При этом затрудняется нежелательное распознавание PRN (псевдослучайной) последовательности для пары ключ/IV, используемой в обмене. Таким образом, реализация данного алгоритма требует избегать использования одинаковой пары ключ/IV для последовательных фреймов.

STA не должна начинать аутентификационный обмен «Общий Ключ» до тех пор, пока ее атрибут aPrivacyOptionImplemented не станет равным “true”.

В приведенных ниже описаниях STA, инициирующая аутентификационный обмен, обозначена как requester, а STA, которой адресован начальный фрейм обмена, – как responder.

8.1.2.1 Аутентификация «Общий Ключ» (первый фрейм)

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Объявление идентификатора станции (в поле SA заголовка)
  - Идентификация алгоритма аутентификации = “Shared Key ”
  - Последовательный номер процедуры аутентификации = 1
  - Информация, зависящая от алгоритма аутентификации (нет)
- Направление сообщения: от requester к responder

8.1.2.2 Аутентификация «Общий Ключ» (второй фрейм)

Перед посылкой второго фрейма в последовательности «Общий Ключ» responder должен использовать WEP для генерации байтовой строки, которая будет использоваться в качестве текста вызова аутентификации.

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации = “Shared Key ”
  - Последовательный номер процедуры аутентификации = 2
  - Информация, зависящая от алгоритма аутентификации = результат аутентификации. Результат запрошенной аутентификации, как определено в 7.3.1.9.
    Если код состояния не равен “successful”, то это должен быть последний фрейм в последовательности. Если код состояния не равен “successful”, содержимое поля текста вызова не определено.
    Если код состояния равен “successful”, то следующие дополнительные информационные пункты должны иметь действительное содержание:
    Информация, зависящая от алгоритма аутентификации = текст вызова.
    Данное поле должно иметь фиксированную длину 128 байт. Поле должно быть заполнено байтами, сгенерированными WEP PRNG. Реальное значение поля вызова не важно, поскольку оно не должно иметь одиночный статический характер. Ключ и вектор инициализации (IV), использованные для генерации текста вызова, не определены, так как это значение ключ/IV не является общим и не предоставляет возможности взаимодействия.
- Направление сообщения: от responder к requester

8.1.2.3 Аутентификация «Общий Ключ» (третий фрейм)

Requester должен скопировать текст вызова из второго фрейма в третий фрейм. Третий фрейм должен быть передан после шифрования в WEP, как определено в 8.2.3, с помощью общего секретного ключа.

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации = “Shared Key ”
  - Последовательный номер процедуры аутентификации = 3
  - Информация, зависящая от алгоритма аутентификации = текст вызова из второго фрейма последовательности
- Направление сообщения: от requester к responder

Данный фрейм должен быть зашифрован, как описано ниже.

8.1.2.4 Аутентификация «Общий Ключ» (последний фрейм)

Responder должен попытаться дешифрировать содержимое третьего фрейма аутентификационной последовательности, как описано ниже. Если проверка WEP ICV является успешной, responder должен сравнить содержимое дешифрированного поля Challenge Text с текстом вызова, который он посылал во втором фрейме последовательности. Если они одинаковы, responder должен ответить успешным кодом состояния в четвертом фрейме последовательности. Если проверка WEP ICV является неуспешной, responder должен ответить успешным кодом состояния в четвертом фрейме последовательности, как описано ниже.

- Тип сообщения: менеджмент
- Подтип сообщения: аутентификация
- Информационные пункты:
  - Идентификация алгоритма аутентификации = “Shared Key ”
  - Последовательный номер процедуры аутентификации = 4
  - Информация, зависящая от алгоритма аутентификации = результат аутентификации.
    Результат запрошенной аутентификации.
    Это пункт с фиксированной длиной и значением “successful” или “unsuccessful”.
- Направление сообщения: от responder к requester

8.2 *Алгоритм WEP (секретность, эквивалентная проводу)*

8.2.1 Введение

Подслушивание является общей проблемой для пользователей любых типов беспроводных систем связи. IEEE 802.11 определяет алгоритм конфиденциальности, эквивалентный проводным LAN. WEP определяется как защита авторизованных пользователей беспроводных LAN от случайного подслушивания. Конфиденциальность данных зависит от внешней службы управления ключом, которая распределяет ключи шифрования/дешифрирования. Комитет стандартов IEEE 802.11 настоятельно рекомендует не использовать способ работы с секретностью, но без аутентификации. Хотя такая комбинация возможна, она все же оставляет систему открытой для несанкционированного доступа.

8.2.2 Особенности алгоритма WEP

Алгоритм WEP имеет следующие особенности:

- *Он является достаточно строгим*. Секретность, обеспечиваемая данным алгоритмом, основана на трудности распознавания секретного ключа с помощью обычной лобовой атаки. Это предположение, в свою очередь, базируется на длине секретного ключа и частоте смены ключей. WEP позволяет изменять ключ и частоту изменения IV.
- *Он является самосинхронизированным*. WEP самосинхронизирован для каждого сообщения. Эта особенность является критичной для алгоритма шифрования связного уровня данных, где пред-

полагается доставка с «наилучшим усилием», и скорость потери пакетов может быть весьма высокой.

- *Он является эффективным*. Алгоритм WEP является эффективным и может быть реализован аппаратным или программным способом.
- *Он может быть экспортируемым*. Предприняты все возможные усилия для того, чтобы увеличить шансы на то, что система с реализованным WEP могла бы быть экспортирована за пределы США по соответствующему разрешению Министерства Торговли. Тем не менее, это не гарантируется.
- *Он является не обязательным*. Реализация и использование WEP является необязательной в IEEE 802.11.

8.2.3 Теория работы WEP

Шифрованием называется процесс маскировки (двоичных) данных, предназначенный для того, чтобы спрятать содержащуюся в них информацию (см. [B4]). Данные, которые не зашифрованы, называются открытым текстом (обозначены как P), а данные, которые зашифрованы, называются зашифрованным текстом (обозначены как C). Процесс преобразования зашифрованного текста обратно в открытый называется дешифрированием. Алгоритм криптографии, или шифрования, – это математическая функция, используемая для шифрования и дешифрирования данных. Функция шифрования E обрабатывает P, чтобы получить C:

$$E_k(P) = C$$

При обратной процедуре функция дешифрирования D обрабатывает C, чтобы получить P:

$$D_k(C) = P$$

Как показано на Рис. 12, при использовании одинакового ключа для шифрования и дешифрирования получаем:

$$D_k(E_k(P)) = P$$

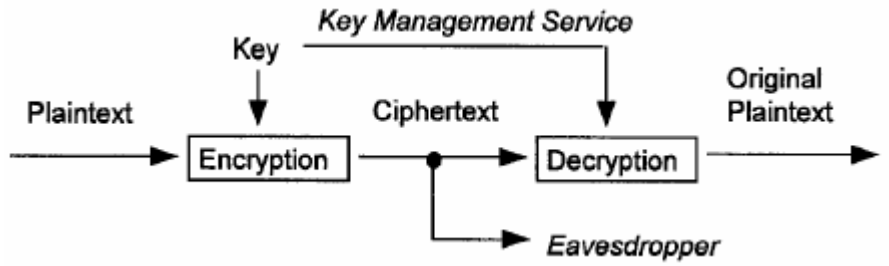


Рис. 12. Канал секретных данных.

Алгоритм WEP является одним из видов электронной кодовой книги, в которой блок открытого текста побитно складывается исключающим ИЛИ с псевдослучайной ключевой последовательностью одинаковой длины. Ключевая последовательность генерируется алгоритмом WEP.

Рассмотрим Рис. 13 слева направо; шифрование начинается с того, что секретный ключ распределяется на взаимодействующие STA внешней службой управления. WEP является симметричным алгоритмом, в котором один и тот же ключ используется для шифрования и дешифрирования.

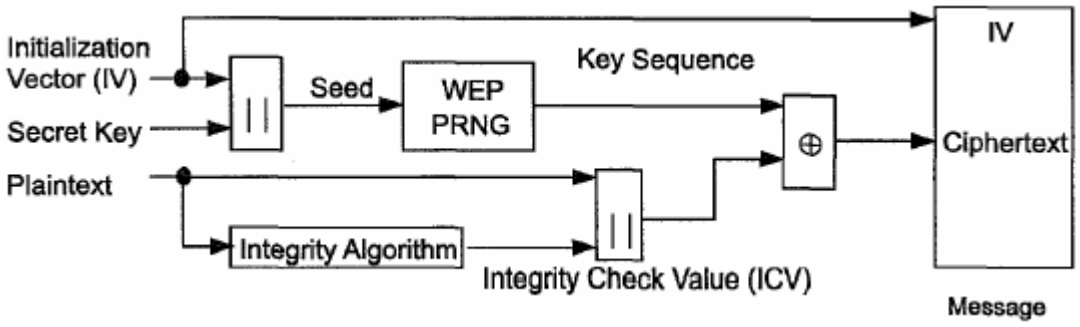


Рис. 13. Блок-схема шифрования WEP.

Секретный ключ объединяется с вектором инициализации (IV), а результирующий seed является входным значением для генератора псевдослучайных чисел (PRNG). На выходе PRNG получается ключевая последовательность из k псевдослучайных байт, равных по длине количеству байт данных, которые должны быть переданы в расширенных MPDU, плюс 4 (поскольку ключевая последовательность также используется для защиты интегрального проверочного числа (ICV)). Открытым MPDU назначены два процесса. Для защиты от несанкционированной модификации данных алгоритм обрабатывает P для получения ICV. Затем выполняется шифрование путем математического комбинирования ключевой последовательности с открытым текстом, объединенным с ICV. Результатом процесса является сообщение, содержащее IV и зашифрованный текст.

WEP PRNG является критичным компонентом данного процесса, поскольку он преобразовывает относительно короткий секретный ключ в ключевую последовательность произвольной длины. При этом сильно упрощается задача распределения ключа, так как только он нужен для связи между STA. IV увеличивает полезное время жизни секретного ключа и обеспечивает свойство самосинхронизации алгоритма. Секретный ключ остается постоянным, в то время как IV периодически изменяется. Каждый новый IV дает в результате новое seed и новую ключевую последовательность, так что существует однозначное соответствие между IV и k. IV может изменяться так же часто, как и сами MPDU, и, поскольку он передается вместе с сообщением, приемник всегда может дешифрировать любое сообщение. IV передается в явном виде, поскольку в нем не содержится какой-либо информации о секретном ключе, и поскольку его значение должно быть известно приемнику для того, чтобы выполнить дешифрирование. При выборе частоты изменения IV разработчик должен принимать во внимание то, что содержимое некоторых полей в заголовках протокола верхнего уровня, а также некоторая другая информация, являются постоянными либо легко предсказуемыми. При подслушивании передачи такой информации, зашифрованной через пару (ключ, IV), можно достаточно легко определить часть ключевой последовательности, сгенерированной с помощью этой пары. Если та же пара используется для шифрования последующих MPDU, то степень секретности, обеспечиваемая алгоритмом WEP, может значительно уменьшиться, а подслушивающее устройство сможет восстановить данные без знания секретного ключа. Изменение IV для каждых новых MPDU является простейшим способом сохранения эффективности WEP в такой ситуации.

Алгоритм WEP обрабатывает тело фрейма MPDU.

Для WEP-защищенных фреймов первые четыре байта из тела фрейма содержат поле IV для данных MPDU. Это поле определено в 8.2.5. 64-битовое seed PRNG формируется, используя секретный ключ как 40 старших значащих бит и IV как 24 младших значащих бита. За IV следуют MPDU, за которыми следует ICV. WEP ICV имеет длину 32 бита. Алгоритм проверки целостности WEP – это CRC-32, как определено в 7.1.2.6.

Как утверждалось ранее, WEP комбинирует k с P, используя побитовое исключающее ИЛИ.

Рассмотрим Рис. 14 слева направо; дешифрирование начинается с принятия сообщения. IV входящего сообщения должен использоваться для генерации ключевой последовательности, необходимой для дешифрирования входящего сообщения. Комбинирование зашифрованного текста с правильной ключевой последовательностью дает в результате оригинальный открытый текст и ICV. Правильность дешифрирования должна проверяться с помощью алгоритма проверки целостности, для этого результирующий ICV’ сравнивается с присланным ICV. Если ICV’ не равен ICV, то принятые MPDU являются ошибоч-

ными, и на менеджмент MAC должна быть послана индикация об ошибке. MSDU с ошибочными MPDU (из-за невозможности дешифрирования) не должны передаваться на LLC.

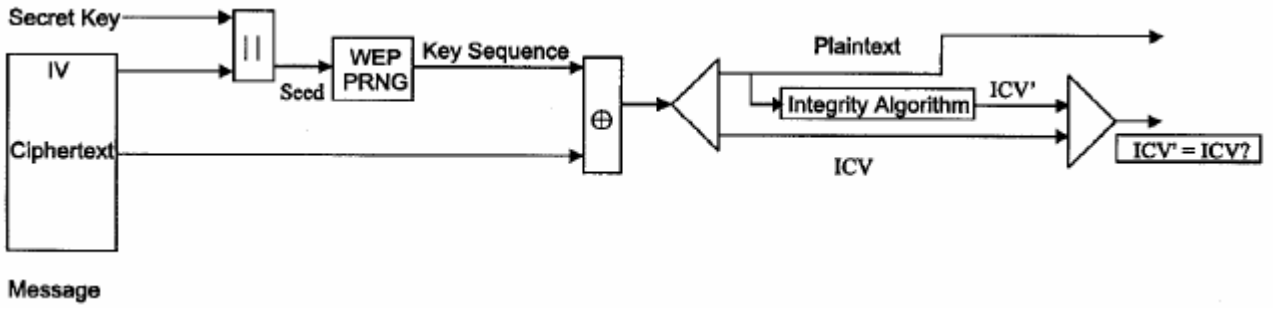


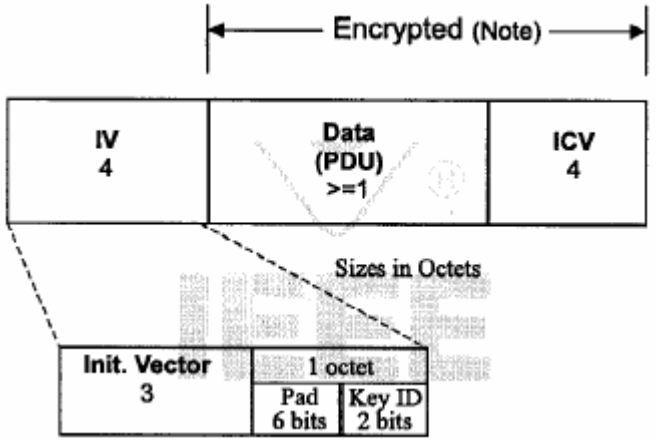
Рис. 14. Блок-схема дешифрирования WEP.

8.2.4 Спецификация алгоритма WEP

WEP использует алгоритм RC4 PRNG из RSA Data Security, Inc.[1]

8.2.5 WEP расширение MPDU

На Рис. 15 показано конструирование зашифрованных MPDU как часть алгоритма WEP.



Примечание: процесс шифрования расширяет оригинальные MPDU на 8 байт, 4 для поля вектора инициализации (IV) и 4 для проверочного значения идентичности (ICV). ICV вычисляется только по полю Data.

Рис. 15. Конструирование WEP расширенных MPDU.

ICV должен быть 32-битовым полем, содержащим 32-битовый CRC, определенный в 7.1.2.6, который вычисляется над полем Data (PDU). Расширенные MPDU должны включать 32-битовое поле IV, предшествующее MPDU. Это поле должно содержать три подполя: трехбайтовое поле, которое содержит вектор инициализации, 2-битовое поле key ID и 6-битовое поле заполнения. Порядок следования полей определен в 7.1.1. Подполе key ID содержит одно из четырех возможных значений секретного ключа, используемого для дешифрирования данных MPDU. Интерпретация этих бит определяется позже в 8.3.2. Подполе заполнения должно быть заполнено нулями. Key ID занимает два младших значащих бита последнего байта IV.

8.3 *Атрибуты MIB, связанные с безопасностью*

[1] Детали алгоритма RC4 доступны в RSA. Получить его можно по соответствующему лицензионному соглашению.

9 Функциональное описание подуровня MAC

В этом пункте представлено функциональное описание MAC. Архитектура подуровня MAC, включая распределенную функцию координации (DCF), функция координации точки (PCF), и их сосуществование в IEEE 802.11 LAN, представленная в 9.1. Эти функции подробно рассмотрены в 9.2 и 9.3, где обеспечивается полное функциональное описание каждой функции. Фрагментация и дефрагментация охвачены в 9.4 и 9.5. Поддержка разных скоростей описана в 9.6. Допустимые последовательности обмена фреймами - в 9.7. Наконец, множество дополнительных ограничений, чтобы ограничить случаи, в которых MSDUs перенаправляется или отбрасывается, описаны в 9.8.

9.1 *Архитектура MAC*

Архитектура MAC может быть описана, как показано на Рис. 16 как обеспечение PCF через услуги DCF.

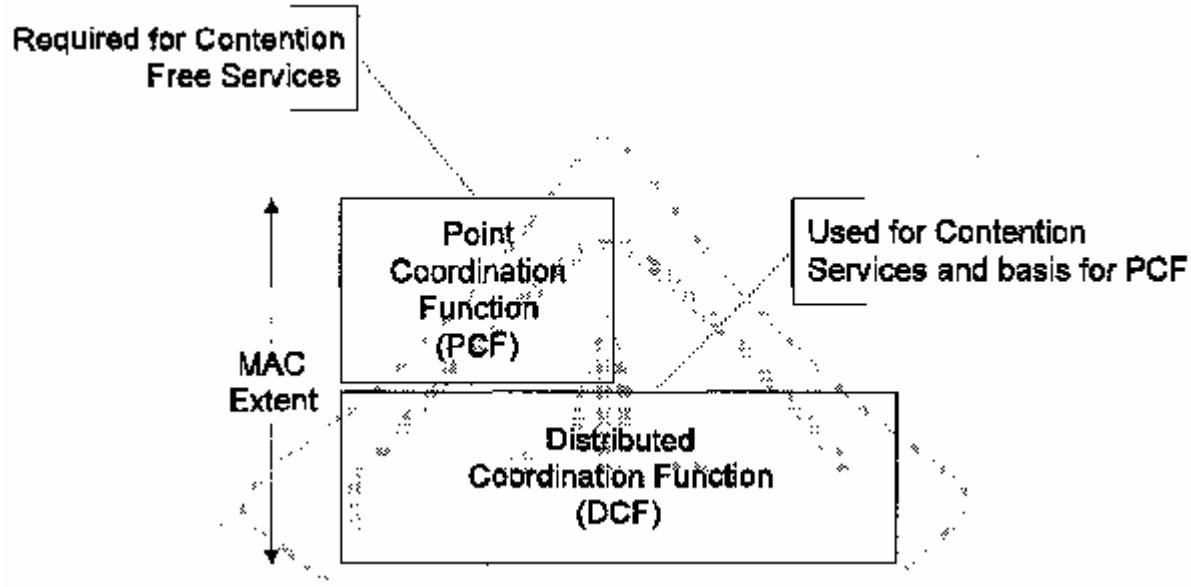


Рис. 16. Архитектура MAC.

9.1.1 Функция распределенной координации (DCF)

Фундаментальный метод доступа IEEE 80211 MAC - DCF известный как *чувствительный к несущей множественный доступ с предотвращением столкновения* или CSMA/CA. DCF должен быть реализован во всех STA, использующихся в пределах конфигураций IBSS и сети инфраструктуры.

При передаче STA должна проверить среду, чтобы определить, передает ли другая STA. Если среда не занята (см. 9.2.1), можно передавать. CSMA/CA распределяет полномочия алгоритма так, что существует минимальный промежуток времени между смежными последовательностями фреймов. Передающая STA, перед попыткой передать, должна гарантировать, что среда будет свободна требуемого количества времени. Если определяется, что среда занята, STA должна ожидать до конца текущей передачи. После ожидания или предыдущей попытки передачи сразу после успешной передачи, STA должна выбрать случайный интервал ожидания и декрементировать счетчик интервала ожидания во время простоя среды. Обработка или метод могут использоваться при различных обстоятельствах, чтобы далее минимизировать столкновения - здесь передающая и принимающая STA обмениваются короткими фреймами контроля [фреймы готов к передаче (RTS) и готов к приему (CTS)] после определения, что среда является свободной и после любых отсрочек или ожиданий до передачи данных. Детали CSMA/CA, отсрочек и ожиданий описаны в 9.2. RTS/CTS обмены также представлены в 9.2.

9.1.2 Функция координации точки (PCF)

IEEE 802.11 MAC может также включать дополнительный метод доступа называемый PCF, который применяется на конфигурациях сети инфраструктуры. Этот метод доступа использует координатор точки (PC), который должен работать в точке доступа BSS и определять, какая STA в настоящее время имеет право передавать. По существу – это опрос с PC, выполняющим роль мастера опроса. Действие PCF может требовать дополнительной координации, не указанной в этом стандарте, разрешать эффективное действие в случаях, где многократные скоординированные точкой BSS работает на том же самом канале с перекрытием физического пространства.

PCF использует виртуальный механизм чувствительности несущей, которому помогает механизм приоритета доступа. PCF должен распределить информацию в пределах фреймов управления Маяка, чтобы получить контроль над средой, устанавливая вектор распределения сети (NAV) в STA. Кроме того, все передачи фрейма под PCF могут использовать межфреймовое пространство (IFS), которое является меньшим чем IFS для фреймов, переданных через DCF. Использование меньшего IFS подразумевает, что скоординированное пунктом движение, будет иметь приоритетный доступ к среде через STA в перекрывающихся BSS, использующих методом доступа DCF.

Приоритет доступа, обеспеченный PCF может использоваться, чтобы создать метод доступа *без состязаний* (CF). PC управляет передачами фрейма STA, чтобы устранить состязания в течение ограниченного периода времени.

9.1.3 Сосуществование DCF и PCF

DCF и PCF должны сосуществовать в манере, которая разрешает обоим работать одновременно в пределах того же самого BSS. Когда PC работает в BSS, двух альтернативных методах доступа, с периодом без состязаний (CFP) сопровождаемый к периоду с состязаниями (CP). Подробнее это описано в 9.3.

9.1.4 Краткий обзор Фрагментации/Дефрагментации

Процесс разделения блока данных службы MAC (MSDU) или блока данных протокола управления MAC (MMPDU) в меньшие фреймы уровня MAC - блоки данных протокола MAC (MPDU), называются фрагментацией. Фрагментация создает MPDU меньшей длины, чем оригинал MSDU или MMPDU, чтобы увеличить надежность, увеличивая вероятность успешной передачи MSDU или MMPDU в случаях, где характеристики канала ограничивают надежный прием для более длинных фреймов. Фрагментация выполняется в непосредственном каждом передатчике. Процесс сборки MPDU в один MSDU или MMPDU определен как дефрагментация. Дефрагментация выполняется непосредственный в каждом приемнике.

Должны быть фрагментированы только MPDU с уникальным адресом приемника. Фреймы broadcast/multicast не должны быть фрагментированы, даже если их длина превышает aFragmentationThreshod.

Когда направленный MSDU получен от LLC, или направленный MMPDU получен от объекта управления подуровня MAC (MLME) с длиной большей, чем aFragmentationThreshold, MSDU или MMPDU должны быть фрагментированы. MSDU или MMPDU разделяются на MPDU. Длина каждого фрагмента - фрейма не больше чем aFragmentationThreshold. Возможно, что любой фрагмент может быть фреймом меньшим чем aFragmentationThreshold. Иллюстрация фрагментации показана на Рис. 17.

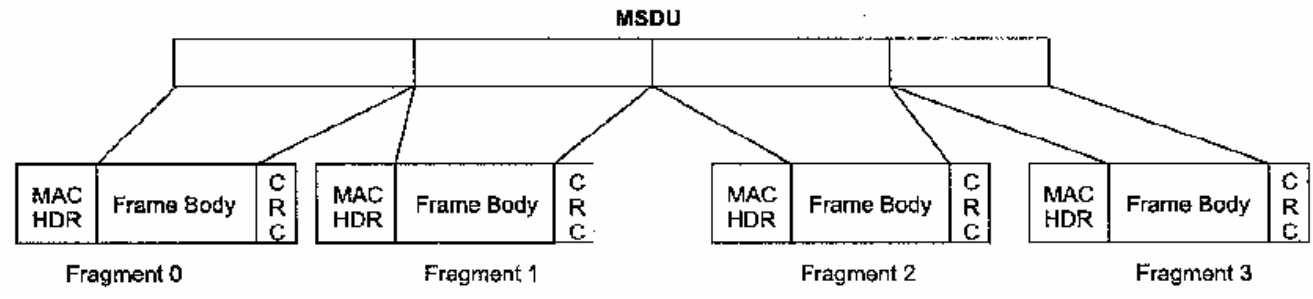


Рис. 17. Фрагментация

MPDU, полученные в результате фрагментации MSDU или MMPDU отправляются как независимые посылки, каждая из которых отдельно подтверждается. Это позволяет производить повтор посылок пофрагментно, что быстрее, чем повторять MSDU или MMPDU. Пока нет прерывания из-за ограничений среды для данного PHY, фрагменты одного MSDU или MMPDU отправляются как пакет в течение CP, используя одно обращение DCF процедуры доступа к среде. Фрагменты одного MSDU или MMPDU отправляются в течение CFP как индивидуальные фреймы по правилам процедуры доступа к среде PC.

9.1.5 Служба данных MAC

Служба данных MAC должна транслировать запросы обслуживания MAC от LLC во входные сигналы, используемые машинами состояния MAC. Служба данных MAC должна также транслировать выходные сигналы от машин состояния MAC в индикации службы к LLC. Трансляция показана в машине состояния службы данных MAC, определенной в Приложении C.

9.2 ***DCF***

Базовым протоколом MAC является DCF, который позволяет автоматически делить среду распространения между совместимыми PHY с помощью использования CSMA/CA и случайного времени backoff вслед за условием занятости среды. Кроме того, весь направленный трафик использует немедленное положительное подтверждение (фрейм ACK), где повторная передача планируется отправителем, если ACK не принят.

Протокол CSMA/CA разработан для уменьшения вероятности коллизий между несколькими STA, имеющими доступ к среде, в точке, где возникновение коллизий наиболее вероятно. Наибольшая вероятность возникновения коллизий имеет место сразу после того, как среда становится свободной (idle) после занятой (busy), что указывается функцией CS. Это происходит потому, что несколько STA могут зависнуть, ожидая доступа к среде. Подобная ситуация приводит к необходимости использования процедуры случайного backoff для разрешения конфликтов соединения со средой.

Контроль несущей (carrier sense) должен выполняться как с помощью физических, так и виртуальных механизмов.

Виртуальный механизм контроля несущей заключается в распределении резервирующей информации, которая анонсирует предстоящее использование среды. Распределение такой информации осуществляется путем обмена фреймами RTS и CTS перед началом передачи фреймов данных. Фреймы RTS и CTS содержат поле Duration/ID, которое определяет период времени, на который среда считается зарезервированной для передачи фреймов данных и ответных фреймов ACK. Все STA в течение приемного диапазона времени, включая вызывающие STA (которые передают RTS) и принимающие STA (которые передают CTS), должны исследовать резервацию среды. Таким образом, STA не может осуществить прием от вызывающей STA до тех пор, пока она не будет точно знать о предстоящем использовании среды для передачи фреймов данных.

Другим способом распределения резервирующей информации является наличие поля Duration/ID в направленных фреймах. Данное поле указывает время, на которое резервируется среда, либо до конца идущего вслед ACK, либо, в случае фрагментированной последовательности, до конца ACK, идущего вслед за следующим фрагментом.

Обмен RTS/CTS также выполняет быстрое устранение коллизий и проверку пути передачи. Если STA, отправившая RTS, не обнаруживает возврат CTS, она может повторить процесс (после просмотра других правил использования среды) более быстро, чем если бы передавались длинные фреймы данных и не был бы обнаружен возврат фрейма ACK.

Следующее преимущество механизма RTS/CTS в использовании перекрытия одного и того же канала. Механизм резервирования среды работает в рамках границ BSA. Механизм RTS/CTS может также улучшать работу в типичной ситуации, где все STA могут принимать от AP, но не могут от всех других STA в BSA.

Механизм RTS/CTS не может быть использован для MPDU с непосредственной broadcast и multicast адресацией, т.к. при этом получается много адресатов для RTS и возможно появление конкурентных отправителей ответа CTS. Нет необходимости использовать механизм RTS/CTS для передачи каждого фрейма данных. Т.к. дополнительные фреймы RTS и CTS добавляют непроизводительные накладные расходы, этот механизм не всегда обоснован, особенно для коротких фреймов данных.

Использование механизма RTS/CTS находится под контролем атрибута aRTSThreshold. Этот атрибут может быть установлен в базисе STA. Этот механизм допускает конфигурирование STA для использования RTS/CTS всегда, никогда или только для фреймов длиннее указанной длины.

STA, сконфигурированная как не инициирующая RTS/CTS механизм должна обновлять свой механизм чувствительности несущей с помощью информации продолжительности, содержащейся в принятых RTS или CTS фреймах и должна всегда отвечать фреймом CTS на адресованный ей RTS.

Протокол доступа к среде допускает, что STA поддерживает различные наборы скоростей. Все STA должны принимать на всех скоростях в aBasicRateSet и передавать на одной или более скорости из aBasicRateSet. Для поддержки правильного функционирования механизма RTS/CTS и виртуального механизма чувствительности к несущей, все STA должны иметь возможность работать на скоростях aBasicRateSet. (См. 9.6 для детального описания работы на нескольких скоростях.)

Фреймы данных, передаваемые через DCF, должны использовать тип фрейма «Данные», подтип «Данные» или «Пустая функция». STA, принимающие фреймы типа «Данные» должны рассматривать только тело фрейма как основу возможной индикации на LLC.

9.2.1 Механизм чувствительности к несущей

Физические и виртуальные функции чувствительные к несущей используются для определения состояния среды. Когда любая функция показывает занятость среды, среда должна распознаваться как занятая, иначе – как свободная.

Физический механизм чувствительности к несущей обеспечивается уровнем PHY. См. 12 для понимания, как эта информация передается на MAC. Детали физического механизма чувствительности к несущей определены в индивидуальной спецификации PHY.

Виртуальный механизм чувствительности к несущей обеспечивается уровнем MAC. Этот механизм ссылается на вектор распределения сети (NAV). NAV содержит прогноз будущего трафика в среде, базирующийся на информации длительности, которая анонсируется в фреймах RTS/CTS перед обменом данными. Информация длительности также доступна в заголовках MAC всех фреймов, отличных от PS-Poll Control, передаваемых в течение CP. Механизм установки NAV с использованием RTS/CTS в DCF описан в 9.2.5.4, а использование NAV в PCF рассматривается в 9.3.2.2.

Механизм чувствительности к несущей сочетает состояние NAV и статус передатчика STA с физической чувствительностью к несущей для определения занятого/свободного состояния среды. NAV может восприниматься как счетчик, который уменьшает свое состояние до нуля с постоянной скоростью. Когда счетчик доходит до нуля, виртуальный механизм чувствительности несущей говорит о свободности среды, иначе о занятости. Когда STA передает, среда должна определяться как занятая.

9.2.2 Подтверждения уровня MAC

Прием некоторых фреймов, как описано в 9.7, 9.2.8 и 9.3.3.4, требует подтверждения от принимающей STA, обычно фреймом ACK, если FCS принятого фрейма правильная. Эта технология понимается как позитивный ответ.

Отсутствие приема ожидаемого фрейма ACK индицирует STA-источнику, что обнаружена ошибка. Примечание: несмотря на то, что STA-приемник приняла фрейм корректно, ошибка может быть в приеме фрейма ACK. Для инициатора обмена это условие неотличимо от ошибки в исходном фрейме.

9.2.3 Межфреймовое пространство (IFS)

Интервал времени между фреймами называется межфреймовым пространством (IFS). STA должна определять состояние простоя среды через функцию чувствительности к несущей для определенных интервалов. Определены 4 различных IFS для обеспечения уровней приоритетного доступа к среде; они указаны в списке в порядке возрастания. На Рис. 18 показаны некоторые из этих соотношений.

a) SIFS короткое межфреймовое пространство
b) PIFS межфреймовое пространство PCF
c) DIFS межфреймовое пространство DCF
d) EIFS расширенное межфреймовое пространство

Различные IFS должны быть независимы от скорости данных STA. Времена IFS должны быть определены как временные промежутки среды и фиксированы для каждого PHY (PHY с одинаковой совместимостью многоскоростного режима). Значения IFS выделяются из атрибутов, определенных в PHY MIB.

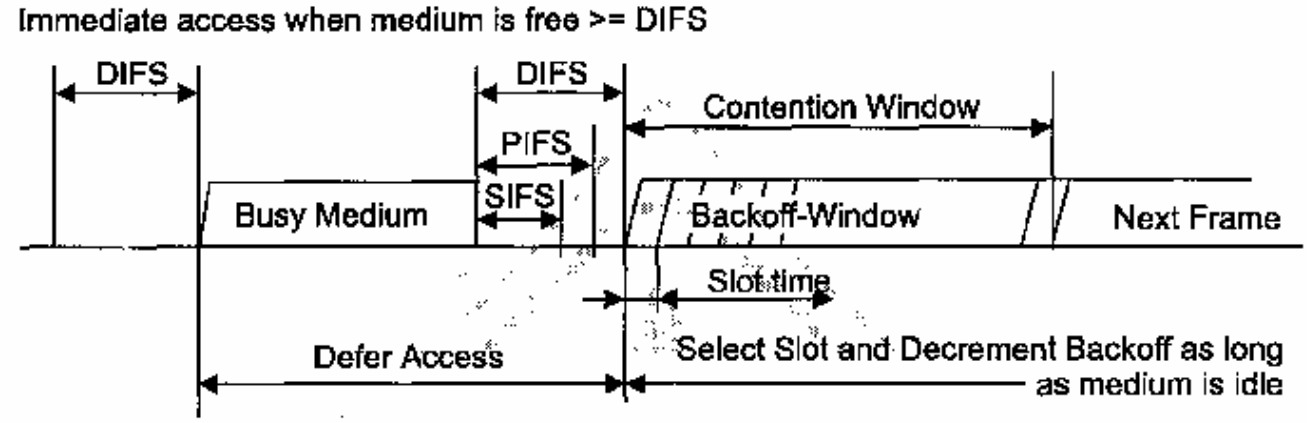


Рис. 18. Некоторые соотношения IFS.

9.2.3.1 Короткий IFS (SIFS)

SIFS должен использоваться для фрейма ACK, CTS, второго или последующего MPDU пакета фрагментов, STA отвечающей на любое голосование посредством PCF и может быть использован PC для любых типов фреймов в течение CFP (см. 9.3). SIFS – время между концом последнего символа предыдущего фрейма и началом первого символа преамбулы последующего фрейма, как показано в радиоинтерфейсе. Допустимые случаи, где может или должен использоваться SIFS перечислены в последовательности обмена фреймами (см. 9.7).

Временные соотношения SIFS должны быть соблюдены когда передача последующего фрейма начинается на границе TxSIFS Slot, как показано в 9.2.10. Реализация IEEE 802.11 не допускает пространства между фреймами, которые разделены SIFS интервалом, отличающимся от номинального значения SIFS на величину, большую чем +-10% от aSlotTime используемого PHY при измерении в среде.

SIFS – наиболее короткий из межфреймовых промежутков. SIFS должен использоваться, когда STA захватывают среду и нуждаются в ее удержании на время выполнения обмена последовательностью фреймов. использование наименьшего зазора между передачами в последовательности обмена фреймами, предупреждает другие STA, которым требуется ожидание большего зазора при освобождения среды и попытки ее использовать, что дает приоритет выполнения в процессе выполнения обмена фреймами.

9.2.3.2 PCF IFS (PIFS)

PIFS должен использоваться только STA, работающими под PCF для получения приоритетного доступа к среде при старте CFP. STA, использующая PCF, должна допускать передачу трафика без коллизий после того, как ее механизм чувствительности к несущей (см. 9.2.1) определит, что среда свободна на границе TxPIFS слота, как определено в 9.2.10. Подпункт 9.3 обсуждает использование PIFS STA, работающими под PCF.

9.2.3.3 DCF IFS (DIFS)

DIFS должен использоваться STA, работающими под DCF для передачи фреймов данных MPDU и фреймов управления MMPDU. STA, использующая DCF должна допускать передачу, если ее механизм чувствительности к среде (см. 9.2.1) распознает, что среда свободна на границе слота TxDIFS, как определено в 9.2.10, после правильно принятого фрейма и истечения интервала ожидания. STA, использующая DCF не должна передавать в EIFS после определения что среда свободна после приема фрейма для которого проимитив PHY-RXEND.indication содержит ошибку или фрейм, в котором значение MAC FCS не правильное. STA может передавать после приема свободного от ошибок фрейма, ресинхронизирующего STA. Это допускает, что STA передает с использованием DIFS, следующий за этим фреймом.

9.2.3.4 Расширенный IFS (EIFS)

EIFS должен использоваться DCF всякий раз, когда PHY указывает MAC что была начата передача фрейма, не являющаяся результатом корректного приема полного фрейма MAC с корректным значением FCS. Длительность EIFS определена в 9.2.10. EIFS должен начинаться после индикации от PHY что среда свободна после определения ошибочного фрейма без рассмотрения виртуальным механизмом чувствительности к несущей. EIFS определен для обеспечения достаточного времени для ответа другой STA этой STA, что был неправильно принят фрейм, до того как эта STA начала передачу. Прием свободного от ошибок фрейма в течение EIFS ресинхронизирует STA к текущему занятому/свободному состоянию среды, таким образом, EIFS прерывается и продолжается нормальный доступ к среде (с использованием DIFS и, если нужно, интервал ожидания) вслед за приемом этого фрейма.

9.2.4 Случайное время ожидания (backoff time)

STA, желая инициировать передачу MPDU данных и/или MMPDU управления должна запросить механизм чувствительности к несущей (см. 9.2.1) для определения занятого/свободного состояния среды. Если среда занята, STA должна подождать, пока среда освободится на период времени, эквивалентный DIFS, когда последний фрейм, обнаруженный в среде, принят корректно, или после того, как определено, что среда свободна в течение периода времени, равному EIFS, когда последний фрейм, обнаруженный в среде принят не корректно. После DIFS или EIFS времени простоя среды, STA должна выполнить случайный интервал ожидания для дополнительной задержки времени перед передачей, пока таймер задержки содержит ненулевое значение, выбор случайного не нужен и не производится. Этот процесс минимизирует коллизии в процессе состязаний нескольких STA, т.к. разносит во времени наступление одинаковых событий.

$$BackoffTime = Ramdom() \times aSlotTime,$$

где

Random() – псевдослучайное целое число равномерно распределенное в интервале [0, CW], где CW – целое в диапазоне значений атрибутов MIB aCWmin и aCWmax $aCW\min \le CW \le aCW\max$. Это важно, что проектировщик распознает необходимость статистической независимости между потоками случайных значений между STA.

aSlotTime –значение соответствующего MIB атрибута.

Параметр окно конкурренции (CW) должен принимать начальное значение aCWmin. Каждая STA должна поддерживать короткий счетчик повторов STA (SSRC) так же как и длинный счетчик повторов STA (SLRC), которые должны инициализироваться нулем. SSRC должен инкрементироваться всякий раз, когда инкрементируется короткий счетчик повторов, ассоциированный с любым MSDU. SLRC должен инкрементироваться всякий раз, когда инкрементируется длинный счетчик повторов, ассоциированный с любым MSDU. CW получает следующее значение из серии при каждой неудачной по-

пытке передачи MPDU по причине инкрементирования счетчика повторов STA, пока CW не достигнет aCWmax. Повтор определен как последовательность наборов фреймов, разделенных SIFS интервалами, в попытке доставки MPDU, как описано в 9.7. При достижении aCWmax, CW сохраняет это значение до сброса. Это увеличиваает стабильность протокола доступа при условии высокой загрузки. См. Рис. 19.

CW должен сбросится в значение aCWmin после каждой успешной попытки передачи MSDU или MMPDU, когда SLRC достиг aLongRetryLimit или SSRC достиг aShortRetryLimit. SSRC должен сброситься в 0 всякий раз, когда принят фрейм CTS в ответ на фрейм RTS, когда принят фрейм ACK в ответ на передачу MPDU или MMPDU, или когда передан фрейм с групповым адресом в поле «Адрес 1». SLRC должен сброситься в 0 всякий раз, когда принят фрейм ACK в ответ на передачу MPDU или MMPDU, длина которых превышала aRTSThreshold, или когда передан фрейм с групповым адресом в поле «Адрес 1».

Набор значений CW должен быть последовательно увеличиваться по степени 2 минус 1 (2^n-1), начиная со значения aCWmin до aCWmax, зависимых от PHY.

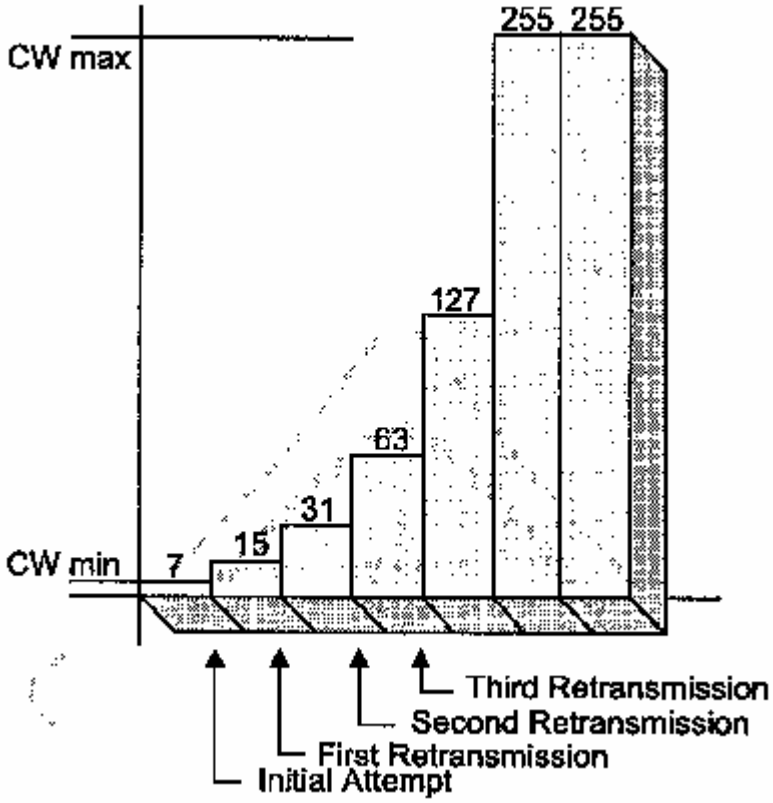


Рис. 19. Пример экспоненциального увеличения CW.

9.2.5 Процедура доступа DCF

Метод CSMA/CA является основой DCF. Правила функционирования DCF и PCF отличаются незначительно.

9.2.5.1 Базовый доступ

Базовый доступ относится к ядру механизма STA и используется для определения, может ли она передавать.

В общем случае STA может передавать задержанные MPDU при работе или по методу доступа DCF, при отсутствии PC, или по методу доступа PCF CP, когда STA определяет, что среда свободна в течение периода времени, большего или равного DIFS или EIFS, если этому предшествовало событие

занятости среды по причине приема STA фрейма с неправильным значением FCS. В этом случае, если механизмом чувствительности к несущей определяется занятость среды когда STA желает инициировать начальный фрейм одного из обменов фреймами, описанных в 9.7, исключая CF период, обмену должен предшествовать алгоритм случайного интервала ожидания, описанный в 9.2.5.2. Существуют условия, описанные в других местах пункта 9, где алгоритм случайного интервала ожидания должен предшествовать каждой первой попытке начать последовательность обмена фреймами.

В STA с FH PHY контролируется потеря канала на границе времени dwell и STA должна бороться за канал после границы dwell. Это требуется для STA, имеющих законченную передачу полного MPDU и ассоциированный ответ (если требуется) до нахождения на границе времени dwell. Если, когда передается или повторяется MPDU, недостаточно времени в dwell для передачи MPDU + ответ (если требуется), STA должна отложить передачу путем выбора случайного интервала ожидания с использованием активного CW (без продвижения к следующему значению в серии). Короткие и длинные счетчики повторов для MSDU не затрагиваются.

Базовый механизм доступа показан на Рис. 20.

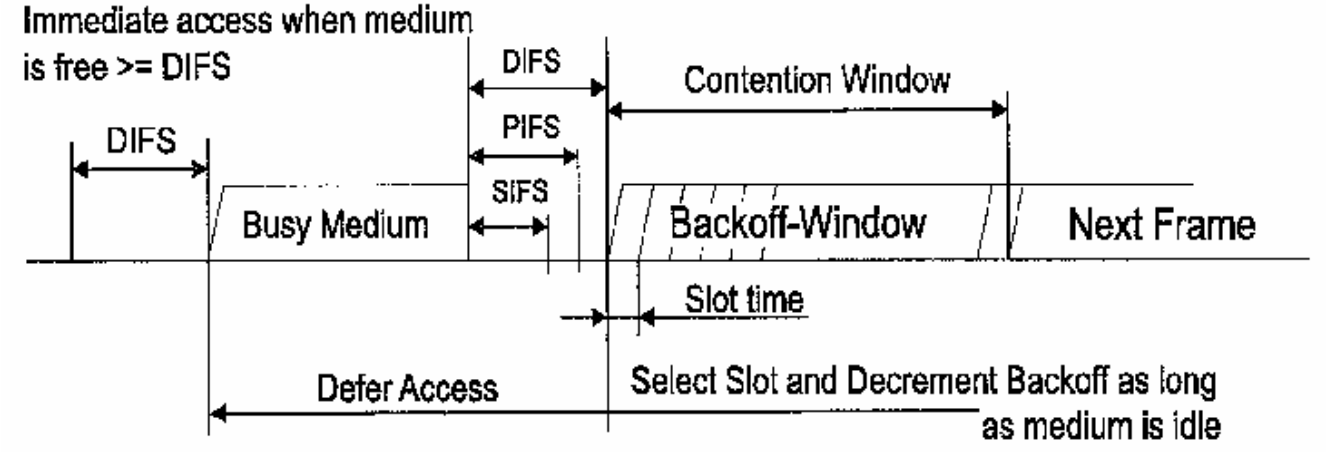


Рис. 20. Базовый метод доступа.

9.2.5.2 Процедура интервала ожидания (backoff)

Процедура backoff должна быть вызвана STA для передачи фрейма, когда обнаружена занятость среды, указываемая физическим или виртуальным механизмом чувствительности к несущей см. Рис. 21. Процедура backoff должна быть вызвана также, когда передающая STA делает вывод о неудачной передаче как определено в 9.2.5.7 или 9.2.8.

Для начала процедуры backoff, STA должна установить свой таймер backoff в случайное значение времени, используя равенство в 9.2.4. Все слоты backoff, находящиеся в следующем DIFS периоде в течение которого определено, что среда свободна на всем протяжении DIFS периода или следующем EIFS периоде в течение которого определено, что среда свободна на всем протяжении EIFS периода, следующего за определением, что фрейм принят некорректно.

STA, выполняющая процедуру backoff должна использовать механизм чувствительности к несущей (9.2.1) для определения наличия активности в процессе каждого слота backoff. Если не было замечено активности в течение слота backoff, то процедура backoff должна уменьшить свое время backoff на aSlotTime.

Если среда определена, как занятая в любое время слота backoff, процедура приостанавливается, т.е. таймер backoff не декрементируется для этого слота. Среда должна быть свободной в течение периода DIFS или EIFS, как определено (см. 9.2.3) для возможности восстановления работы процедуры backoff. Передача должна начинаться всякий раз, когда таймер backoff достигает нуля.

Процедура backoff должна быть выполнена непосредственно после окончания каждой передачи с битом MoreFragments установленным в 0 для MPDU типа «Данные», «Управление» или «Контроль» с подтипом PS-Poll даже если в очереди нет дополнительных передач. В случае успешно подтвержденных

передач, эта процедура backoff должна начинаться сразу по окончании принятого фрейма ACK. В случае неуспешных передач, требующих подтверждения, процедура backoff, должна начинаться после истечения интервала ожидания фрейма ACK. Если передача успешна, значение CW возвращается к CWmin до выбора случайного интервала ожидания и счетчик короткого и/или длинного повтора изменяются как указано в 9.2.4. Это гарантирует, что передачи фреймов от STA всегда разделены по крайней мере одним интервалом backoff.

Эффект этой процедуры заключается в следующем: когда несколько STA и откладывают и начинают со случайным временем backoff, STA, имеющая наименьший backoff, выигрывает состязание.

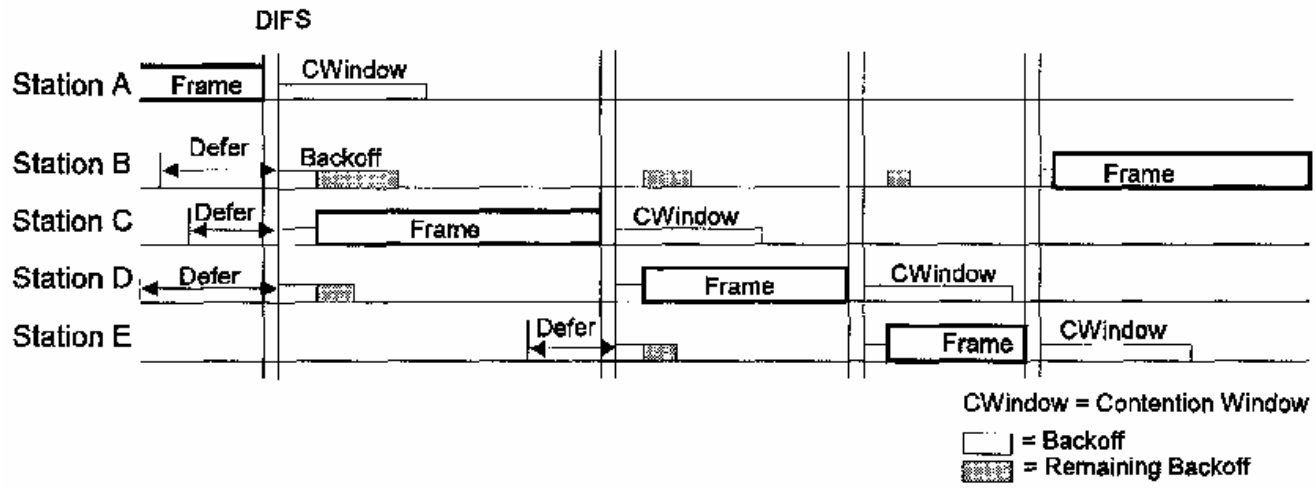


Рис. 21. Процедура backoff.

В IBSS время backoff для ожидания передач не-маяка и не-ATIM не должно уменьшаться с времени передачи целевого маяка (TBTT) до истечения окна ATIM и время backoff для ожидания фрейма управления ATIM должно уменьшаться только во время окна ATIM. (См. п. 11.) В IBSS разделительные интервалы backoff должны вырабатываться перед передачей маяка, как описано в 11.1.2.2.

9.2.5.3 Процедуры восстановления и лимиты повторных передач

Устранение ошибок всегда является обязанностью STA, начавшей последовательность обмена фреймами, как определено в 9.7. Может быть много случаев, когда потребуется устранение ошибок. Например, не обнаружен фрейм CTS после отправки RTS. Это может произойти из-за коллизии с другим передатчиком, интерференции с другим каналом в течение фреймов RTS или CTS, или потому, что на STA, принимающая фрейм RTS, было активно условие виртуального механизма чувствительности к несущей (показывающее занятость среды в данном периоде времени).

Устранение ошибок должно ограничиваться повтором передачи для неудавшейся последовательности обмена фреймами, начатой STA. Повторы должны продолжаться для каждой ошибочной последовательности обмена, пока передача не будет выполнена или если израсходован соответствующий лимит, что бы ни было первым. STA должна поддерживать короткий и длинный счетчик повтора для каждого MSDU или MMPDU, ожидающих передачу. Эти счетчики должны инкрементироваться и сбрасываться независимо друг от друга.

После передачи фрейма RTS, STA должна выполнить процедуру CTS, как описано в 9.2.5.7. Если передача RTS была неудачной, короткий счетчик повтора MSDU или MMPDU и короткого счетчик повтора STA инкрементируются. Это должно продолжаться, пока количество попыток передачи MSDU или MMPDU не превысит aShortRetryLimit.

После передачи фрейма, требующего подтверждения, STA должна выполнить процедуру ACK, как определено в 9.2.8. Короткий счетчик повтора MSDU или MMPDU и короткий счетчик повтора STA должны инкрементироваться при каждой неудачной передаче фрейма MAC, длинной меньшей либо равной aRTSThreshold для этого MSDU или MMPDU. Короткий счетчик повтора и короткий счетчик повтора STA должны быть сброшены при успешной передаче фрейма MAC, длинной меньшей либо

равной aRTSThreshold для этого MSDU или MMPDU. Длинный счетчик повтора MSDU или MMPDU и длинный счетчик повтора STA должны инкрементироваться при каждой неудачной передаче фрейма MAC, длинной большей aRTSThreshold для этого MSDU или MMPDU. Длинный счетчик повтора и длинный счетчик повтора STA должны быть сброшены при успешной передаче фрейма MAC, длинной большей aRTSThreshold для этого MSDU или MMPDU. Попытки повторной передачи MSDU или MMPDU, с безуспешной процедурой ACK один или несколька раз должны установить поле Retry в «1» в фреймах данных и управления.

Повторы неудачных попыток передачи должны продолжаться пока короткий счетчик повторов для MSDU или MMPDU не достигнет aShortRetryLimit или пока длинный счетчик повторов для MSDU или MMPDU не достигнет aLongRetryLimit. Когда любой из этих пределов достнут, попытки повтора прекращаются и MSDU или MMPDU отбрасывается.

STA в режиме энергосбережения в ESS, начинает последовательность обмена фреймами с посылки фрейма PS-Poll для запроса данных с AP. Если от AP, в ответ на фрейм PS-Poll, не принят ни фрейм ACK, ни фрейм данных, STA должна повторить последовательность передачей другого фрейма PS-Poll, как ей будет удобно. Если AP отправляет фрейм данных в ответ на фрейм PS-Poll, но не принимает фрейм ACK, подтверждающий этот фрейм данных, следующий фрейм PS-Poll от той же STA может вызвать повтор передачи последнего MSDU. Дубликаты MSDU должны фильтроваться принимающей STA с использованием нормального механизма фильтрации дубликатов фреймов. Если AP ответила на PS-Poll передачей фрейма ACK, ответственность за устранение ошибок при доставке фрейма смещается к AP, т.к. данные передаются в последовательности обмена фреймами, начатой AP. AP должна попытаться доставить один MSDU до STA, передавшей PS-Poll, используя любую последовательность обмена фреймами, доступную для направленных MSDU. Если STA в режиме энергосбережения, передавшая PS-Poll вернулась в состояние «Doze» после передачи фрейма ACK в ответ на успешный прием этого MSDU, но AP не приняла фрейм ACK, AP должна повторять передачу этого MSDU пока не истечет действующий лимит повторов. Детали фильтрации дополнительных PS-Poll фреймов см. в п. 11.

9.2.5.4 Установка и сброс NAV

STA, принимающая правильный фрейм, должна обновить свой NAV информацией, полученной в поле Duration/ID, но только тогда, когда новое значение NAV больше чем текущее значение NAV и только когда фрейм не адресован принимающей STA. Различные дополнительные условия при которых выполняется установка и сброс NAV, обсуждаются в 9.3.2.2. Когда сбрасывается NAV, должен быть отправлен примитив PHY-CCARESET.request.

Рис. 22 показывает NAV для STA, которые могут слушать фрейм RTS, пока другие STA могут только принимать фрейм CTS, результат чего показан на нижней шкале (исключая STA, которой адресован RTS).

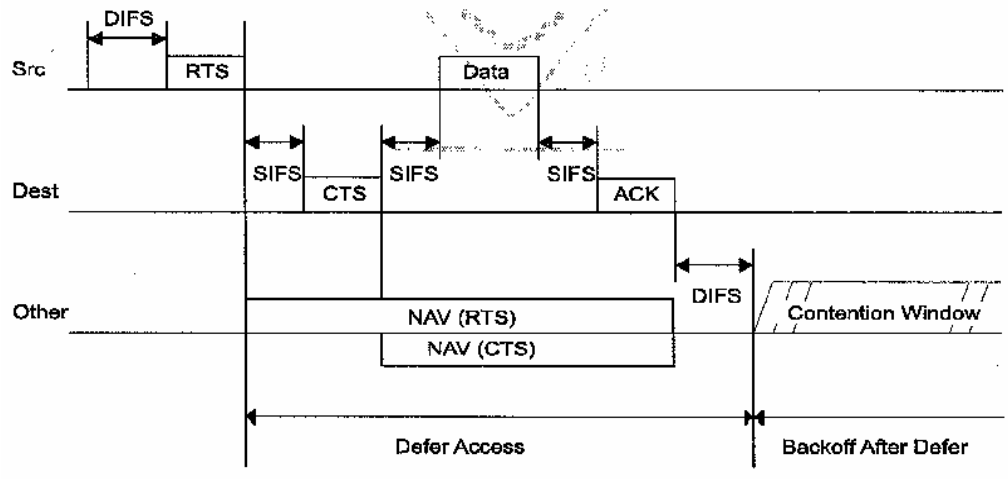


Рис. 22. Установка RTS/CTS/данных/ACK и NAV.

STA, использующей информацию фрейма RTS как более новую базу для обновления установок NAV, разрешается сбросить свой NAV, если нет примитива PHY-RXSTART.indication от PHY в течение периода длиной $(2 \times aSIFSTime) + (CTS\_Time) + (2 \times aSlotTime)$ начатого с момента получения PHY-RXEND.indication, соответствующего нахождению фрейма RTS. CTS_Time должно быть вычислено с использованием длины фрейма CTS и скорости данных принятого фрейма RTS, использованного для обновления NAV.

9.2.5.5 Контроль канала

SIFS используется для обеспечения эффективного механизма доставки MSDU. Если STA вступила в борьбу за канал, она должна продолжать отправлять фрагменты, пока все отправленные фрагменты одного MSDU или MMPDU не будут подтверждены или STA ограничится передачей нескольких дополнительных фрагментов из-за границы времени dwell. STA должна восстановить передачу когда предоставляется следующая благоприятная возможность, если отправка фрагментов была прервана по одной из причин. Алгоритм по которому STA принимает решение, что нужна следующая попытка после безуспешной передачи MSDU, находится за пределами этого стандарта, но некоторые пункты алгоритма должны быть выполнены в соответствии с требованиями, перечисленными в 9.8.

Рис. 23 иллюстрирует передачу многофрагментного MSDU с использованием SIFS.

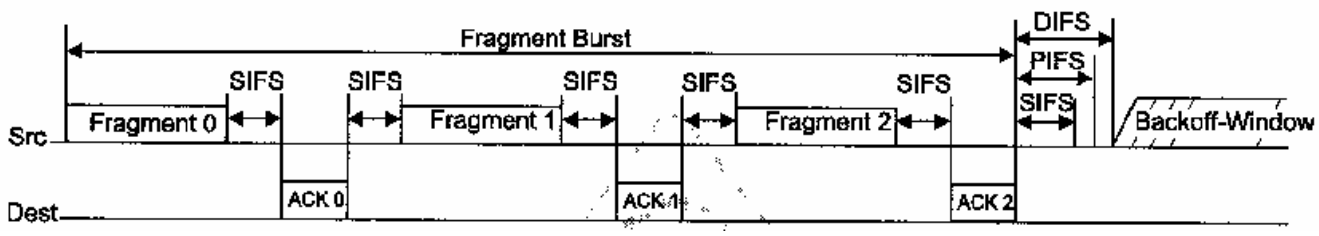


Рис. 23. Передача многофрагментного MSDU с использованием SIFS

Когда STA-источник передала фрагмент, она должна освободить канал и немедленно начать его мониторинг на предмет подтверждения, как описано в 9.2.8.

Когда STA-получатель завершает передачу подтверждения, должен быть зарезервирован SIFS для STA-источника для продолжения (если необходимо) со следующего фрагмента. STA, передающая подтверждение не должна передавать на канале сразу за подтверждением.

Процесс передачи нескольких фрагментов после соперничества за канал, определен как пакет фрагментов.

Если STA-источник приняла подтверждение, но недостаточно времени для передачи следующего фрагмента и приема подтверждения из-за появления границы dwell, она должна бороться за канал в следующем времени dwell.

Если STA-источник не приняла фрейм подтверждения она должна попытаться повторить неудачно переданный MPDU или другой пригодный MPDU, как определено в 9.8, после выполнения процедуры backoff и процесса соперничества.

После того, как в результате соперничества STA получила канал для передачи фрагмента MSDU, она должна начать с первого неподтвержденного фрагмента. STA-получатель должна принять фрагменты по порядку (в том как передает источник и каждый индивидуально подтверждается). Однако возможно, что STA-получатель будет принимать дубликаты фрагментов. Принимающая STA должна определять и отбрасывать дубликаты.

STA должна передавать после SIFS в течение пакета фрагментов только в случае, если:

- STA приняла фрагмент, требующий подтверждения.
- STA-источник приняла подтверждение предыдущего фрагмента, имеет фрагменты одного MSDU для передачи и не имеет времени перед следующей границей dwell для отправки следующего фрагмента и принятия подтверждения.

Должны также применяться следующие правила:

- Когда STA передает фрейм, отличный от начального или промежуточного фрагмента, она не должна передавать на канале после подтверждения этого фрейма без выполнения процедуры backoff.
- Когда MSDU успешно доставлен, все попытки повторной передачи должны быть исключены и STA имеет следующий MSDU для передачи, должна быть выполнена процедура backoff.
- Повторно передаваться должны только неподтвержденные фрагменты.

9.2.5.6 Использование RTS/CTS с фрагментацией

Далее следует описание использования RTS/CTS для фрагментированных MSDU или MMPDU. Фреймы RTS/CTS определяют длину последующего фрейма и подтверждения. Поле Duration/ID в фреймах данных и фреймах подтверждения (ACK) определяют полную длину следующего фрагмента и подтверждения. Это показано на .

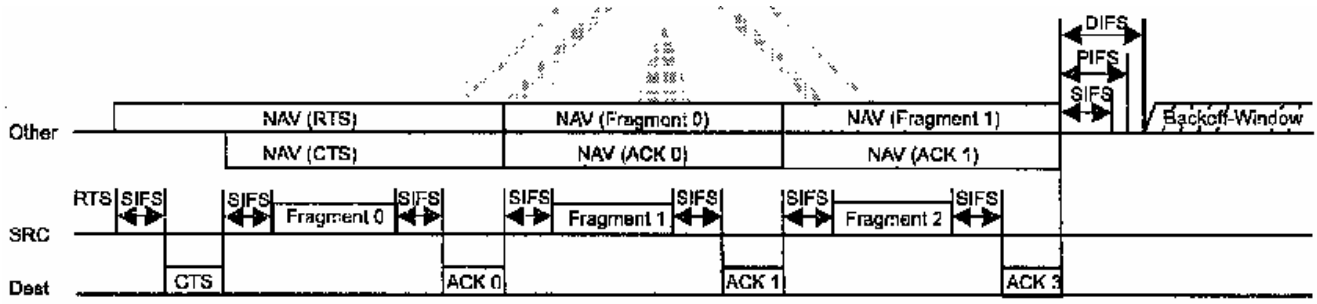


Рис. 24. RTS/CTS с фрагментированным MSDU.

Каждый фрейм содержит информацию, определяющую продолжительность следующей передачи. Информация продолжительности из фрейма RTS должна быть использована для обновления NAV для индикации занятости до конца ACK 0. Информация продолжительности из фрейма CTS также должна использоваться для обновления NAV для индикации занятости до конца ACK 0. Fragment 0 и ACK 0 должны содержать информацию продолжительности для обновления NAV для индикации занятости до конца ACK 1. Это должно быть выполнено путем использования поля Duration/ID в фреймах данных и ACK. Это должно продолжаться, пока не обнаружится последний фрагмент, имеющий продолжительность одного времени ACK плюс одно время SIFS и его подтверждение с полем Duration/ID, установленным в 0. Каждый фрагмент и ACK действуют как виртуальные RTS и CTS; следовательно, нет нужды вырабатывать фреймы RTS/CTS после начала последовательности обмена, несмотря на то, что последовательные фрагменты могут быть больше чем aRTSThreshold. STA, использующие PHY с ППРЧ, когда недостаточно времени до границы dwell для передачи следующего фрагмента, STA, начавша последовательность обмена, может установить поле Duration/ID в последнем фрейме данных или управления, переданном перед границей dwell, в значение продолжительности одного времени ACK плюс одно время SIFS.

В случае, если подтверждение отправлено, но не принято STA-источником, STA, слышавшие фрагмент или ACK, должны указать занятость канала для следующего обмена фреймами, по причине обновления NAV из этих фреймов. Это наихудшая ситуация и она показана на Рис. 25. Если подтверждение не было отправлено STA-получателем, STA, которые могут слышать только STA-получатель, не обновляют свой NAV и могут производить попытки доступа к каналу когда их NAV, обновленный из предыдущего принятого фрейма достигнет нуля. Все STA, которые слышали источник могут получить доступ к каналу когда их NAV, обновленный из переданного фрагмента истечет.

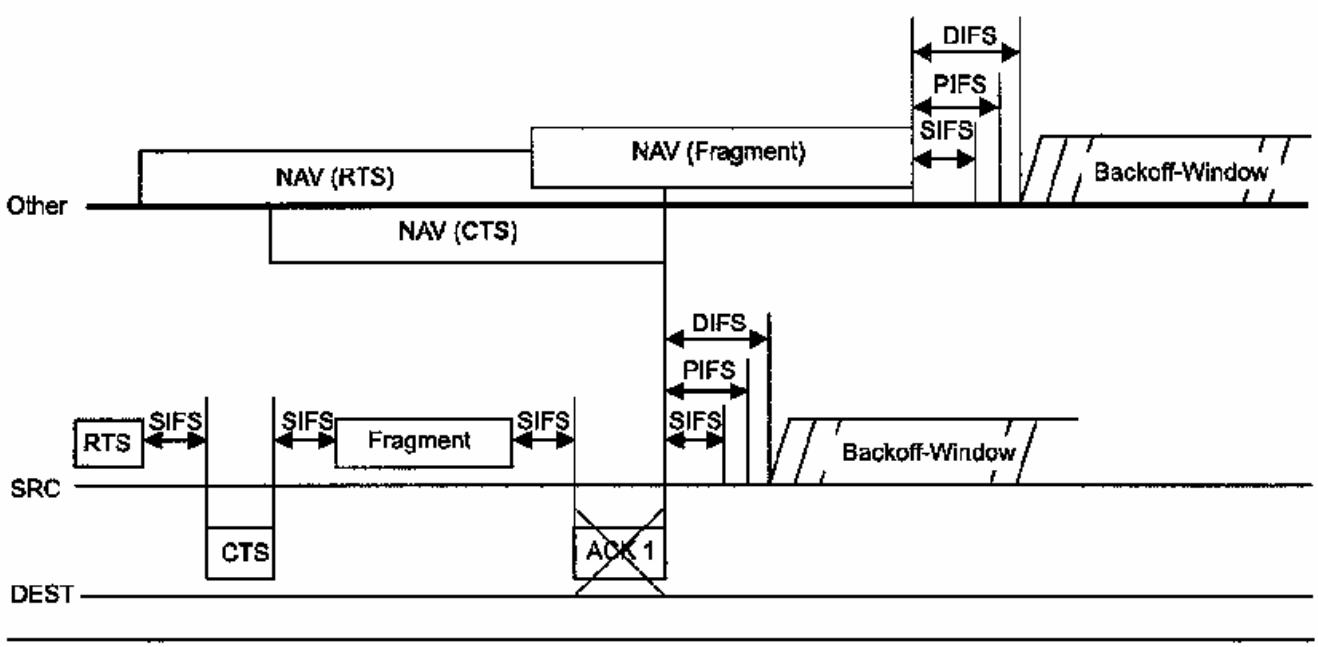


Рис. 25. RTS/CTS с приоритетом передатчика и отсутствующим подтверждением.

9.2.5.7 Процедура CTS

STA, которой адресован фрейм RTS, должна передать фрейм CTS после периода SIFS, если NAV на STA, принявшей фрейм RTS показывает, что среда свободна. Если NAV на STA, принявшей фрейм RTS показывает, что среда занята, STA не должна отвечать на на фрейм RTS. Поле RA фрейма CTS должно содержать значение, полученное из поля TA фрейма RTS, на который отвечает CTS. Поле Duration/ID фрейма CTS должно содержать значение из поля продолжительности фрейма RTS минус aSIFSTime и число микросекунд, требуемого для передачи фрейма CTS на скорости фрейма RTS, для которого CTS является ответом.

После пердачи фрейма RTS, STA должна выждать интервал CTSTimeout, начиная с PHY-TXEND.confirm. Если нет PHY-RXSTART.indication в течение интервала CTSTimeout, STA должна завершить неудачную передачу RTS и выполнить процедуру backoff по истечении интервала CTSTimeout. Если обнаружен PHY-RXSTART.indication в течение интервала CTSTimeout, STA должна ожидать соответствующий PHY-RXEND.indication для определения что передачу RTS выполнена. Распознавание правильного фрейма CTS, отправленного приемником фрейма RTS, указанного этим PHY-RXEND.indication, должно пониматься как удачное подтверждение и позволяет продолжать последовательность фреймов (см. 9.7). Распознавание любых других, включая другие правильные фреймы, должно пониматься как неудачная передача фрейма RTS. В этом случае STA должна выполнить процедуру backoff после PHY-RXEND.indication и может обрабатывать принятый фрейм.

9.2.6 Процедура передачи направленных MPDU

STA должна использовать обмен RTS/CTS для направленных фреймов только в том случае, если длина MPDU больше, чем пороговая длина, указываемая атрибутом aRTSTreshold.
Атрибут aRTSTreshold должен быть управляемым объектом внутри MAC MIB, а его значение может устанавливаться и читаться MAC LME. Значение 0 должно использоваться для индикации того, что MPDU будут доставляться с использованием RTS/CTS. Значения aRTSTreshold, большие, чем максимальная длина MSDU, должны указывать на то, что все MPDU будут доставляться без обмена RTS/CTS.
Когда обмен RTS/CTS используется, асинхронные фреймы данных должны передаваться по окончании фрейма CTS и периода SIFS. При передаче такого фрейма не должно уделяться никакого внимания состоянию среды (свободна или занята).

Когда обмен RTS/CTS не используется, асинхронные фреймы данных должны передаваться после успешной процедуры базового доступа. В присутствии или без обмена RTS/CTS та STA, которая является местом назначения асинхронного фрейма данных, должна следовать процедуре ACK.

9.2.7 Процедура передачи broadcast/multicast MPDU

В отсутствии PCF, когда broadcast/multicast MPDU передаются от STA с очищенным битом ToDS, должна использоваться только базовая процедура доступа. Независимо от длины фрейма обмен RTS/CTS использоваться не должен. Кроме того, приемниками фреймов не должны передаваться никакие ACK.
Любые broadcast/multicast MPDU, передаваемые от STA с установленным битом ToDS, должны кроме соблюдения базовой процедуры доступа CSMA/CA удовлетворять правилам обмена RTS/CTS, поскольку MPDU направлены к AP. Broadcast/multicast сообщение должно быть распределено внутри BSS. STA, создающая сообщение, должна принимать сообщение как broadcast/multicast сообщение. Таким образом, все STA должны отфильтровывать broadcast/multicast сообщения, которые содержат их адрес в качестве источника. Broadcast и multicast MSDU должны распространяться внутри ESS.
На уровне MAC не существует процедуры восстановления broadcast/multicast фреймов, за исключением фреймов, посылаемых с установленным битом ToDS. Поэтому надежность такого трафика уменьшается по сравнению с направленным трафиком из-за увеличения вероятности потери фреймов в результате интерференции, коллизий или меняющихся со временем свойств канала.

9.2.8 Процедура ACK

Фрейм ACK должен генерироваться так, как показано во фреймовых последовательностях, перечисленных в 9.7.
После успешного приема фрейма с типом, требующим подтверждения, и установленным битом ToDS, AP должна сгенерировать фрейм ACK. Фрейм ACK должен передаваться STA, являющейся местом назначения и не являющейся AP, всякий раз, когда она принимает однонаправленный фрейм с типом, требующим подтверждения, и не должен передаваться, если она принимает broadcast/multicast фрейм такого типа. После успешного приема фрейма, требующего подтверждения, передача фрейма ACK должна начинаться через период SIFS, независимо от состояния среды (свободна или занята).
STA-источник должна ожидать приема фрейма ACK время, равное ACKTimeout, до того, как она посчитает, что MPDU потеряны (см. Рис. 26).

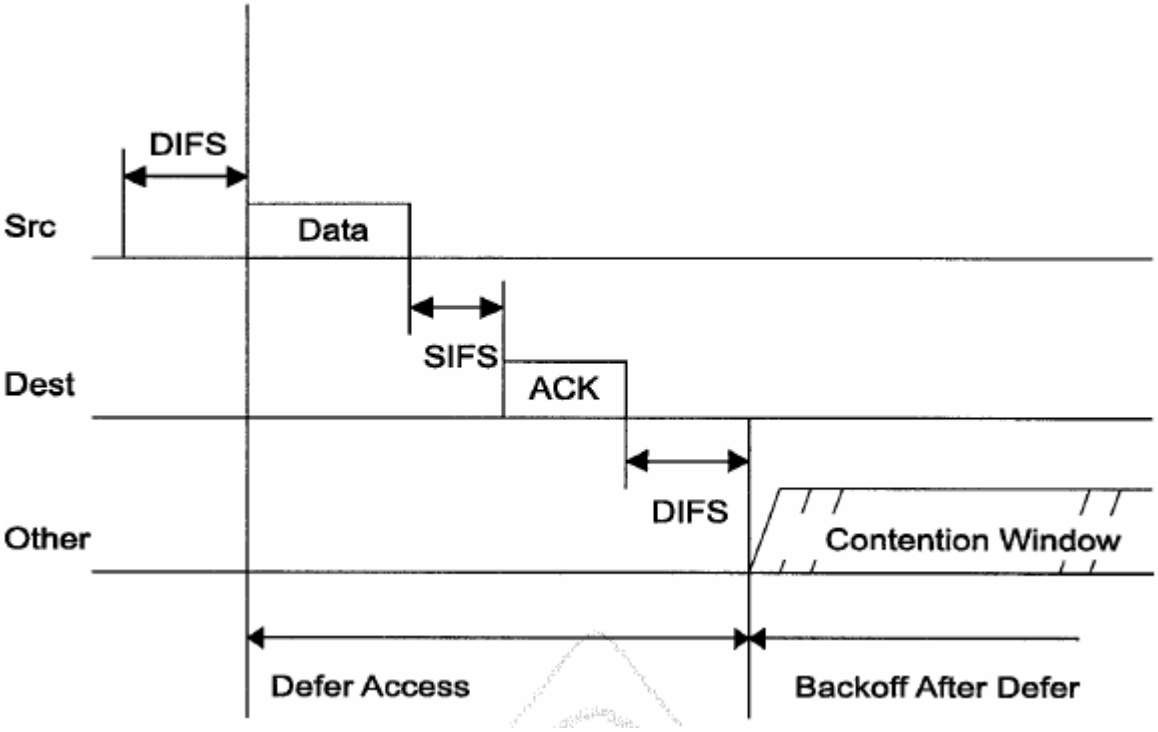


Рис. 26. Направленные данные/ACK MPDU.

После передачи MPDU, которые требуют ответа фреймом ACK (см. 9.7), STA должна ждать в течение интервала ACKTimeout, начинающегося от PHY-TXEND.confirm. Если в течение этого интервала не придет PHY-RXSTART.indication, STA делает вывод о том, что передача MPDU была неудачной, и должна вызвать свою процедуру backoff по истечении интервала ACKTimeout. Если в течение интервала ACKTimeout приходит PHY-RXSTART.indication, STA должна ожидать соответствующий PHY-RXEND.indication для определения, была ли передача MPDU успешной. Если действительный фрейм ACK, посланный приемником MPDU, требовавших подтверждения, и соответствующий данной PHY-RXEND.indication, распознан, то это должно считаться успешным подтверждением, разрешением продолжения фреймовой последовательности либо ее окончанием, в зависимости от текущего состояния работы. Распознавание чего-либо другого, включая любой другой действительный фрейм, должно интерпретироваться как ошибка передачи MPDU. В этом случае STA должна вызвать свою процедуру backoff по приему PHY-RXEND.indication и может обработать принятый фрейм. Единственным исключением является действительный фрейм данных, посланный приемником фрейма PS-Poll, он также должен считаться успешным подтверждением фрейма PS-Poll.

9.2.9 Обнаружение дубликатов и восстановление данных

Поскольку подтверждения и повторные передачи уровня MAC встроены в протокол, существует вероятность того, что фрейм может быть принят более одного раза. Такие дубликаты должны быть отфильтрованы внутри MAC места назначения.

Фильтрация дубликатов облегчается наличием поля Sequence Control (содержащим последовательный номер и номер фрагмента) внутри фреймов данных и управления. MPDU, которые являются частью одних и тех же MSDU, должны иметь одинаковый последовательный номер, а разные MSDU должны (с высокой степенью вероятности) иметь различный последовательный номер.

Последовательный номер генерируется передающей STA как инкрементирующаяся последовательность целых чисел.

Приемная STA должна хранить кэш-запись из последних принятых <Address 2, sequence-number, fragment-number>. Приемной STA необходимо хранить только самую последнюю пару Address 2 – sequence-number, а также самый последний fragment-number для этой пары. Приемная STA может опустить запись, полученную от фреймов broadcast/multicast или ATIM, из кэш.

STA, являющаяся местом назначения, должна отражать как дубликат любой фрейм, который имеет установленный бит Retry в поле Frame Control и совпадающую с хранимой запись <Address 2, sequence-number, fragment-number>.
Существует небольшая вероятность того, что фрейм будет отражен неправильно в результате такого совпадения; однако, это может случаться нечасто, и приведет просто к потере фрейма (аналогично ошибке FCS в других протоколах LAN).
STA, являющаяся местом назначения, должна выполнять процедуру ACK для всех успешно принятых фреймов, требующих подтверждения, даже если фрейм отражен как дубликат.

9.2.10 Временные соотношения DCF

Отношения между спецификациями IFS определяются как временные промежутки среды. Соответствующие атрибуты MIB обеспечиваются частным PHY (см. Рис. 27).
Все времена, которые указаны от конца передачи, относятся к концу последнего символа фрейма в среде. Начало передачи относится к первому символу следующего фрейма в среде.

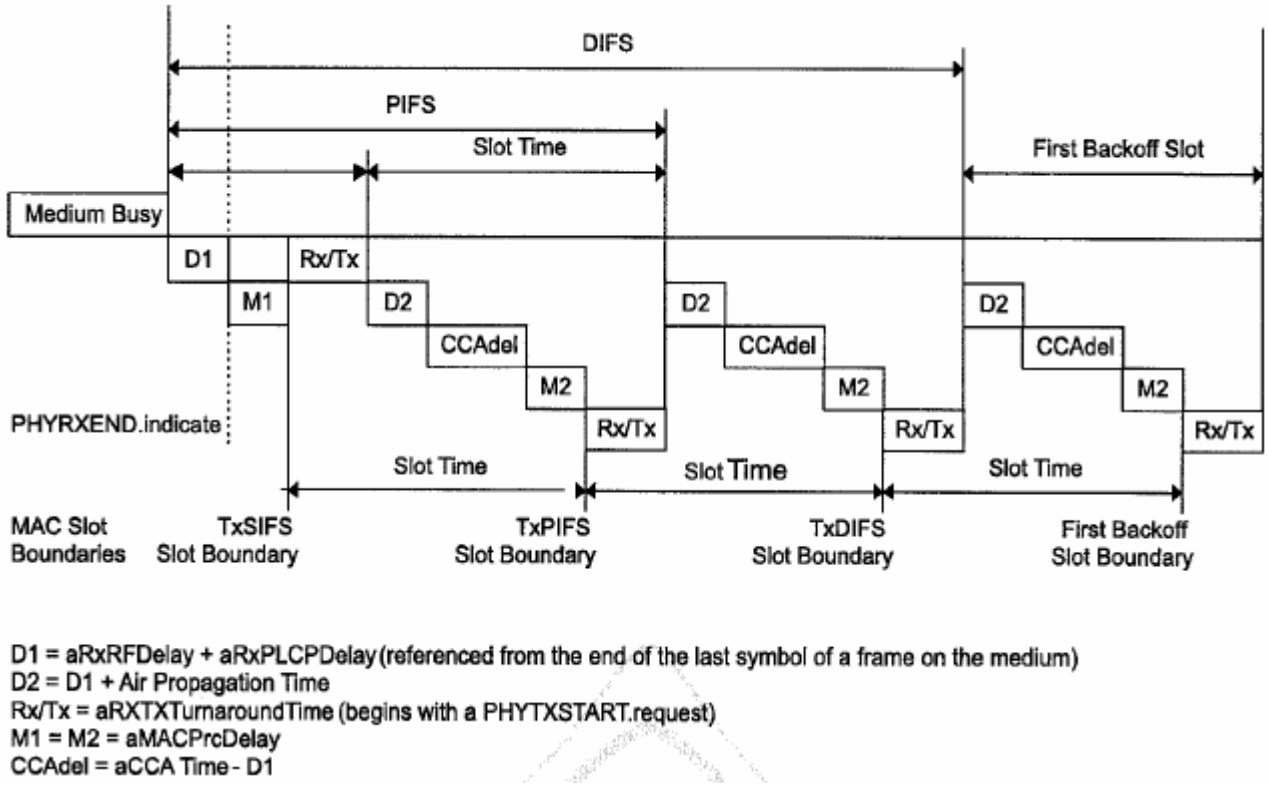


Рис. 27. Временные соотношения DCF.

aSIFSTime и aSlotTime определены в MIB и фиксированы для PHY.

aSIFSTime = aRxRFDelay + aRxPLCPDelay + aMACPrcDelay + aRxTxTurnaroundTime

aSlotTime = aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay

PIFS и DIFS получаются из следующих уравнений, как показано на Рис. 27:

PIFS = aSIFSTime + aSlotTime

DIFS = aSIFSTime + 2 x aSlotTime

EIFS получается из SIFS, DIFS и длины времени передачи управляющего фрейма ACK на скорости 1 Мбит/с по следующему уравнению:

EIFS = aSIFSTime + (8 x ACKSize) + aPreambleLength + aPLCPHeaderLength + DIFS,

где

ACKSize – длина в байтах фрейма ACK;

(8 x ACKSize) + aPreambleLength + aPLCPHeaderLength – время в микросекундах, необходимое для передачи на наименьшей разрешенной скорости PHY.

Рис. 27 показывает соотношения между SIFS, PIFS и DIFS, измеренные в среде, и различные границы слотов TxSIFS, TxPIFS и TxDIFS. Эти границы определяют, когда MACдолжен включать передатчик, чтобы удовлетворять спецификациям IFS, после обнаружения последующего результата CCA для предыдущего фрейма.

Следующие уравнения определяют границы слотов MAC с помощью атрибутов, определенных в MIB, которые предназначены для компенсации зависимости от времени. Начало границы этих слотов – это конец последнего символа предыдущего фрейма в среде.

TxSIFS = SIFS – aRxTxTurnaroundTime

TxPIFS = TxSIFS + aSlotTime

TxDIFS = TxSIFS + 2 x aSlotTime

Допустимые отклонения указаны в PHY MIB и должны назначаться только спецификациям SIFS, так что отклонения не накапливаются.

9.3 *PCF*

PCF обеспечивает свободную передачу фреймов. PC должен находиться в AP. Для AP является необязательным то, чтобы она могла становиться PC. На самом деле все STA подчиняются правилам доступа к среде через PCF, поскольку эти правила базируются на DCF, поэтому они устанавливают свои NAV в начале каждого CFP. Рабочие характеристики PCF таковы, что все STA могут правильно работать в присутствии BSS, в котором есть PC, и, если они ассоциированы с точечно координированной BSS, то они могут принимать все фреймы, посылаемые под управлением PCF. Для STA также существует опция, когда она может ответить на свободный запрос (CF-Poll), принятый от PC. STA, которая может отвечать на CF-Poll, считается CF-опрашиваемой, и может опрашиваться активным PC. CF-опрашиваемые STA и PC не используют RTS/CTS в CFP. При опросе от PC CF-опрашиваемая STA может передать только одни MPDU, которые могут быть предназначены кому угодно (кроме самого PC), а также передать подтверждение фрейма, принятого от PC, используя частные подтипы фреймов данных для такой передачи. Если фрейм данных не подтверждается, CF-опрашиваемая STA не должна передавать его повторно до тех пор, пока она не будет снова опрошена PC, либо пока она не решит осуществить повторную передачу в течение CP. Если адресованный приемник CF передачи не является CF-опрашиваемым, такая STA подтверждает передачу с помощью правил подтверждения DCF, а PC продолжает управлять средой. PC может использовать свободную передачу фреймов исключительно для доставки фреймов на STA, но никогда для опроса CF-неопрашиваемых STA.

PC может выполнять backoff повторной передачи неподтвержденного фрейма в течение CFP. PC, который поддерживает опросный список, может попытаться повторно передать неподтвержденный фрейм в следующий раз, когда соответствующий AID окажется в начале списка.

PC может повторно передать неподтвержденный фрейм в течение CFP после времени PIFS.

Если на одном и том же физическом канале в перекрывающемся пространстве работает более одного точечно скоординированного BSS, возможно возникновение коллизий между передачей PCF независимыми PC. Правила, при которых могут сосуществовать несколько перекрывающихся точечно скоординированных BSS, приведены в 9.3.3.2. Как показано на Рис. 16, PCF стоит сверху CSMA/CA-базированного DCF, используя приоритеты доступа, обеспечиваемые данной схемой. Активный PC должен быть расположен в AP, которая ограничивает работу PCF в инфраструктурных сетях. PCF активируется на PC-способной AP путем установки в ненулевое значение параметра CFPMaxDuration в CF Parameter Set примитива MLMEStart.request.

Фреймы данных, посылаемые в течение DCF, должны использовать только подтипы данных Data и Null Function. Фреймы данных, посылаемые PC, или в ответ на опрос PC, в течение CFP должны использовать соответствующие подтипы данных по следующим правилам:

- Data+CF-Poll, Data+CF-Ack+CF-Poll, CF-Poll и CF-Ack+CF-Poll должны посылаться только от PC.
- Data, Data+CF-Ack, Null Function и CF-Ack могут посылаться от PC либо от любой CF-опрашиваемой STA.

STA, принимающие фреймы типа Data, должны рассматривать в качестве основы для индикации на LLC только тело фрейма, если фрейм имеет подтип Data, Data+CF-Ack, Data+CF-Poll или Data+CF-Ack+CF-Poll. CF-опрашиваемые STA должны интерпретировать все биты подтипа принятых фреймов Data для целей CF, но должны просматривать только тело фрейма, если фрейм имеет подтип Data, Data+CF-Ack, Data+CF-Poll или Data+CF-Ack+CF-Poll.

9.3.1 Структура и синхронизация CFP

Управляющие фреймы PCF передаются в течение CFP. CFP должен чередоваться с CP при передаче управляющих фреймов DCF, как показано на Рис. 28. Каждый CFP должен начинаться с маякового фрейма, который содержит элемент DTIM (далее просто “DTIM”). CFP должны повторяться с определенной скоростью, которая должна быть синхронизирована с маяковым интервалом, как определено в последующих параграфах.

PC генерируют CFP со скоростью повторения свободного соединения (CFPRate), которая определяется как количество интервалов DTIM. PC должен определять используемую CFPRate (изображена как повторяющийся интервал на иллюстрациях ниже), исходя из параметра CFPRate в наборе CF Parameter Set. Это значение, в единицах интервалов DTIM, должно быть сообщено остальным STA данной BSS в поле CFPPeriod элемента CF Parameter Set в маяковых фреймах. Элемент CF Parameter Set должен присутствовать только в маяковых фреймах и фреймах ответа доступа, передаваемых от STA, содержащей активный PC.

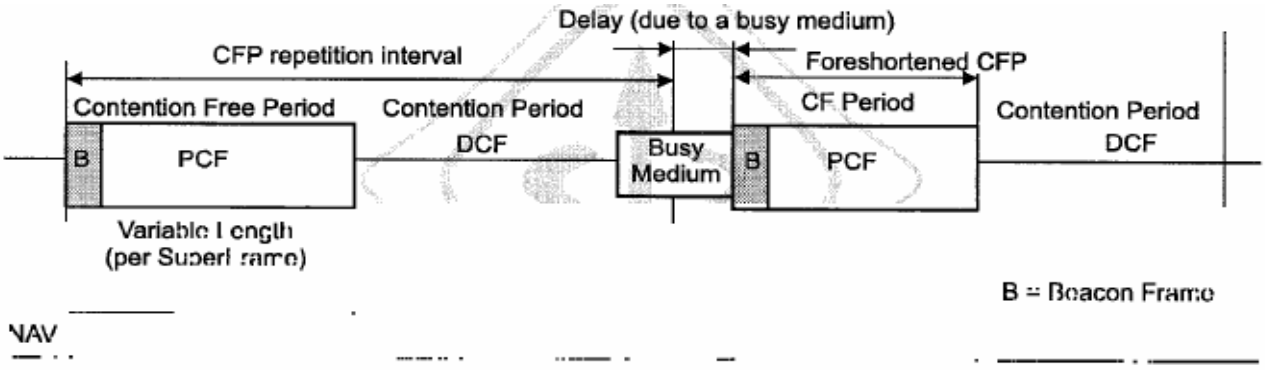


Рис. 28. Чередование CFP/CP.

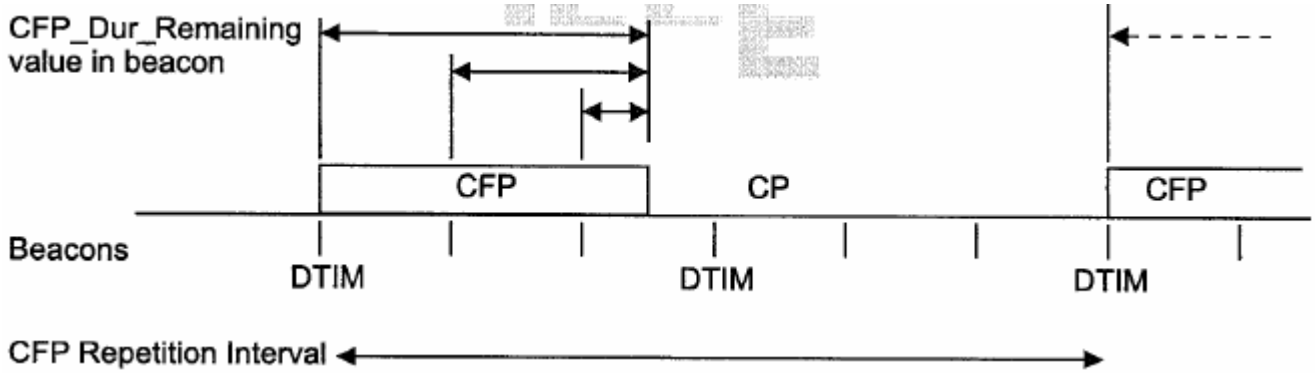


Рис. 29. Маяки и CFP.

Длина CFP управляется PC с максимальной длительностью, определяемой значением параметра CFPMaxDuration в наборе CF Parameter Set. Ни максимальная, ни действительная длительность (сигнализируемая передачей фрейма Control с подтипом CF-End или CF-End+Ack от PC) не ограничена кратностью интервала маяка. Если длительность CFP больше интервала маяка, то PC должен передавать маяки в соответствующие моменты времени в течение CFP (при условии задержки из-за наличия трафика в номинальное время, как и для всех маяков). Элемент CF Parameter Set всех маяков в начале или в течение CFP должен содержать ненулевое значение в поле CFPDurRemaining. Это значение, в единицах TU, должно указывать максимальное время от передачи данного маяка до конца данного CFP. Значение поля CFPDurRemaining должно быть нулевым во фреймах, посылаемых в течение CP. Пример такой взаимосвязи показан на Рис. 29, где приведен случай, когда CFP занимает два интервала DTIM, интервал DTIM занимает три интервала маяка, а значение aCFPMaxDuration – примерно 2,5 интервала маяка.
PC может оборвать любой CFP в момент или до истечения aCFPMaxDuration, основываясь на доступном трафике и размере опросного списка. Поскольку передача любого маяка может быть задержана относительно номинального времени из-за условия занятости среды, CFP может быть укорочен на величину этой задержки. Если среда занята из-за наличия трафика DCF, маяк должен быть задержан на время, необходимое для завершения обмена фреймами DCF. В том случае, если передача маяка задерживается, значение CFPDurRemaining в маяке в начале CFP должно указывать время окончания CFP, которое не позже, чем TBTT плюс значение aCFPMaxDuration. Это показано на Рис. 30.

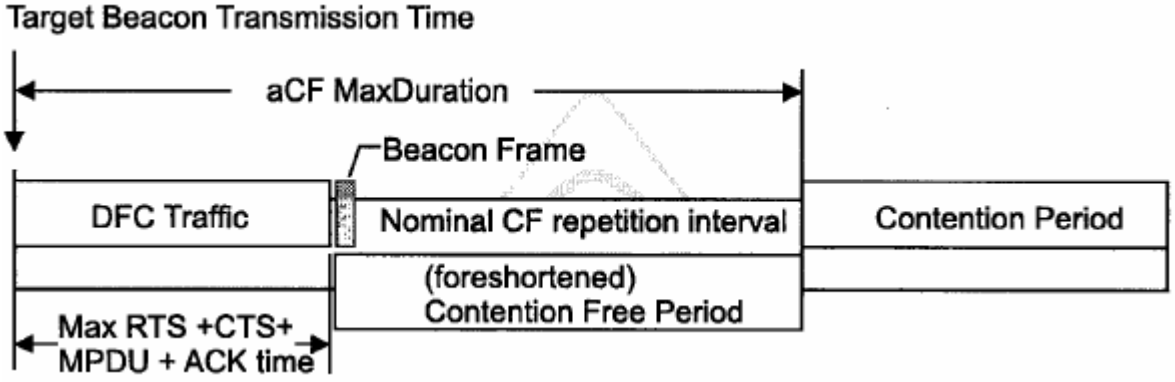


Рис. 30. Пример задержки маяка и укорачивания CFP.

9.3.2 Процедура доступа PCF

Протокол передачи со свободным соединением базируется на схеме опроса, управляемой с PC, работающим на AP в BSS. PC получает управление средой в начале CFP и пытается поддержать управление всем CFP, ожидая между передачами более короткое время, чем STA, использующие процедуру доступа DCF. Все STA в IBSS (кроме PC) устанавливают свои NAV в значение CFPMaxDuration в номинальное время начала каждого CFP. При этом соединение практически отсутствует, так как предотвращены не опросные передачи STA независимо от того, являются ли они CF-опрашиваемыми. Подтверждение фреймов, посылаемых в течение CFP, может быть выполнено с помощью фреймов Data+CF-Ack, CF-Ack, Data+CF-Ack+CF-Poll (только от PC) или CF-Ack+CF-Poll (только от PC) в тех случаях, когда фрейм Data (или Null) следует сразу за подтверждаемым, таким образом, можно избежать перекрытия отдельных управляющих фреймов ACK. Фреймы подтверждения не-CF-опрашиваемых или не опрашиваемых CF-опрашиваемых STA в течение CFP используют процедуру DCF ACK.

9.3.2.1 Основной доступ

В номинальное время начала каждого CFP PC должен контролировать среду. Если установлено, что среда свободна на один период PIFS, PC должен передать маяковый фрейм, содержащий элементы CF Parameter Set и DTIM.
После начального маякового фрейма PC должен ожидать по меньшей мере один период SIFS, а затем передать одно из следующего: фрейм данных, фрейм CF-Poll, фрейм Data+CF-Poll либо фрейм CF-End.

Если CFP является нулевым, т.е., у PC нет буферизированного трафика и очередных опросов, то сразу после начального маяка должен быть передан фрейм CF-End.
Ожидается, что STA, принимающая направленный безошибочный фрейм от PC, ответит после периода SIFS в соответствии с процедурами передачи, определенными в 9.3.3. Если приемная STA не является CF-опрашиваемой, то ответом на принятый безошибочный фрейм данных всегда должен быть фрейм ACK.

9.3.2.2 Обслуживание NAV в течение CFP

Механизм обработки NAV в течение CFP разработан для улучшения работы CFP перекрывающихся скоординированных инфраструктурных BSS. Механизм, с помощью которого инфраструктурные BSS координируют свои CFP, лежит за пределами обзора данного стандарта.
Каждая STA, за исключением STA с PC, должна предварительно установить свой NAV в значение CFPMaxDuration (полученное из элемента CF Parameter Set маяка данного PC) в каждый момент передачи маяка (TBTT – см. 11), в котором по графику стартует CFP (в зависимости от поля CFPPeriod из элемента CF Parameter Set маяка данного PC). Каждая не-PC STA должна обновлять свой NAV, используя значение CFDurRemaining из элемента CF Parameter Set любого принятого безошибочного маякового фрейма. При этом значение CFDurRemaining может быть принято из элемента CF Parameter Set маякового фрейма от другой (перекрывающейся) BSS.
Такие действия предотвращают захват STA управления средой в течение CFP, что особенно важно в тех случаях, когда CFP занимает несколько интервалов, таких как dwell периоды FH PHY. Кроме того, такой способ обработки NAV уменьшает риск того, что «скрытые» STA посчитают среду свободной на период DIFS в течение CFP и попытаются выйти на передачу.
STA, которая присоединяется к BSS, работающей с PC, должна использовать информацию элемента CFDurRemaining из набора CF Parameter Set любого принятого маякового фрейма или фрейма ответа доступа для обновления своего NAV до начала какой-либо передачи.
PC должен передавать фрейм CF-End или CF-End+ACK в конце каждого CFP. STA, которая принимает любой из этих фреймов от любой BSS, должна сбросить свой NAV.

9.3.3 Процедура передачи PCF

При передаче с использованием PCF фреймы обычно чередуются: от AP/PC и на AP/PC. Порядок этих передач и STA, которым разрешено передавать фреймы на PC в любой данный момент времени, должны управляться PC в течение CFP. На показана типичная передача фреймов в течение CFP. Правила, по которым эти фреймы занимают свое место, детализированы в последующих главах.

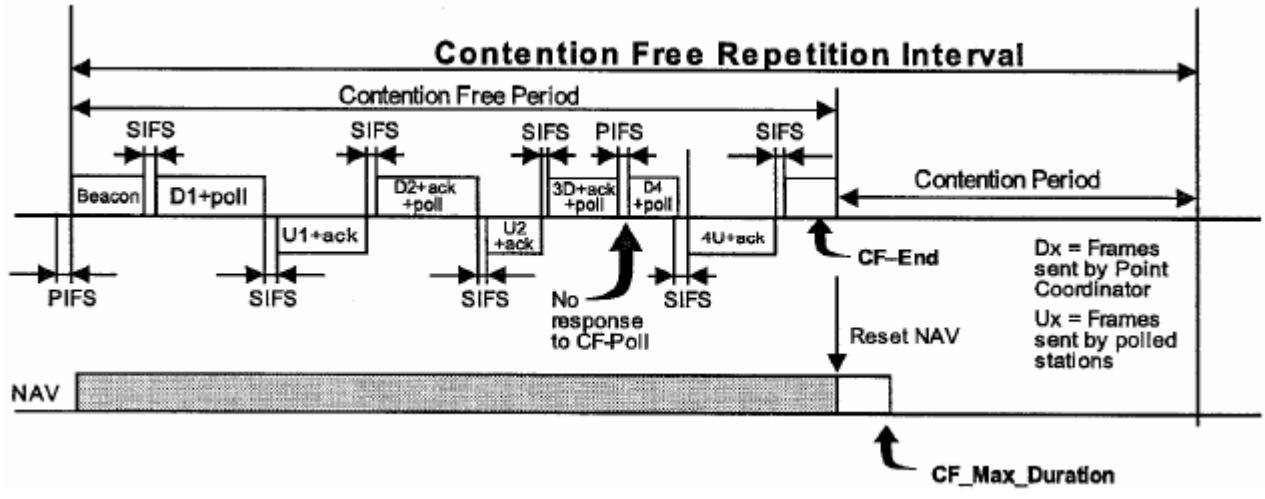


Рис. 31. Пример PCF передачи фреймов.

В STA, имеющей FH PHY, управление каналом теряется на границе времени dwell. Поэтому необходимо, чтобы текущая передача MPDU и соответствующих подтверждений осуществлялась до границы времени dwell. После того, как она будет опрошена PC, и если осталось недостаточно времени до грани-

цы dwell, чтобы передать MPDU плюс подтверждение, STA должна отложить передачу MPDU и передать фрейм Null или CF-Ack. При этом не нужно изменять короткий и длинный счетчики повторной попытки (retry counter).
PC не должен передавать CF-Poll на STA, имеющую FH PHY, если осталось недостаточно времени до границы dwell, чтобы STA могла ответить фреймом Null или CF-Ack.

9.3.3.1 Передача PCF, когда PCF STA является передатчиком или приемником

PC должен передавать фреймы между маяком, с которым начинается CFP, и CF-End, используя SIFS, за исключением тех случаев, когда он ожидает передачу от другой STA, и период SIFS истекает без приема ожидаемой передачи. В таких случаях PC может начать свою следующую подвисшую передачу через один PIFS после окончания своей последней передачи. При этом PC продолжает управлять средой в присутствии перекрывающей BSS. PC может передавать любой из следующих фреймов на CF-опрашиваемую STA:

- Data; используется для посылки данных от PC, когда адресуемый приемник не опрашивается и нет предыдущего фрейма для подтверждения.
- Data+CF-ACK; используется для посылки данных от PC, когда адресуемый приемник не опрашивается и PC нужно подтвердить фрейм, принятый от CF-опрашиваемой STA на SIFS период раньше начала данной передачи.
- Data+CF-Poll; используется для посылки данных от PC, когда адресуемый приемник является следующей STA, которой разрешается передача в течение данного CFP, и нет предыдущего фрейма для подтверждения.
- Data+CF-ACK+CF-Poll; используется для посылки данных от PC, когда адресуемый приемник является следующей STA, которой разрешается передача в течение данного CFP, и PC нужно подтвердить фрейм, принятый от CF-опрашиваемой STA на SIFS период раньше начала данной передачи.
- CF-Poll; используется, когда PC не посылает данных адресуемому приемнику, но адресуемый приемник является следующей STA, которой разрешается передача в течение данного CFP, и нет предыдущего фрейма для подтверждения.
- CF-ACK+CF-Poll; используется, когда PC не посылает данных адресуемому приемнику, но адресуемый приемник является следующей STA, которой разрешается передача в течение данного CFP, и PC нужно подтвердить фрейм, принятый от CF-опрашиваемой STA на SIFS период раньше начала данной передачи.
- CF-ACK; используется, когда PC не посылает данных и не опрашивает адресуемый приемник, но PC нужно подтвердить фрейм, принятый от CF-опрашиваемой STA на SIFS период раньше начала данной передачи (полезен, когда следующей передачей PC является управляющий фрейм, такой как маяк).
- Любой управляющий фрейм, соответствующий AP, который посылается по своим собственным правилам.

PC может передавать данные или фреймы управления на не-CF-опрашиваемые не спящие STA в течение CFP. Эти STA должны подтверждать прием фреймами ACK после SIFS так же, как и при DCF. PC также может передавать broadcast или multicast фреймы в течение CFP. Поскольку маяковый фрейм, который начинает CFP, содержит элемент DTIM, то, в случае наличия STA, использующих режим пониженного потребления, буферизированные broadcast и multicast фреймы должны передаваться сразу после любого маяка, содержащего элемент TIM с полем DTIM count равным нулю.
CF-опрашиваемая STA, принимающая направленный фрейм данных с любым из подтипов, которые включают CF-Poll, может передать один фрейм данных через период SIFS после приема CF-Poll. CF-опрашиваемые STA должны игнорировать, но не сбрасывать свои NAV, когда они выполняют передачу в ответ на CF-Poll.
Не-CF-опрашиваемые STA, принимающие направленный фрейм в течение CFP, должны передавать ACK, но не должны сбрасывать свои NAV.
Для фреймов, которые требуют подтверждения уровня MAC, CF-опрашиваемые STA, принимающие CF-Poll (любого типа), могут выполнить данное подтверждение с помощью подтипа Data+CF-ACK. На-

пример, фрейм U1 на Рис. 31 содержит подтверждение на предшествующий фрейм D1, а фрейм D2 – подтверждение на предшествующий фрейм U1. PC может использовать подтипы CF-ACK для подтверждения принятого фрейма даже тогда, когда фрейм данных, посылаемый с подтипом CF-ACK, адресован другой STA, а не той, которой предназначено подтверждение. CF-опрашиваемые STA, ожидающие подтверждения, должны интерпретировать подтип фрейма (если таковой есть), посылаемого от PC через период SIFS после передачи от STA на PC. Если фрейм, требующий подтверждения уровня MAC, принят не-CF-опрашиваемой STA, эта STA не должна интерпретировать индикацию CF-Poll (если таковая есть) и должна подтвердить фрейм, послав фрейм Ack Control после периода SIFS.

Длины фреймов могут меняться и ограничены только требованиями к длине фрейма и/или фрагмента, назначенными для BSS. Если CF-опрашиваемая STA не отвечает на CF-Poll (любого типа) в течение периода SIFS вслед за передачей от PC, либо если не-CF-опрашиваемая STA не отвечает фреймом ACK в течение периода SIFS вслед за передачей от PC, требующей подтверждения, то PC должна возобновить свое управление и может передать свой следующий фрейм через период PIFS от окончания последней передачи PC.

CF-опрашиваемая STA всегда должна отвечать на CF-Poll, направленный по ее адресу MAC, и принятый без ошибок. Если у STA нет фрейма для посылки в ответ на опрос, она должна ответить фреймом Null. Если у STA нет фрейма для посылки в ответ на опрос, но требуется подтверждение на фрейм, который доставил CF-Poll, ответом должен быть фрейм CF-ACK (без данных). Нулевой ответ необходим для разрешения ситуации «без трафика», чтобы избежать коллизий между перекрывающимися PC.

CFP должен заканчиваться при истечении времени CFPDurRemaining после начала маякового фрейма, вызвавшего CFP, либо когда у PC нет больше ни фреймов для передачи, ни STA для опроса. В любом случае окончание CFP должно быть обозначено передачей CF-End от PC. Если на момент передачи CF-End есть принятый фрейм, требующий подтверждения, PC должен передать фрейм CF-End+ACK. Все STA в BSS, принявшие CF-End или CF-End+ACK, должны сбросить свои NAV, чтобы они могли попытаться начать передачу в течение CP.

9.3.3.2 Работа с перекрывающимися точечно скоординированными BSS

Поскольку PCF работает без рандомизации окна соединения CSMA/CA и без backoff DCF, существует риск повторяющихся коллизий в том случае, если несколько перекрывающихся точечно скоординированных BSS работают на одном физическом канале, и их CFP скорости и интервалы маяков примерно равны. Для минимизации риска значительной потери фреймов из-за CF коллизий PC должен использовать случайную backoff задержку (с CW в диапазоне от 1 до aCWMin) для запуска CFP, когда начальный маяк задерживается из-за занятости среды. PC может необязательно использовать такой backoff в течение CFP перед повторной передачей неподтвержденных направленных данных или фреймов управления.

Для дополнительного уменьшения чувствительности к коллизиям PC должен требовать, чтобы среда считалась свободной на период DIFS плюс случайное (в диапазоне от 1 до aCWMin) количество слотов по одному на каждый aMediumOccupancyLimit TU в течение CFP. При этом достигается потеря контроля над средой перекрывающихся BSS или скрытого трафика STA, поскольку STA в данном BSS запрещена передача путем установки их NAV в CFPMaxDuration или CFPDurRemaining. Для работы PCF совместно с FH PHY aMediumOccupancyLimit должен быть установлен в значение времени dwell. Для работы с другими типами PHY aMediumOccupancyLimit может быть установлен равным CFPMaxDuration, если не требуется дополнительная защита от PCF коллизий. AMediumOccupancyLimit также пригоден для согласования в управляющих доменах, которые накладывают ограничения на непрерывное время передачи одиночной STA как часть «спектрального этикета».

9.3.3.3 Ограничение CFPMaxDuration

Значение CFPMaxDuration должно быть ограничено для возможности сосуществования трафика contention и contention-free.

Минимальным значением CFPMaxDuration является два раза по MaxMPDUTime плюс время, необходимое для посылки начального маякового фрейма и фрейма CF-End для CFP. При этом может остаться достаточно времени для AP, чтобы послать фрейм данных на STA, опрашивая ее, и для ответа опрашиваемой STA одним фреймом данных.

Максимальным значением CFPMaxDuration является длительность (BeaconPeriod x DTIMPeriod x CFPRate) минус (MaxMPDUTime + (2 x aSIFSTime) + (2 x aSlotTime) + (8 x ACKSize)), выраженная в микросекундах, при работе с окном соединения CWMin. При этом может остаться достаточно времени для посылки как минимум одного фрейма данных в течение CP.

9.3.3.4 Правила использования свободного соединения

PC может посылать broadcast/multicast фреймы, а также направленные фреймы данных или фреймы управления любой активной STA так же, как и CF-опрашиваемой PS STA. В течение CFP CF-опрашиваемые STA должны подтверждать (через период SIFS) прием каждого фрейма Data+CF-Poll или Data+CF-ACK+CF-Poll, используя фреймы Data+CF-ACK или CF-ACK (без данных); прием каждого CF-Poll (без данных), используя Data или Null (без данных), и прием всех остальных данных и фреймов управления, используя фреймы ACK Control. Не-CF-опрашиваемые STA должны подтверждать прием фреймов данных и фреймов управления, используя фреймы ACK Control, посылаемые после периода SIFS. Такая работа аналогична той, что эти STA выполняют для DCF.

При опросе PCF (Data+CF-Poll, Data+CF-ACK+CF-Poll, CF-Poll или CF-ACK+CF-Poll) CF-опрашиваемая STA может послать один фрейм данных в любое место назначения. Такой фрейм, направленный на или через PC STA, должен быть подтвержден PC с помощью индикации CF-ACK (Data+CF-ACK, Data+CF-ACK+CF-Poll, CF-ACK, CF-ACK+CF-Poll или CF-End+CF-ACK), посылаемой после SIFS. Такой фрейм, направленный на не-CF-опрашиваемую STA, должен быть подтвержден фреймом ACK Control, посылаемым после периода SIFS. Опрошенная CF-опрашиваемая STA, у которой нет для посылки ни фрейма данных, ни подтверждения, должна ответить фреймом Null после периода SIFS. Опрошенная CF-опрашиваемая STA, которой не хватает времени до окончания CFP или текущего предела занятости среды, чтобы послать свои очередные MPDU и принять подтверждение, должна ответить фреймом Null, либо фреймом CF-ACK, если опрос осуществлялся с помощью Data+CF-Poll или Data+CF-ACK+CF-Poll, после периода SIFS. CF-опрашиваемая STA может установить бит More Data в своем ответе для того, чтобы PC мог различить пустую очередь STA и ответ из-за недостаточного времени для передачи MPDU.

PC не должен передавать фреймы с подтипами CF-Poll, если в текущем CFP осталось недостаточно времени для того, чтобы опрашиваемая STA передала фрейм данных с минимальной длиной MPDU.

9.3.4 Опросный список свободного соединения

Если PC поддерживает CFP для возвратной передачи фреймов так же, как и для их доставки, PC должен поддерживать «опросный список», используемый для выбора STA, подходящих для приема CF-Poll в течение CFP. Функциональные характеристики опросного списка определены ниже. Если PC поддерживает CFP исключительно для доставки фреймов, то ему не нужен опросный список, и он никогда не должен генерировать фреймы данных с подтипами CF-Poll. Тип поддержки свободного соединения, обеспечиваемый PC, определяется полем Capability Information в маяке, ответе ассоциации, ответе реассоциации и ответе доступа, которые посылаются от AP. Любые подобные фреймы, посылаемые от STA, должны всегда содержать эти биты установленными в ноль так же, как в не инфраструктурных сетях. Опросный список используется для опроса CF-опрашиваемых STA независимо от того, имеет ли PC подвисший трафик для этих STA. Опросный список может использоваться (с типами Data+CF-Poll и Data+CF-ACK+CF-Poll) для управления передачей фреймов данных от CF-опрашиваемых STA на PC. Опросный список является логической конструкцией, которая не выходит за пределы PC. Минимальным требованием к технике поддержки опросного списка является требование гарантированного взаимодействия случайных CF-опрашиваемых STA в BSS, управляемых случайными точками доступа с активными PC. AP могут также обеспечивать дополнительные возможности опросного списка, лежащие за пределами обзора данного стандарта.

9.3.4.1 Обработка опросного списка

PC должен посылать CF-Poll как минимум одной STA в течение CFP, если существуют записи в опросном списке. В течение каждого CFP PC должен передавать запросы подмножеству STA из опросного списка в восходящем порядке значений их AID.

Если доставка всех CF фреймов завершена, все STA из опросного списка уже опрошены, но в CFP еще осталось время, PC может сгенерировать один или несколько CF-Poll для любых STA из списка. Если доставка всех CF фреймов завершена, все STA из опросного списка уже опрошены, но в CFP еще осталось время, PC может послать данные или фреймы управления на любые STA.
Для того, чтобы увеличить эффективность CFP и вероятность возврата подтверждений успешных фреймов данных в обратном направлении, PC следует в основном использовать типы Data+CF-Poll и Data+CF-ACK+CF-Poll для каждого передаваемого фрейма данных до тех пор, пока остается достаточно времени для потенциального ответа на CF-Poll, оставшиеся в CFP.

9.3.4.2 Процедура обновления опросного списка

STA указывает свою CF-опрашиваемость в подполе CF-Pollable поля Capability Information во фреймах запроса ассоциации или реассоциации. Если STA хочет изменить запись PC о своей CF-опрашиваемости, она должна выполнить реассоциацию. В течение ассоциации CF-опрашиваемая STA может также запросить размещение в опросном списке на время своей ассоциации, либо это может быть сделано путем установки подполя CF-Poll Request в поле Capability Information. Если CF-опрашиваемая STA не хочет находиться в опросном списке, она должна выполнить ассоциацию, установив подполе CF-Pollable в false, а CF-Poll Request – в true. Никогда не быть опрашиваемыми полезно для тех STA, которые обычно используют режим пониженного потребления; при этом они могут принимать буферизированный трафик в течение CFP (поскольку они должны просыпаться для приема DTIM, с которого начинается CFP), однако не должны оставаться разбуженными для приема CF-Poll, когда у них нет трафика для передачи. Если STA хочет удалить себя из опросного списка, она должна выполнить реассоциацию.
CF-опрашиваемые STA, которых нет в опросном списке, но которые не запрашивали своего отсутствия в опросном списке, могут быть динамически включены туда PC для соответствующей обработки.

9.4 *Фрагментация*

MAC может фрагментировать и реассемблировать направленные MSDU и MMPDU. Механизмы фрагментации и дефрагментации позволяют осуществлять повторную передачу фрагментов.
Длина фрагмента MPDU должна быть равна количеству байт во всех фрагментах, за исключением последнего, который может быть меньше. Длина фрагмента MPDU всегда должна быть четным количеством байт, за исключением последнего, который может быть четным или нечетным. Длина фрагмента никогда не должна быть больше aFragmentationTreshold, пока для MPDU не будет вызван WEP. Если WEP активирован для MPDU, то MPDU должны быть расширены IV и ICV (см. 8.2.5); при этом фрагмент может превысить aFragmentationTreshold.
При передаче данных количество байт во фрагменте (до WEP обработки) должно определяться aFragmentationTreshold и количеством байт MPDU, которые уже назначены фрагменту постоянно при его сборке в первый раз. Как только фрагмент передан в первый раз, содержимое тела его фрейма и длина должны быть постоянными до тех пор, пока он не будет успешно доставлен ближайшей принимающей STA. STA должна быть способной принимать фрагменты произвольной длины.
Если требуется повторная передача фрагмента, содержимое тела его фрейма и длина должны оставаться постоянными в течение всего времени жизни MSDU или MMPDU на данной STA. После того, как фрагмент однажды передан, его содержимое и длина не должны изменяться для подстройки к временным границам dwell. Каждый фрагмент должен содержать поле Sequence Control, которое состоит из последовательного номера и номера фрагмента. Когда STA передает MSDU или MMPDU, последовательный номер должен оставаться постоянным для всех фрагментов данных MSDU или MMPDU. Фрагменты должны посылаться в восходящем порядке, от фрагмента с наименьшим номером к фрагменту с наибольшим номером, причем номер фрагмента начинается с нуля и увеличивается на единицу с каждым успешным фрагментом. Поле Frame Control также содержит бит More Fragments, который в случае равенства нулю указывает на последний (либо единственный) фрагмент MSDU или MMPDU.
STA-источник должна поддерживать таймер передачи MSDU для каждых передаваемых MSDU. Атрибут aMaxTransmitMSDULifetime определяет максимальное количество времени, разрешенное для передачи MSDU. Таймер запускается при попытке передачи первого фрагмента MSDU. Если таймер превы-

шает aMaxTransmitMSDULifetime, то все оставшиеся фрагменты уничтожаются STA-источником, и не делается никаких попыток завершить передачу MSDU.

9.5 *Дефрагментация*

Каждый фрагмент содержит информацию, позволяющую завершить реассемблирование MSDU или MMPDU из составных фрагментов. Заголовок каждого фрагмента содержит следующую информацию, которая используется STA-приемником для реассемблирования MSDU или MMPDU:

- Тип фрейма
- Адрес передатчика, получаемый из поля Address2
- Адрес назначения
- Поле Sequence Control: это поле позволяет STA-приемнику проверить, принадлежат ли все входящие фрагменты одним и тем же MSDU или MMPDU, а также установить последовательность, в которой фрагменты должны быть реассемблированы. Последовательный номер внутри поля Sequence Control остается постоянным для всех фрагментов MSDU или MMPDU, в то время как номер фрагмента инкрементируется для каждого фрагмента.
- Индикатор More Fragments: указывает STA-приемнику на то, что это не последний фрагмент MSDU или MMPDU. Только у последнего или единственного фрагмента MSDU или MMPDU это бит должен быть установлен в ноль. У всех остальных фрагментов MSDU или MMPDU это бит должен быть установлен в единицу.

STA-приемник должна реконструировать MSDU или MMPDU, комбинируя фрагменты в соответствии с их номерами в поле Sequence Control. Если фрагменту был назначен WEP, он должен быть дешифрован перед использованием для дефрагментации MSDU или MMPDU. Если фрагмент с битом More Fragments, установленным в ноль, еще не принят, STA-приемник знает, что MSDU или MMPDU еще не закончились. Как только STA примет фрагмент с битом More Fragments, установленным в ноль, она узнает, что больше не может быть принято никаких фрагментов MSDU или MMPDU.

Все STA должны поддерживать параллельный прием фрагментов как минимум трех MSDU или MMPDU. Заметим, что STA, принимающая более трех фрагментированных MSDU или MMPDU одновременно, может значительно увеличить количество отраженных фреймов.

STA-приемник должна поддерживать Receive Timer для каждых принимаемых MSDU или MMPDU, как минимум для трех. STA может поддерживать дополнительные таймера для приема параллельных MSDU или MMPDU. Приемная STA должна отражать все фрагменты, которые являются частью MSDU или MMPDU, для которых не поддерживается таймер. Также существует атрибут aMaxReceiveLifetime, который определяет максимальное количество времени, разрешенное для приема MSDU. Таймер приема MSDU или MMPDU запускается после приема первого фрагмента MSDU или MMPDU. Если таймер превышает aMaxReceiveLifetime, то все принятые фрагменты данных MSDU или MMPDU отражаются STA-приемником. Если после того, как таймер превысил aMaxReceiveLifetime, принимаются дополнительные фрагменты направленных MSDU или MMPDU, эти фрагменты должны быть подтверждены и отражены.

Для правильного реассемблирования MPDU в MSDU или MMPDU STA-приемник должна отражать любые дубликаты принятых фрагментов. STA должна отражать дубликаты, как описано в 9.2.9. Тем не менее, на фрагмент-дубликат должно посылаться подтверждение.

9.6 *Поддержка нескольких скоростей*

Некоторые PHY имеют возможность многоскоростной передачи данных, которая позволяет увеличить динамический диапазон и улучшить качество. Алгоритм переключения скоростей находится за пределами обзора данного стандарта, однако, для уверенного взаимодействия с многоскоростными PHY данный стандарт определяет набор правил, которым должны следовать все STA.

Все управляющие фреймы должны передаваться на одной из скоростей BSSBasicRateSet (см. 10.3.10.1), либо на одной из скоростей из обязательного набора PHY так, чтобы они были понятны всем STA.

Все фреймы с multicast и broadcast RA должны передаваться на одной из скоростей BSSBasicRateSet, независимо от их типа.

MPDU данных и/или управления с уникальным адресом должны посылаться на любой поддерживаемой скорости, выбираемой с помощью механизма переключения скоростей (результатом которого является значение внутренней переменной MAC, называемой MACCurrentRate, определяемой в единицах 500 кбит/с, которая используется для вычисления поля Duration/ID каждого фрейма). STA не должна передавать на скорости, если известно, что она точно не поддерживается STA-приемником, как указывается в элементе поддерживаемых скоростей во фреймах управления. Для фреймов типов Data+CF-ACK, Data+CF-Poll+CF-ACK и CF-Poll+CF-ACK выбранная скорость передачи фрейма должна поддерживаться как адресуемой приемной STA, так и STA, которая ожидает ACK.

Ни при каких условиях STA не должна начинать передачу фреймов данных или управления на скорости, большей чем наибольшая скорость в OperationalRateSet, параметре примитива MLME-JOIN.request.

Для того, чтобы позволить передающей STA вычислить содержимое поля Duration/ID, отвечающая STA должна передавать свой фрейм Control Response (CTS либо ACK) на той же скорости, что и предыдущий фрейм из последовательности обмена (как определено в 9.7), если данная скорость есть в обязательном наборе PHY, либо, иначе, на наибольшей возможной скорости из обязательного набора PHY в BSSBasicRateSet.

9.7 *Последовательности обмена фреймами*

Допустимые последовательности обмена фреймами приведены в Таблица 52 и Таблица 53. Пояснения обозначений применимы к обеим таблицам.

Таблица 52. Последовательности фреймов.

Последовательность	Количество фреймов	Использование
Data (bc/mc)	1	Broadcast или multicast MSDU
Mgmt (bc)	1	Broadcast MMPDU
{RTS – CTS –} [Frag – ACK –] Last – ACK	2	Направленные MSDU или MMPDU
PS-Poll – ACK	2	Задержанный ответ PS-POLL
PS-Poll – [Frag – ACK –] Last – ACK	3	Немедленный ответ PS-POLL
DTIM(CF) – [≤CF-Sequence>–] {CF-End}	2 или более	Начало CFP
[<CF-Sequence>–] {CF-End}	2 или более	Продолжение CFP после отсутствия Ack или границы занятости среды

Таблица 53. Последовательности CF фреймов.

CF последовательность	Количество фреймов	Использование
Beacon (CF)	1	Маяк в течение CFP
Data (bc/mc)	1	Broadcast или multicast MSDU
Mgmt (bc)	1 или 2	Broadcast MMPDU
Mgmt (dir) – ACK	2 или 3	Направленные MMPDU
Data (dir) + CF-Poll{+CF-Ack} – Data (dir) + CF-Ack – {CF-Ack(no data)}	2	Опрос и ACK, посылаемые вместе с MPDU
Data (dir) + CF-Poll{+CF-Ack} – {CF-Ack(no data)}	2	Опрос STA с пустой очередью, недостаточное время для очередных MPDU или слишком малое количество времени для передачи фрейма, оставшееся до dwell или границы занятости среды
CF-Poll(no data){+CF-Ack} – Data (dir) – {CF-Ack(no data)}	2	Отдельный опрос, ACK, посылаемый вместе с MPDU
CF-Poll(no data){+CF-Ack} – Data (dir) – ACK	3	Опрашиваемые STA передают на STA в BSS

CF последовательность	Количество фреймов	Использование
CF-Poll(no data){+CF-Ack} – Null(no data)	2	Отдельный опрос, пустая очередь STA или недостаточное время для очередных MPDU, или слишком малое количество времени для передачи фрейма, оставшееся до dwell или границы занятости среды
Data (dir) {+CF-Ack} – ACK	2	ACK, если не-CF-опрашиваемая или не опрашиваемая

Пояснения к Таблица 52 и Таблица 53.

1 – пункты, заключенные в скобки […], могут включаться в последовательность ноль и более раз.
2 – пункты, заключенные в скобки {…}, могут включаться в последовательность ноль или один раз.
3 – отдельное тире «–» представляет интервал SIFS, разделяющий пару фреймов.
4 – “Data (bc/mc)” представляет любой фрейм типа Data с broadcast или multicast адресом в поле Address1.
5 – “Mgmt (bc)” представляет любой фрейм типа Management с broadcast адресом в поле DA.
6 – “RTS” представляет фрейм Control с подтипом RTS.
7 – “CTS” представляет фрейм Control с подтипом CTS.
8 – “ACK” представляет фрейм Control с подтипом ACK.
9 – “Frag” представляет MPDU типа Data или MMPDU типа Management с индивидуальным адресом в поле Address1, у которого поле More Fragments установлено в “1”.
10 – “Last” представляет MPDU типа Data или MMPDU типа Management с индивидуальным адресом в поле Address1, у которого поле More Fragments установлено в “0”.
11 – “PS-Poll” представляет фрейм Control с подтипом PS-Poll.
12 – “DTIM(CF)” представляет фрейм управления с подтипом Beacon, содержащим информационный элемент DTIM с ненулевым значением поля CFDurRemaining его элемента Parameter Set.
13 – “CF-End” представляет фрейм Control типа CF-End, либо типа CF-End+Ack (если последним фреймом, предшествующим <CF-последовательности>, был направленный фрейм данных или управления, требующий подтверждения от AP).
14 – “Beacon(CF)” представляет фрейм управления с подтипом Beacon с ненулевым значением поля CFDurRemaining его элемента Parameter Set.
15 – “Data(dir)” представляет любые MPDU типа Data с индивидуальным адресом в поле Address1.
16 – “Mgmt(dir)” представляет любые MMPDU типа Management с индивидуальным адресом в поле Address1.
17 – “CF-ACK(no data)” представляет фрейм данных с подтипом CF-ACK(no data).
18 – “CF-Poll(no data)” представляет фрейм данных с подтипом CF-Poll(no data).
19 – “Null(no data)” представляет фрейм данных с подтипом Null Function (no data).
20 – обозначение “{+CF-Ack}” указывает, что фрейм может включать или не включать подтверждение периода свободной связи.
21 – обозначение “+CF-Ack” указывает, что фрейм включает подтверждение периода свободной связи.
22 – обозначение “+CF-Poll” указывает, что фрейм включает опрос периода свободной связи.
23 – <CF-Sequence> представляет последовательность из одного или более фреймов, посылаемы в течение CFP. Действительная <CF-последовательность> должна состоять из одной из фреймовых последовательностей, представленных в Таблица 53. Набор последовательностей обмена фреймами, соответствующий [<CF-Sequence>], может быть произвольным в течение CFP.

Отдельные фреймы внутри каждой из данных последовательностей разделяются SIFS.

9.8 *Ограничения на передачу MSDU*

Во избежание переупорядочивания MSDU между парами устройств LLC и/или нежелательного отражения MSDU вводятся следующие ограничения, которые должны соблюдаться любой STA, которая может параллельно обрабатывать несколько отдельных MSDU. Заметим, что термин «отдельные» относится к MSDU или MMPDU, которые передаются в разное время. STA может иметь любое количество (больше или равное единице) таких MSDU, соблюдая ограничения, приведенные ниже.

STA должна быть уверена, что в данный конкретный момент времени существует не более одного MSDU или MMPDU от частного SA для отдельного частного RA. Заметим, что проще, но более строго поддерживать в данный конкретный момент времени существует не более одного MSDU для отдельного частного RA.

STA, у которой реализован необязательный класс обслуживания StrictlyOrdered, должна быть уверена, что у нее нет группо-адресованных (multidestination) MSDU класса StrictlyOrdered от SA, для которой есть другие MSDU (направленные либо группо-адресованные).

Рекомендуется, чтобы STA выбирала значение aMaxMSDUTransmitLifetime, которое достаточно велико, чтобы STA не отражала MSDU из-за истечения тайм-аута Transmit MSDU при нормальных рабочих условиях.

10 Управление уровнями

10.1 *Краткий обзор модели управления*

MAC и физические уровни концептуально включают объекты управления, называемые объектами управления подуровня MAC и объектами управления подуровня PHY (MLME и PLME, соответственно). Эти объекты обеспечивают интерфейсы обслуживания управления уровня, через которые могут быть вызваны функции управления уровня.

Чтобы обеспечивать правильное функционировние MAC, объект управления станцией (SME) должен присутствовать в пределах каждой STA. SME - независимый от уровня объект, который может рассматриваться как принадлежащий отдельному уровню управления, или как принадлежащий "off to the side". Точные функции SME не определены в этом стандарте, но вообще этот объект может рассматриваться ответственным за такие функции как сбор зависимого от уровня статуса от различных объектов управления уровня и установка значений определенных параметров уровня. Типовой SME исполняет функции общих объектов управления системы и осуществит стандартные протоколы управления. Рис. 11 изображает отношения среди объектов управления. Различные объекты в пределах этой модели взаимодействуют различными способами. В пределах этого стандарта определены явно взаимодействия, через точку доступа обслуживания (SAP), через которую производится обмен определенными примитивами. Другие взаимодействия явно не определены в пределах этого стандарта: типы интерфейсов между MAC и MLME и между PLCP и PLME представлены на Рис. 11 как двойные стрелки. Определенная манера, в которой эти MAC и PHY объекты управления объединены в полный MAC и PHY слои, не определена в пределах этого стандарта.

В пределах этой модели имеются следующее SAP управления:

- SME-MLME SAP
- SME-PLME SAP
- MLME-PLME SAP

Последние две SAP поддерживают идентичные примитивы, и фактически могут рассматриваться как одна SAP (называемый SAP PLME) которая может использоваться непосредственно MIME или SMB. Таким образом, модель отражает то, что является общим подходом реализации, в котором функции PLME управляются MLMK (от имени SME). В частности реализация PHY не требует отдельных интерфейсов, определенных иначе чем интерфейсы с MAC и MLME.

10.2 *Общие примитивы управления*

Информация управления, специфичная для каждого уровня представлена как базовая информации управления (MIB) для этого уровня. Объекты управления уровня MAC и PHY рассматриваются как "содержащие" MIB для этого слоя. Общая модель MIB-связанных примитивов управления – обмен через SAP управления должна позволять пользовательскому объекту SAP получать (GET) или устанавливать (SET) значение признака MIB. Поступление примитива SET.request может требовать, чтобы объект уровня исполнил некоторые определенные действия.

Примитивы GET и SET фактически представлены, как и REQUEST со связанным CONFIRM примитивом. Этим примитивам предшествует MLME или PLME в зависимости от того, вовлечена ли SAP управления уровня MAC. Далее XX обозначает MLME или PLME:

XX-GET.request (MIBattribute)	Запрашивает значение данного MIBattribute.
XX-GET.confirm (status, MIB attribute, MIBattribute value)	Возвращает соответствующее значение признака MIB если статус = «success», иначе возвращает признак ошибки в поле Status. Возможные значения статуса ошибки включают «недействительный признак MIB» и «попытка получать только для записи MIB признак».
XX-SET.request (MIBattribute, MIBattributevalue)	Запрашивает, чтобы обозначенный признак MIB был установлен в данное значение. Если этот MIBaltribute подразумевает определенное действие, то этот запрос о том, что действие выполнено.
XX-SET.confirm (status, MIBattrib-	Если статус = «success», это подтверждает, что обозначенный при-

ute)	знак MIB был установлен в требуемое значение, иначе возвращает состояние ошибки в поле status. Если этот MIBattribute подразумевает определенное действие, то это подтверждает, что действие было выполнено. Возможные значения статуса ошибки включают «Недействительный признак MIB» и «попытка устанавливать только для чтения MIB признак».

Дополнительно, есть некоторые запросы (со связанными, confirms) которые могут быть переданы через данную SAP, и которые не вызывают получение или установку определенного признака MIB. Один из них поддержан каждой SAP, Следующим образом:

XX-RESET.request: где XX соответственно MLME или PLME.

XX-RESET.confirm

Эта служба используется инициализации объектов управления MIB и маршрутизаторов данных. Она может включать список признаков для инициализации для установки значений, отличных от значений по умолчанию. Связанный confirm указывает успех или неудачу запроса. Другие SAP-зависимые примитивы определены в 10.3.

10.3 *MLME SAP интерфейс*

В этом подпункте определены службы, предоставляемые MLME для SME. Эти службы описаны абстрактным способом и не подразумевают никакое специфическое выполнение или отдельный интерфейс. Примитивы MLME SAP имеют общую форму ACTION.request сопровождаемый ACTION.confirm. SME использует службы, предоставленные MLME через MLME SAP.

10.3.1 Управление питанием

Этот механизм поддерживает процесс установки и обслуживания способа управления питанием STA.

10.3.1.1 MLME-POWERMGT.request

Функция	Запрашивает изменения в режиме управления питанием.		
Параметры	PowerManagementMode	Enum {ACTIVE, POWER_SAVE}	Тип перечисление, который описывает желательный способ управления питанием STA.
	WakeUp	Boolean {True, false}	Когда истинно, MAC должен немедленно перейти в Активное состояние. Этот параметр не имеет никакого эффекта, если текущий способ управления питанием ACTIVE.
	ReceiveDTIMs	Boolean {True, false}	Когда истинно, этот параметр заставляет STA пробуждаться, чтобы получить все фреймы DTIM. Когда ложно, STA не требуется пробуждаться для каждого фрейма DTIM.
Назначение	Этот примитив производится SME для осуществления выполнения стратегии сбережения питания.		
Действия при получении	Этот запрос устанавливает параметры управления питанием STA. MLME впоследствии отправляет MLME-POWERMGT.confirm, который отражает результаты запроса изменения управления питанием.		

10.3.1.2 MLME-POWERMGT.confirm

Функция	Подтверждает изменения в режиме управления питанием.

Параметры	ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED}	Отражает результат MLME-POWERMGT.request.

Назначение Этот примитив производится MLME как результат MLME-POWERMGT.request для подтверждения нового режима управления питанием. Не вырабатывается, пока изменения не завершатся.

Действия при получении SME уведомляется об изменении режима управления питанием.

10.3.2 Сканирование

Этот механизм поддерживает процесс определения характеристик доступного BSS.

10.3.2.1 MLME-SCAN.request

Функция Этот примитив запрашивает список потенциальных BSS, к которым STA может позже пробовать присоединиться.

Параметры	BSSType	Enum {INFRASTRUCTURE, INDEPENDENT, ANY_BSS}	Определяет тип BSS: Инфраструктуры, Независимая или любая.
	BSSID	MACAddress (Любой правильный индивидуальный или broadcast MAC адрес)	Выделяет определенный или broadcast BSSID.
	SSID	Octet string (0-32 октета)	Определяет желательный SSID или broadcast SSID.
	ScanType	Enum {ACTIVE, PASSIVE}	Указывает активный или пассивный просмотр.
	ProbeDelay	Integer	Задержка (в мкс), перед сканированием.
	ChannelList	Определенный набор целых чисел (Каждый канал будет отобран из диапазона каналов для соответствующего PHY и набора транспорта)	Определяет список каналов, которые должны быть исследованы при сканировании на BSS.
	MinChannelTime	Integer (>= ProbeDelay)	Минимальное время (в TU), затрачиваемое на просмотр каждого канала.
	MaxChannelTime	Integer (>= MineChannelTime)	Максимальное время (в TU), затрачиваемое на просмотр каждого канала.

Назначение Этот примитив производится SME для STA, чтобы определить, есть ли другой BSSs, к которому он может присоединяться.

Действия при получении Этот запрос начинает процесс просмотра, когда закончена текущая последовательность обмена фреймами.

10.3.2.2 MLME-SCAN. confirm

Функция Этот примитив возвращает описания набора BSS, обнаруженных процессом просмотра.

	BSSDescription-Set	Набор BSSDescriptions (см. Таблица 54)	Возвращается BSSDescriptionSet для указания результатов запроса просмотра. Содержит ноль или большее количество BSSDescription.
	ResultCode	SUCCESS, INVALID PARAMETERS	Показывает результат MLME-SCAN.corifirm

Назначение Этот примитив производится MLME в результате MLME-SCAN.request, чтобы установить оперативное окружение STA.

Действия при получении SME уведомляется относительно результатов процедуры просмотра.

Таблица 54. Элементы каждого BSSDescription

Имя	Тип, диапазон	Описание
BSSID	MACAddress	BSSID найденного BSS.
SSID	Octet siring (1-32 октета)	SSID найденного BSS.
BSSType	Enum {INSFRASTRUCTURE, INDEPENDENT}	Тип найденного BSS.
Beacon Period	Integer	Период Маяка найденного BSS (в TU).
DTIM Period	Integer (Как определено в формате фрейма)	DTIM период BSS (в периодах маяка).
Timestamp	Integer	Timestamp полученного фрейма (ответ пробы/маяк) от найденного BSS.
Local Time	Integer	Значение TSF таймера STA в начале приема первого октета поля timestamp полученного фрейма (ответ пробы или маяк) от найденного BSS.
PHY parameter set	Как определено в формате фрейма	Набор параметров соответствующего PHY.
CF parameter set	Как определено в формате фрейма	Набор параметров в CF периодов, если найденный BSS поддерживает режим CF.
IBSS parameter set	Как определено в формате фрейма	Набор параметров для IBSS, если найденный BSS – IBSS.
Capability Information	Как определено в формате фрейма	Рекламируемые способности BSS.
BSSBasicRateSet	Набор целых от 1 до 127 включительно (для каждого целого в наборе)	Набор скоростей передачи данных который должен быть поддержан всеми STA, которые желают присоединиться к этому BSS. STA должны иметь возможность получать и передавать на каждой из скоростей передачи данных, внесенных в список.

10.3.3 Синхронизация

Этот механизм поддерживает процесс выбора равноправного в процессе аутентификации.

10.3.3.1 MLME-JOIN.request

Функция Этот примитив запрашивает синхронизацию с BSS.

Параметры	BSSDescription	BSSDescription	BSSDescription BSS, к которой нужно присоединиться. BSSDcacription - член набора описаний, который был возвращен в результате MLME-SCAN.request.
	JoinFailureTimeout	Integer (>=1)	Время в единицах интервала маяка, после, которого процедура присоединения будет закончена
	ProbeDelay	Integer	Задержка (в мкс), перед передачей фрейма пробы в течение активного сканирования.
	OperalionalRateSet	Набор целых от 1 до 127 включительно (для каждого целого в наборе)	Набор скоростей передачи данных который STA может использовать для связи в пределах BSS. STA должен иметь возможность получать данные на каждой из скоростей, внесенных в список. Это - супернабор базового набора скоростей BSS, рекламируемый BSS.
Назначение	Этот примитив произведен SME для STA, чтобы установить синхронизацию с BSS.		
Действия при получении	Этот примитив начинает процедуру синхронизации, как только текущая последовательность обмена фреймами выполнена. MLME синхронизирует свои времена с специфическими особенностями BSS, основываясь на элементах, полученных в параметре BSSDescription. MLME впоследствии отправляет MLME-JOIN.confirm, отражающий результаты.		

10.3.3.2 MLME-JOIN.confirm

Функция	Этот примитив подтверждает синхронизацию с BSS.		
Параметры	ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, TIMEOUT}	Показывает результат выполнения MLME- JOIN. Request.
Назначение	Этот примитив производится MLME в результате MLME-JOIN.request для установки синхронизации с BSS.		
Действия при получении	SME уведомляется относительно результатов процедуры синхронизации.		

10.3.4 Аутентификация

Этот механизм поддерживает процесс установления аутентификации с равноправным объектом MAC.

10.3.4.1 MLME-AUTHENTICATE.request

Функция	Этот примитив запроса аутентификации с равноправным объектом MAC.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC для просесса аутентификации.
	Authentication-Type	Enum {OPEN_SYSTEM, SHARED_KEY}	Определяет тип алгоритма аутентификации для использования в течение процесса аутентификации.
	Authentication-FailureTimeout	Integer (>=1)	Определяет срок (в TU) после, которого процедура аутентификации будет закончена

Назначение	Этот примитив производится SME для STA, чтобы установить аутентификацию с равноправным объектом MAC и разрешить обмен фреймами Класса 2 между двумя STA. В течение процедуры аутентификации, SME может производить дополнительные примитивы MLME-AUTHENTICATE.request.
Действия при получении	Этот примитив начинает процедуру аутентификации. MLME затем отправляет MLME-AUTHENTICATE.confirm, который отражает результаты.

10.3.4.2 MLME-AUTHENTICATE.confirm

Функция Этот примитив сообщает результаты аутентификации с равноправным объектом MAC.

Параметры

PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC, с которым проведился процесс аутентификации, указанный в STAAddress параметре переданного MLME-AUTHENTICATE.request.
Authentication-Type	Enum {OPEN_SYSTEM, SHARED_KEY}	Определяет тип алгоритма аутентификации, который использовался в течение процесса аутентификации. Это значение должно соответствовать параметру AuthenticationType, указанному в переданном MLME-AUTHENTICATE.request..
ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, TIMEOUT, TOO_MANY_SIMULTANEOUS_REQUEST_REFUSED}	Показывает результат выполнения MLME-AUTHENTICATE, request.

Назначение	Этот примитив произведится MLME, как результат MLME-AUTHENTICATE.request, для подтверждения аутентификации с равноправным объектом MAC.
Действия при получении	SME уведомляется о результатах процедуры аутентификации.

10.3.4.3 MLME-AUTHENTICATE.indication

Функция Этот примитив сообщает об установлении аутентификации с равноправным объектом MAC.

Параметры

PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC, с которым установилась аутентификация.
Authentication-Type	Enum {OPEN_SYSTEM, SHARED_KEY}	Определяет тип алгоритма аутентификации, который использовался в течение процесса аутентификации..

Назначение	Этот примитив производится MLME в результате установления аутентификации с определенным равноправным объектом MAC, последовавшим из процедуры аутентификации, которая была начата тем определенным равноправным объектом MAC.
Действия при получении	SME уведомляется относительно установления аутентификации.

10.3.5 Деаутентификация

Этот механизм поддерживает процесс разрыва аутентификации с равноправным объектом MAC.

10.3.5.1 MLME-DEAUTHENTICATE.request

Функция	Этот примитив запрашивает деаутентификацию с указанным равноправным объектом MAC.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC для выполнения процесса деаутентификации.
	ReasonCode	Как определено в формате фрейма	Определяет причину для начала процедуры деаутентификации.
Назначение	Этот примитив производится SME для STA, чтобы отменить аутентификацию с равноправным объектом MAC и запретить обмен фреймами Класса 2 между двумя STA. В течение процедуры деаутентификации, SME может производить дополнительные примитивы MLME-DEAUTHENTICATE.request.		
Действия при получении	Этот примитив начинает процедуру деаутентификации. MLME затем отправляет MLME-DEAUTHENTICATE.confirm, который отражает результаты.		

10.3.5.2 MLME-DEAUTHENTICATE.confirm

Функция	Этот примитив сообщает о результатах попытки деаутентификации с указанным равноправным объектом MAC.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC с которым был предпринят процесс деаутентификации.
	ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, TIMEOUT, TOO_MANY_SIMULTANEOUS_REQUEST_REFUSED}	Показывает результат выполнения MLME-DEAUTHENTICATE.request.
Назначение	Этот примитив производится MLME как результат MLME-DEAUTHENTICATE. request, для деаутентификации с указанным равноправным объектом MAC.		
Действия при получении	SME уведомляется относительно результатов процедуры деаутентификации.		

10.3.5.3 MLME-DEAUTHENTICATE.indication

Функция	Этот примитив сообщает о деаутентификации с указанным равноправным объектом MAC.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC с которым был произведен процесс деаутентификации.
	ReasonCode	Как определено в формате фрейма	Определяет причину, из-за которой была начата процедура деаутентификации.
Назначение	Этот примитив производится MLME в результате деаутентификации с указанным равноправным объектом MAC.		
Действия при получении	SME уведомляется относительно процедуры деаутентификации.		

10.3.6 Ассоциация

Следующие примитивы описывают, как STA становится связанным с точкой доступа (AP).

10.3.6.1 MLME-ASSOCIATE.request

Функция	Этот примитив запрашивает ассоциацию с указанным равноправным объектом MAC, который действует как AP.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC для выполнения процесса ассоциации.
	AssociateFailure-Timeuut	Integer (>=1)	Определяет время (в TU) после, которого процедура ассоциации будет закончена.
	Capabilitylnformation	Как определено в формате фрейма	Эксплуатационные определения способности, используемые объектом MAC
	ListenInterval	Integer (>=0)	Определяет число интервалов маяка, которые могут пройти перед тем, как STA проснется и будет слушать следующий маяк.
Назначение	Этот примитив производится SME когда STA желает установить ассоциацию с AP.		
Действия при получении	Этот примитив начинает процедуру ассоциации. MLME затем отправляет MLME-ASSOCIATE.confirm, который отражает результаты.		

10.3.6.2 MLME-ASSOCIATE.confirm

Функция	Этот примитив сообщает о результатах попытки ассоциации с указанным равноправным объектом MAC, который действует как AP.		
Параметры	ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED}	Показывает результат выполнения MLME-ASSOCIATE.request.
Назначение	Этот примитив производится MLME как результат MLME-ASSOCIATE .request, чтобы ассоциироваться с указанным равноправным объектом MAC, который действует как AP.		
Действия при получении	SME уведомляется относительно, результатов процедуры ассоциации.		

10.3.6.3 MLME-ASSOCIATE.indication

Функция	Этот примитив сообщает об установлении ассоциации с указанным равноправным объектом MAC.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC с которым произведена ассоциация.
Назначение	Этот примитив производится MLME в результате установления ассоциации с указанным равноправным объектом MAC, последовавшей из процедуры ассоциации, которая была начата тем указанным равноправным объектом MAC.		
Действия при получении	SME уведомляется относительно установления ассоциации.		

10.3.7 Реассоциация

Следующие примитивы описывают, как STA ассоциируется с другой AP.

10.3.7.1 MLME-REASSOCIATE.request

Функция Этот примитив запрашивает реассоциацию с новым указанным равноправным объектом MAC, который действует как AP.

Параметры

NewAPAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC для выполнения просесса реассоциации.
ReassociateFailureTimeuut	Integer (>=1)	Определяет время (в TU) после, которого процедура реассоциации будет закончена.
CapabilityInformation	Как определено в формате фрейма	Эксплуатационные определения способности, используемые объектом MAC
ListenInterval	Integer (>=0)	Определяет число интервалов маяка, которые могут пройти перед тем, как STA проснется и будет слушать следующий маяк.

Назначение Этот примитив производится SME для STA чтобы изменить ассоциацию на указанного нового равноправного объекта MAC, который действует как AP.

Действия при получении Этот примитив начинает процедуру реассоциации. MLME затем отправляет MLME-REASSOCIATE.confirm, который отражает результаты.

10.3.7.2 MLME-REASSOCIATE.confirm

Функция Этот примитив сообщает о результатах попытки реассоциации с указанным равноправным объектом MAC, который действует как AP.

Параметры

ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED}	Показывает результат выполнения MLME-REASSOCIATE.request.

Назначение Этот примитив производится MLME как результат MLME-REASSOCIATE.request, чтобы реассоциироваться с указанным равноправным объектом MAC, который действует как AP.

Действия при получении SME уведомляется относительно, результатов процедуры реассоциации.

10.3.7.3 MLME-REASSOCIATE.indication

Функция Этот примитив сообщает об установлении реассоциации с указанным равноправным объектом MAC.

Параметры

PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC с которым произведена реассоциация.

Назначение Этот примитив производится MLME в результате установления реассоциации с указанным равноправным объектом MAC, последовавшей из процедуры реассоциации, которая была начата тем указанным равноправным объектом MAC.

Действия при получении SME уведомляется относительно установления реассоциации.

10.3.8 Дисассоциация

Следующие примитивы описывают, как STA дисассоциируется с AP.

10.3.8.1 MLME-DISASSOCIATE.request

Функция	Этот примитив запрашивает дисассоциацию с указанным равноправным объектом MAC, который действует как AP.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC для выполнения процесса дисассоциации.
	ReasonCode	Как определено в формате фрейма	Указывает причину, по которой производится дисассоциация.
Назначение	Этот примитив производится SME для STA чтобы провести дисассоциацию с AP.		
Действия при получении	Этот примитив начинает процедуру дисассоциации. MLME затем отправляет MLME- DISASSOCIATE.confirm, который отражает результаты.		

10.3.8.2 MLME-DISASSOCIATE.confirm

Функция	Этот примитив сообщает о результатах попытки дисассоциации с указанным равноправным объектом MAC, который действует как AP.		
Параметры	ResultCode	Enum {SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED}	Показывает результат выполнения MLME-DISASSOCIATE.request.
Назначение	Этот примитив производится MLME как результат MLME-DISASSOCIATE.request, чтобы дисассоциироваться с указанным равноправным объектом MAC, который действует как AP.		
Действия при получении	SME уведомляется относительно, результатов процедуры дисассоциации.		

10.3.8.3 MLME-DISASSOCIATE.indication

Функция	Этот примитив сообщает о дисассоциации с указанным равноправным объектом MAC.		
Параметры	PeerSTAAddress	MACAddress (Любой правильный индивидуальный MAC адрес)	Определяет адрес равноправного объекта MAC с которым произведена реассоциация.
	ReasonCode	Как определено в формате фрейма	Указывает причину, по которой производится дисассоциация.
Назначение	Этот примитив производится MLME в результате дисассоциации с указанным равноправным объектом MAC, последовавшей из процедуры дисассоциации, которая была начата тем указанным равноправным объектом MAC.		
Действия при получении	SME уведомляется относительно установления дисассоциации.		

10.3.9 Сброс

Этот механизм поддерживает процесс сброса уровня MAC.

10.3.9.1 MLME-RESET.request

Функция	Этот примитив запрашивает сброс объекта MAC.		
Параметры	STAAddress	MACAddress (Любой правильный MAC адрес)	Определяет адрес MAC, который должен использоваться объектом MAC в процессе сброса. Это значение может использоваться, чтобы обеспечить локальный управляемый адрес STA.
	SetDefaultMIB	Boolean (True, false)	Если истина, все признаки MIB установлены в значения по умолчанию. Значения по умолчанию зависят от реализации. Иначе, MAC сбрасывается, но все признаки MIB сохраняют значения, которые были в установлены до получения примитива MLME-RESET.request.
Назначение	Этот примитив производится SME для сброса уровня MAC в начальное состояние. Примитив MLME-RESET.request должен использоваться до использования примитива MLME-START.request.		
Действия при получении	Этот примитив устанавливает MAC в исходное состояние, очищает все внутренние переменные в значения по умолчанию. Признаки MIB могут быть установлены в их значения по умолчанию, зависимые от реализации, установив флаг SetDefaultMIB в true. MLME впоследствии отправляет MLME-RESET.confirm, который отражает результаты.		

10.3.9.2 MLME-RESET.confirm

Функция	Этот примитив сообщает о сбросе объекта MAC.		
Параметры	ResultCode	Enum {SUCCESS}	Показывает результат выполнения MLME-RESET.request.
Назначение	Этот примитив производится MLME как результат MLME-RESET.request для сброса объекта MAC.		
Действия при получении	SME уведомляется относительно, результатов процедуры сброса.		

10.3.10 Старт

Этот механизм поддерживает создание новой BSS.

10.3.10.1 MLME-START.request

Функция	Этот примитив запрашивает, чтобы объект MAC запустить новую BSS.		
Параметры	SSID	Octet siring (1-32 октета)	SSID BSS.
	BSSType	Enum {INSFRASTRUCTURE, INDEPENDENT}	Тип BSS.
	Beacon Period	Integer (>=1)	Период Маяка BSS (в TU).
	DTIM Period	Как определено в 7.3.2.6	DTIM период BSS (в периодах маяка).
	CF parameter set	Как определено в 7.3.2.5	Набор параметров CF периодов, если BSS поддерживает режим CF. aCFPPeriod модифицируется, как побочный эффект получения примитива MLME-START.request.

PHY parameter set	Как определено в 7.3.2.3 или 7.3.2.4	Набор параметров соответствующего PHY.
IBSS parameter set	Как определено в 7.3.2.7	Набор параметров для IBSS, если BSS – IBSS.
ProbeDelay	Integer	Задержка (в мкс) перед передачей фрейма пробы в процесссе активного сканирования.
Capability Information	Как определено в 7.3.1.4	Рекламируемые способности BSS.
BSSBasicRateSet	Набор целых от 1 до 127 включительно (для каждого целого в наборе)	Набор скоростей передачи данных который должен быть поддержан всеми STA, которые желают присоединиться к этому BSS. STA, образующая BSS, должна иметь возможность получать и передавать на каждой из скоростей передачи данных, внесенных в список.
OperationalRateSet	Набор целых от 1 до 127 включительно (для каждого целого в наборе)	Набор скоростей передачи данных который STA может использовать для связи в пределах BSS. STA должна иметь возможность получать данные на каждой из скоростей, внесенных в список. Это - супернабор базового набора скоростей BSS, рекламируемый BSS.

Назначение Этот примитив производится SME, чтобы запустить BSS инфраструктуры (с объектом MAC, действующим как AP) или независимую BSS (с объектом MAC, действующим как первый STA в IBSS).
Примитив MLME-START.request, должен производиться после использования примитива MLME-RESET.request для сброса объекта MAC и до использовался примитива MLME-JOIN.request для успешного присоединиться к существующей инфраструктуре BSS или независимой BSS.
Примитив MLME-START.request не должен использоваться после успешного использования примитива MLME-START.request или MLME-JOIN.request без вмешательства примитива MLME-RESET.request.

Действия при получении Этот примитив начинает инициализацию процедуры BSS, как только выполнена текущая последовательность обмена фреймами. MLME впоследствии вырабатывает MLME-START.confirm, который отражает результаты процедуры создания.

10.3.10.2 MLME- START.confirm

Функция Этот примитив сообщает о результатах процедуры создания BSS.

Параметры		
ResultCode	Enum {SUCCESS, INVALID_PARAMRTERS, BSS_ALREADY_STARTED_OR_JOINED }	Показывает результат выполнения MLME- START.request.

Назначение Этот примитив производится MLME как результат MLME-START.request для создания новой BSS.

Действия при получении SME уведомляется относительно, результатов процедуры создания новой BSS.

11 Устройство менеджмента подуровня MAC

11.1 *Синхронизация*

Все STA в пределах отдельного BSS должны быть синхронизированы с общими тактами, используя механизмы, определенные ниже.

11.1.1 Основной подход

Функция синхронизации времени (TSF) поддерживает таймеры всех STA в одной и той же BSS синхронизированными. Все STA должны поддерживать локальный TSF таймер.

11.1.1.1 TSF для сетей инфраструктуры

В сети инфраструктуры AP должна быть задатчиком времени и должна выполнять TSF. AP должна инициализировать свой TSF таймер независимо от любых одновременно запущенных AP для того, чтобы минимизировать синхронизацию TSF таймеров нескольких AP. AP должна периодически передавать специальные фреймы называемые маяками, которые содержат копию ее TSF таймера для синхронизации других STA внутри BSS. Приемная STA всегда должна принимать информацию о времени из маяков, посланных от AP, обслуживающей ее BSS. Если STA TSF таймер отличается от временной метки принятого маяка, приемная STA должна установить свой локальный таймер в полученное значение временной метки.

AP должна генерировать маяки для передачи через каждый интервал времени BeaconPeriod.

11.1.1.2 TSF для независимого BSS (IBSS)

TSF в IBSS должен быть реализован в виде распределенного алгоритма, который должен выполняться всеми членами BSS. Каждая STA в BSS должна передавать маяки согласно алгоритму, описанному в этом пункте. Каждая STA в IBSS должна принять время, полученное от любого маяка или ответа пробы, которые содержат значение TSF более позднее, чем собственный TSF таймер.

11.1.2 Поддержание синхронизации

Каждая STA должна поддерживать TSF таймер по модулю 2^{64}, инкрементирующийся микросекундами. STA ожидают получения маяка на номинальной скорости. Интервал между маяками определен STA параметром aBeaconPeriod. STA, передающая маяк должна установить значение его временной метки так, чтобы оно равнялось значению TSF таймера STA в момент передачи первого бита временной метки к PHY плюс задержка распространения STA через его локальный PHY через MAC-PHY интерфейс на интерфейс с беспроводной средой (антенна, светодиодная излучающая поверхность, и т.д.). Алгоритмы в этом пункте определяют механизм, который поддерживает синхронизацию TSF таймеров в BSS в пределах 4 мкс плюс максимальная задержка распространения для PHY 1 Mb/s, или больше.

11.1.2.1 Генерация Маяка в сетях инфраструктуры

AP должна определять синхронизацию для всего BSS, передавая маяки в соответствии с атрибутом aBeaconPeriod в пределах AP. Это определяет серии TBTT поделенные точно на единицы времени aBeaconPeriod. Нулевое время определено, чтобы TBTT был с маяком, являющимся DTIM и передаваемым в начале CFP. В каждом TBTT, AP должна наметить маяк как следующий фрейм для передачи. Если определена среда с недоступным механизмом чувствительности к несущей (см. 9.2.1), AP должна задержать фактическую передачу маяка согласно основным средним правилам доступа, указанным в п. 9. Период маяка включен во фреймы «Маяк» и «Ответ Пробы» и STA должны принять тот период маяка при соединении BSS.

ПРИМЕЧАНИЕ: хотя передача маяка может быть отсрочена из-за отсрочки CSMA, последующие маяки должны быть намечены в номинальном интервале маяка, который показан на Рис. 32.

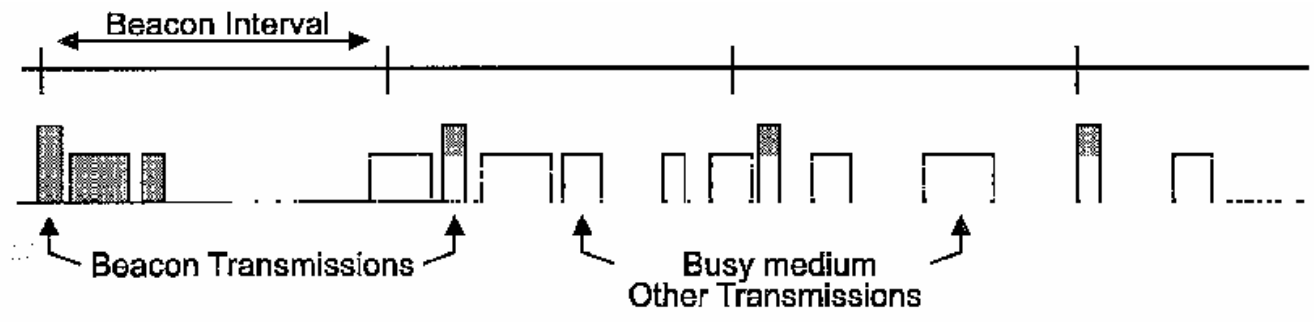


Рис. 32. Передача маяка в занятой сети.

11.1.2.2 Генерация Маяка в IBSS

Генерация Маяка в IBSS распределена. Период маяка включен во фреймы «Маяк» и «Ответ Пробы», и STA должны принимать периода маяка при соединении с IBSS. Все члены IBSS участвуют в генерации маяка. Каждая STA должна поддерживать свой TSF таймер, который используется для временной нарезки aBeaconPeriod. Интервал маяка в пределах IBSS устанавливается STA, обрабатывающей IBSS. Это определяет серии TBTT точно в единицах времени Периода Маяка. Нулевой отсчет Времени определяется, чтобы быть TBTT. В каждом TBTT STA должна:

a) Приостановить декрементирование таймера backoff для любой висящей индикации передачи (ATM), не являющейся маяковой или специальной
b) Вычислить случайную задержку, однородно распределенную в диапазоне между нолем и 2 x aCWmin x aSlotTime,
c) Ждать в течение периода случайной задержки, декрементируя таймер случайной задержки, используя тот же самый алгоритм, что и для backoff,
d) Если маяк прибывает прежде, чем случайный delayTimer истек, то остающаяся случайная задержка отменяется, висящая передача маяка отменяется, и ATIM backoff таймер должен снова декрементироваться.
e) Если случайная задержка истекла, и никакой маяк не прибыл в течение периода задержки, послать маяк.

(См. Рис. 33)

Передача маяка всегда должна происходить в течение Активного Периода STA, которые работают в режиме пониженного потребления. Это описано более подробно в 11.2.

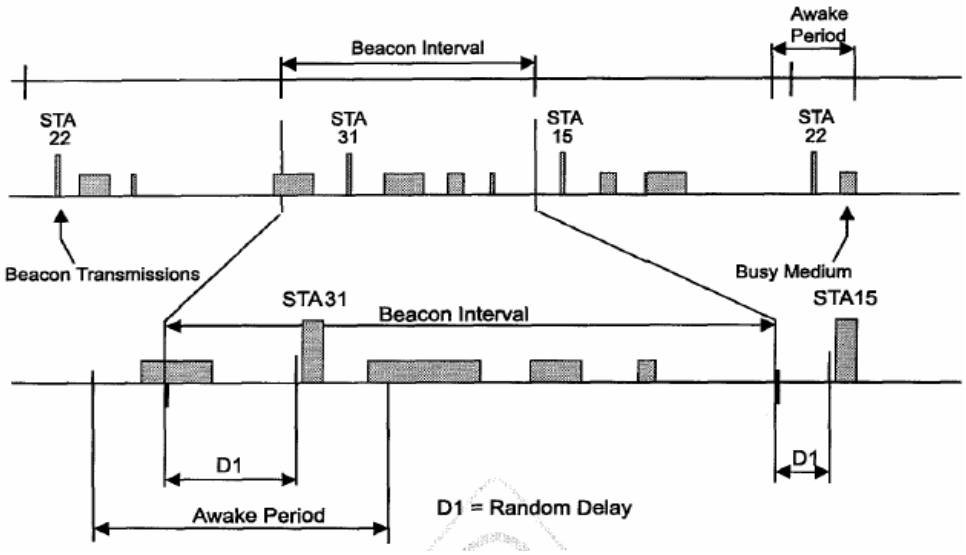


Рис. 33. Передача маяка в IBSS.

11.1.2.3 Прием маяка

STA должны использовать информацию из элемента CF Parameter Set всех принятых маяковых фреймов для обновления своих NAV, как определено в 9.3.2.2.

STA инфраструктурной сети должны использовать только оставшуюся информацию из принятых маяковых фреймов, если поле BSSID равно текущему адресу MAC, используемому STA, входящей в AP BSS.

STA в IBSS должны использовать только оставшуюся информацию из любого принятого маякового фрейма, для которого подполе IBSS поля Capability установлено в 1 и содержимое элемента SSID равно SSID IBSS. Использование этой информации определено в 11.1.4.

11.1.2.4 Точность таймера TSF

После приема маякового фрейма с действительным FCS и BSSID или SSID, как описано в 11.1.2.3, STA должна обновить свой TSF таймер в соответствии со следующим алгоритмом: значение принятой временной метки должно быть подстроено путем добавления числа, равного задержке приема STA через ее локальные PHY компоненты, плюс время, которое прошло с момента приема первого бита временной метки на интерфейсе MAC/PHY. В случае инфраструктурной BSS STA'шный TSF таймер затем должен быть подстроен до значения временной метки. В случае IBSS BSS STA'шный TSF таймер должен быть подстроен до значения временной метки в том случае, если ее значение является более поздним, чем собственное значение TSF таймера. Точность TSF таймера должна быть 0,01%.

11.1.3 Захват синхронизации, сканирование

STA должна работать либо в пассивном, либо в активном режиме сканирования, в зависимости от текущего значения параметра ScanMode из примитива MLME-SCAN.request.

После приема примитива MLME-SCAN.request STA должна выполнить сканирование. Параметр SSID указывает на SSID, который нужно искать. Для того, чтобы стать членом частной ESS, используя пассивное сканирование, STA должна сканировать маяковые фреймы, содержащие тот же самый ESS BSSID, возвращая все маяковые фреймы, совпадающие по нужному BSSID, в параметре BSSDescriptionSet соответствующего примитива MLME-SCAN.confirm вместе с соответствующими битами в поле Capabilities Information, указывающими, откуда пришел маяк – из инфраструктурной BSS или IBSS. Для активного сканирования STA должна передавать Пробные фреймы, содержащие нужный SSID. По завершении сканирования посылается примитив MLME-SCAN.confirm, содержащий всю принятую BSS информацию.

После приема примитива MLME-JOIN.request STA будет присоединяться к BSS, используя BSSID, значение таймера TSF, параметры PHY и период маяка, указанные в запросе.

После приема примитива MLME-SCAN.request с broadcast SSID STA должна пассивно сканировать любые маяковые фреймы либо активно передавать Пробные фреймы, содержащие broadcast SSID, в зависимости от соответствующего значения ScanMode. По завершении сканирования посылается примитив MLME-SCAN.confirm, содержащий всю принятую BSS информацию.

Если сканирование не привело к нахождению BSS с нужным SSID и нужным типом, либо любой BSS вообще, STA может начать IBSS после приема примитива MLME-START.request.

STA может начать свой собственный BSS без первого сканирования для присоединения к BSS.

Если STA начинает BSS, она должна установить для него BSSID. Если BSSType указывает на инфраструктурный BSS, STA должна начать инфраструктурный BSS, а BSSID должен быть равен aStationId. Значение BSSID должно оставаться неизменным, даже если значение aStationId изменяется после завершения MLME-START.request. Если BSSType указывает на IBSS, STA должна начать IBSS, а BSSID должен быть уникально локально администрирован IEEE адресом MAC, как определено в 5.2 IEEE Std 802-1990. Оставшиеся 46 бит этого адреса MAC должны быть выбраны таким образом, чтобы это число минимизировало вероятность генерации станциями STA того же числа, даже если эти станции имеют те же начальные условия. Значение параметра SSID должно использоваться как SSID новой BSS. Важно, чтобы разработчики понимали необходимость статической независимости среди случайных цифровых потоков от разных STA.

11.1.3.1 Пассивное сканирование

Если ScanType является пассивным, STA должна слушать каждый канал, сканируемый в течение времени, не большего, чем максимальная длительность, определяемая параметром ChannelTime.

11.1.3.2 Активное сканирование

Активное сканирование включает в себя генерацию Пробных фреймов и последующую обработку принятых фреймов Ответа Доступа. Подробности смотри ниже.

11.1.3.2.1 Передача ответа доступа

STA, принимающая фреймы Запроса Доступа, должна, в соответствии с критериями, описанными ниже, передавать ответ доступа только в том случае, если SSID в запросе доступа является broadcast SSID либо совпадает со специфическим SSID STA. Фреймы Ответа Доступа должны посылаться как прямые фреймы по адресу STA, которая сгенерировала запрос доступа. Ответ доступа должен посылаться с использованием нормальных правил передачи. AP должна отвечать на все запросы доступа, удовлетворяющие описанным выше критериям. В IBSS STA, которая отвечает на запрос доступа, должна быть той STA, которая сгенерировала последний маяк.

В каждой BSS должна быть по меньшей мере одна STA, которая «не спит» в любой данный момент времени, чтобы отвечать на запросы доступа. STA, которая послала маяк, должна оставаться в разбуженном состоянии и отвечать на запросы доступа до тех пор, пока не будет принят маяковый фрейм с текущим BSSID. Если STA является AP, она всегда должна оставаться в разбуженном состоянии и отвечать на запросы доступа. В IBSS может существовать более одной STA, которые отвечают на любой данный запрос доступа, в частности в тех случаях, когда более одной STA передают маяковый фрейм вслед за последним принятым TBTT, что может происходить из-за неуспешного приема предыдущего маяка либо из-за противоречий между передачами маяка.

11.1.3.2.2 Процедура активного сканирования

После приема примитива MLME-SCAN.request со ScanType, указывающим на активное сканирование, STA должна использовать следующую процедуру:

Для каждого сканируемого канала,

a) Ждать истечения времени ProbeDelay или принятия PHYRxstart.indication.
b) Выполнить процедуру Базового Доступа, как определено в 9.2.5.1.
c) Отослать запрос с broadcast SSID и broadcast BSSID.
d) Очистить и запустить ProbeTimer.
e) Если до того, как ProbeTimer достигнет значения MinChannelTime, не будет получен PHYCCA.indication (busy), очистить NAV и сканировать следующий канал; иначе, когда ProbeTimer достигнет значения MaxChannelTime, обработать все принятые ответы доступа.
f) Очистить NAV и сканировать следующий канал.

См. Рис. 34.

Когда все каналы из списка ChannelList будут просканированы, MLME должен передать MLME-Scan.confirm вместе с BSSDescriptionSet, содержащим всю информацию, полученную в процессе сканирования.

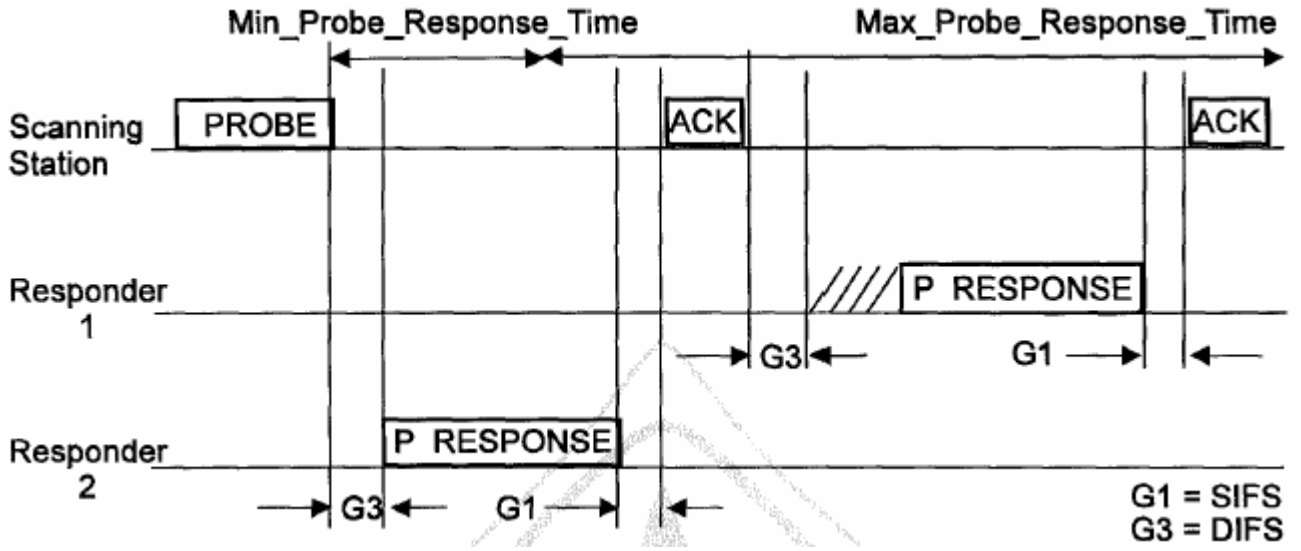


Рис. 34. Ответ доступа

11.1.3.3 Инициализация BSS

После приема примитива MLME-Start.request STA должна установить BSSID (как описано в 11.1.3), выбрать канальную информацию синхронизации, выбрать период маяка, инициализировать и запустить TSF таймер и начать передавать маяки.

11.1.3.4 Синхронизация с BSS

После приема примитива MLME-JOIN.request STA должна настроить BSSID, канальную информацию синхронизации и значение таймера TSF согласно параметрам запроса. После приема маякового фрейма от BSS MLME должен передать MLME-Join.confirm, индицируя об успешной работе. Если до приема маякового фрейма от BSS истекает JoinFailureTimeout, MLME должен передать MLME-Join.confirm, индицируя о неуспешной работе.

11.1.4 Подстройка таймеров STA

В инфраструктурной сети STA всегда должны принимать таймер из маяка или ответа доступа, приходящий от AP из их BSS.

В IBSS STA всегда должны принимать информацию, содержащуюся в маяковом фрейме или фрейме Ответа Доступа, если этот фрейм содержит совпадающий SSID и значение временной метки, более позднее, чем собственный TSF таймер. В ответ на MLME-Join.request STA должна инициализировать TSF таймер в 0 и не должна передавать маяк или ответ доступа до тех пор, пока она не услышит маяк или ответ доступа от члена IBSS с совпадающим SSID.

Во всех маяковых фреймах и фреймах Ответа Доступа содержится поле Timestamp. STA, принявшая такой фрейм от другой STA в IBSS с тем же самым SSID, должна сравнить поле Timestamp со своим собственным временем TSF. Если поле Timestamp принятого фрейма имеет более позднее значение, чем собственное время TSF, STA должна принять все параметры, содержащиеся в маяковом фрейме.

11.1.5 Синхронизация для PHY с ППРЧ (FH)

Каждая STA должна поддерживать таблицу всех прыгающих последовательностей, используемых в системе. Все STA в IBSS должны использовать одинаковую прыгающую последовательность. Каждый маяк и ответ доступа включает информацию о синхронизации канала, необходимую для определения шаблона прыжков и синхронизации с BSS.

STA должны использовать свои TSF таймера для отсчета aCurrentDwellTime. Это значение времени, в течение которого STA должны оставаться на каждой частоте своей прыгающей последовательности. Как только STA синхронизированы, они должны иметь одинаковое значение TSF таймера.

STA в IBSS должны передавать соответствующий PLME примитив на PHY, используемый для настройки на следующую частоту прыгающей последовательности всякий раз, когда

TSF timer MOD aCurrentDwellTime = 0

11.2 *Управление питанием*

11.2.1 Управление питанием в инфраструктурной сети

STA, изменяющая режим управления питанием, должна информировать об этом факте AP, используя биты Управления Питанием в поле Frame Control передаваемых фреймов. AP не должна произвольно передавать MSDU на STA, работающие в режиме пониженного потребления, а должна буферизировать MSDU и передавать их только в назначенное время.

STA, которые имеют текущие буферизированные MSDU внутри AP, идентифицируются в карте индикации трафика (TIM), которая должна быть включена отдельным элементом во все маяки, генерируемые AP. STA должна определять, что для нее есть буферизированные MSDU, путем приема и обработки TIM.

STA, работающие в режимах пониженного потребления (PS), должны периодически слушать маяки, как определено параметрами ListenInterval и ReceiveDTIMs примитива MLME-Pwr-Mgt.request.

В BSS, работающей с DCF, либо в течение периода соединения BSS с использованием PCF, при установлении того, что на AP есть буферизированные MSDU, STA, работающая в режиме PS, должна передать короткий фрейм PS-Poll на AP, которая должна немедленно ответить соответствующими буферизированными MSDU, либо подтверждением PS-Poll и ответом с буферизированными MSDU чуть позже. Если TIM указывает на то, что буферизированные MSDU посылаются в течение периода свободного соединения (CFP), CF-опрашиваемая STA не должна посылать фрейм PS-Poll, но должна оставаться активной до тех пор, пока она не примет буферизированные MSDU (или до окончания CFP). Если любая STA в своем BSS находится в режиме PS, AP должна буферизировать все broadcast и multicast MSDU и доставлять их всем STA сразу после следующего маякового фрейма, содержащего передачу о доставке TIM (DTIM).

STA должна оставаться в своем текущем режиме управления питанием до тех пор, пока она не информирует AP об изменении режима управления питанием с помощью успешного обмена фреймами. Режим управления питанием не должен изменяться в течение любой одиночной последовательности обмена фреймами, как описано в 9.7.

11.2.1.1 Режимы управления питанием STA

STA может находиться в одном из двух различных состояниях питания:

- Awake (разбуженное): питание полностью включено.
- Doze (сонное): STA не может осуществлять прием/передачу и потребляет очень мало энергии.

Способ перехода между этими двумя состояниями питания должен определяться режимом управления питания STA. Эти режимы приведены в Таблица 55.

Режим управления питанием STA выбирается с помощью параметра PowerManagementMode из MLME-POWERMGT.request. Как только STA обновит свой режим управления питанием, MLME должен передать MLME-POWERMGT.confirm для индикации об успешной работе.

Таблица 55. Режимы управления питанием.

Активный режим (AM)	STA может принимать фреймы в любой момент времени. В активном режиме STA должна оставаться в разбуженном состоянии. STA из опросного списка PCF должна находиться в активном режиме в течение длительности CFP.
Пониженное потребление (PS)	STA слушает выбранные маяки (в зависимости от своего aListenInterval) и посылает фреймы PS-Poll на AP в том случае, если элемент TIM из последнего принятого маяка указывает на наличие буферизированных MSDU для данной STA. AP должна передавать буферизированные MSDU на

	PS STA только в ответ на PS-Poll от данной STA, либо в течение CFP для CF-опрашиваемой PS STA. В режиме PS STA должна находиться в сонном состоянии и должна входить в разбуженное состояние для приема выбранных маяков, для приема broadcast и multicast передач, следующих за принятыми маяками, для передачи, и для ожидания ответов на переданные фреймы PS-Poll или (для CF-опрашиваемых STA) для приема буферизированных MSDU из передач свободного соединения.

Для изменения режима управления питанием STA должна информировать AP с помощью обмена фреймами по инициативе STA. Бит управления питанием в поле Frame Control посланного STA фрейма указывает режим управления питанием, который STA должна принять после успешного завершения обмена фреймами.

STA, которая переходит из сонного в разбуженное состояние для передачи, должна выполнять очистку назначения канала (CCA) до тех пор, пока она не обнаружит фреймовую последовательность, по которой она сможет правильно установить свой NAV, либо до истечения времени, равного ProbeDelay.

11.2.1.2 Передача AP TIM

TIM должна идентифицировать STA, для которых подвис и забуферизирован трафик на AP. Эта информация кодируется в частичную виртуальную битовую карту, как описано в 7.3.2.6. Кроме того, TIM содержит индикацию, есть ли подвисший broadcast/multicast трафик. В процессе ассоциации AP назначает каждой STA ассоциативный ID код (AID). AID 0 зарезервирован для индикации о наличии буферизированных broadcast/multicast MSDU. AP должна идентифицировать те STA, для которых она приготовила буферизированные MSDU, путем установки бит в частичной виртуальной битовой карте, которые соответствуют нужным SID.

11.2.1.3 Типы TIM

Различают два типа TIM: TIM и DTIM. После DTIM AP должна послать буферизированные broadcast/multicast MSDU, используя нормальные правила передачи фреймов, перед передачей любых unicast фреймов.

AP должна передавать TIM с каждым маяком. TIM типа “DTIM” передается внутри маяка в каждый DTIMPeriod, раньше, чем обычный TIM.

На Рис. 35 показаны действия AP и STA в предположении, что DTIM передается один раз на каждые три TIM. Верхняя линия на Рис. 35 представляет собой временную ось, показывающую интервал маяка вместе с интервалом DTIM, приходящимся на три интервала маяка. Вторая линия отображает действия AP. AP распределяет маяки для передачи в каждом интервале маяка, однако маяки могут быть задержаны в том случае, если есть трафик в TBTT. Это показано как “busy medium” на второй линии. Заметим, что вторая STA, имеющая ReceiveDTIMs, установленный в false, не включает свой приемник для всех DTIM.

Треть и четвертая линии на Рис. 35 отображают действия двух STA, работающих с различными требованиями к управлению питанием. Обе STA включают свои приемники всякий раз, когда им нужно послушать TIM. Это показано в виде линейно нарастающей мощности приемника до начала TBTT. Например, первая STA включает свой приемник и принимает TIM в первом маяке; этот TIM указывает на присутствие буферизированных MSDU для приемной STA. Приемная STA затем генерирует фрейм PS-Poll, который вызывает передачу буферизированных данных MSDU с AP. Broadcast и multicast MSDU посылаются AP вслед за передачей маяка, содержащего DTIM. DTIM индицируется полем счетчика DTIM из элемента TIM, имеющим значение 0.

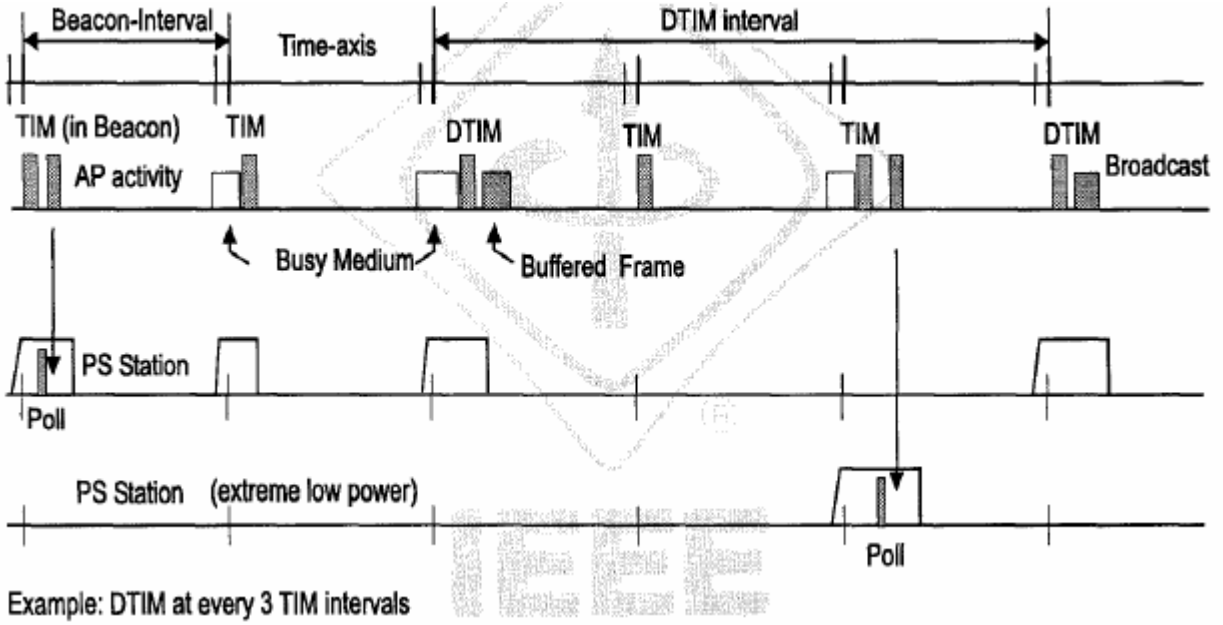


Рис. 35. Инфраструктурное управление питанием (без PCF)

11.2.1.4 Работа AP в течение периода соединения

AP должна хранить состояние управления питанием для каждой ассоциированной STA, которая указывает, в каком режиме управления питанием она работает в настоящее время. AP должна, в зависимости от режима управления питанием STA, временно буферизировать MSDU или фрейм управления, назначенный STA. Никакие MSDU или фреймы управления, принятые для STA, работающей в активном режиме, не должны буферизироваться по причинам управления питанием.

a) MSDU либо фреймы управления, назначенные PS STA, должны временно буферизироваться в AP. Алгоритм управления буферизацией находится за пределами данного стандарта.
b) MSDU либо фреймы управления, назначенные STA, работающей в активном режиме, должны передаваться непосредственно.
c) В каждом интервале маяка AP должна ассемблировать частичную виртуальную битовую карту, содержащую состояние буфера для всех STA в режиме PS, и отсылать ее в поле TIM маяка. Бит AID 0 должен быть установлен всякий раз, когда буферизируется broadcast или multicast трафик.
d) Все broadcast/multicast MSDU с очищенным битом Order в поле Frame Control должны быть буферизированы в том случае, если любая ассоциированная STA работает в режиме PS.
e) Сразу после каждого DTIM AP должна передать все буферизированные broadcast/multicast MSDU. Поле More Data каждого broadcast/multicast фрейма должно быть установлено для индикации наличия дополнительных broadcast/multicast MSDU. Если AP не может передать все broadcast/multicast MSDU до TBTT вслед за DTIM, AP должна указывать, что она продолжит доставку broadcast/multicast MSDU, установив broadcast/multicast бит в частичной виртуальной битовой карте элемента TIM каждого маякового фрейма, до тех пор, пока не будут переданы все буферизированные broadcast/multicast фреймы.
f) Одиночные буферизированные MSDU или фрейм управления для STA в режиме PS должны направляться на STA после того, как будет принят PS-Poll от этой STA. Поле More Data должно быть установлено для индикации наличия дополнительных буферизированных MSDU или фреймов управления для опрашиваемой STA. Дополнительные PS-Poll фреймы от той же STA должны подтверждаться и игнорироваться до тех пор, пока MSDU или фрейм управления также не будет успешно доставлен, либо считаться утраченными при превышении максимального количества попыток. Таким образом, повторный PS-Poll не будет считаться новым запросом для доставки буферизированного фрейма.

g) AP должна иметь функцию старения для стирания подвисшего трафика в том случае, если он буферизирован для истекшего периода времени.
h) Когда бы AP ни была информирована о том, что STA меняет свой режим на активный, AP должна послать буферизированные MSDU и фреймы управления (если они есть) на эту STA без ожидания PS-Poll.

11.2.1.5 Работа AP в течение CFP

AP должна хранить состояние управления питанием для каждой ассоциированной CF-опрашиваемой STA, которая указывает, в каком режиме управления питанием она работает в настоящее время. AP должна временно буферизировать MSDU, назначенные STA в PS режиме.

a) MSDU, назначенные PS STA, должны временно буферизироваться в AP. Алгоритм управления буферизацией находится за пределами данного стандарта.
b) MSDU, назначенные STA, работающей в активном режиме, должны передаваться как указано в разделе 9.
c) До начала CFP в каждом интервале маяка внутри CFP AP должна ассемблировать частичную виртуальную битовую карту, содержащую состояние буфера для всех STA в режиме PS, устанавливать биты точечного координатора (PC), предназначенного для опроса в течение данного CFP, и отсылать все это в поле TIM маяка. Бит AID 0 должен быть установлен всякий раз, когда буферизируется broadcast или multicast трафик.
d) Все broadcast/multicast MSDU с очищенным битом Order в поле Frame Control должны быть буферизированы в том случае, если любая ассоциированная STA работает в режиме PS независимо от того, является она CF-опрашиваемой, или нет.
e) Сразу после каждого DTIM (маяковый фрейм с полем DTIM элемента TIM равным нулю) AP должна передать все буферизированные broadcast/multicast фреймы. Поле More Data должно быть установлено для индикации наличия дополнительных буферизированных broadcast/multicast MSDU. Если AP не может передать все broadcast/multicast MSDU до TBTT вслед за DTIM, AP должна указывать, что она продолжит доставку broadcast/multicast MSDU, установив broadcast/multicast бит в частичной виртуальной битовой карте элемента TIM каждого маякового фрейма, до тех пор, пока не будут переданы все буферизированные broadcast/multicast фреймы.
f) Буферизированные MSDU или фреймы управления для STA в режиме PS должны направляться на CF-опрашиваемую STA под управлением PC. Передача таких буферизированных MSDU или фреймов управления должна начинаться сразу после передачи буферизированных broadcast или multicast фреймов (если таковые имеются) и должна происходить с наращиванием AID CF-опрашиваемых STA. CG-опрашиваемая STA, для которой элемент TIM последнего принятого маяка указывает наличие буферизированных MSDU или фреймов управления, должна находиться в разбуженном состоянии по крайней мере до тех пор, пока она не примет направленный от AP фрейм, поле Frame Control которого указывает на отсутствие дополнительных буферизированных MSDU или фреймов управления. После подтверждения последних буферизированных MSDU или фреймов управления CF-опрашиваемая STA, работающая в PS режиме, может войти в сонное состояния до следующего ожидаемого DTIM.
g) AP должна иметь функцию старения для стирания подвисшего трафика в том случае, если он буферизирован для истекшего периода времени. Точная спецификация для функции старения находится за пределами рассмотрения данного стандарта.
h) Когда бы AP ни была информирована о том, что CF-опрашиваемая STA меняет свой режим с PS на активный, AP должна поставить в очередь любые буферизированные фреймы, адресованные данной STA, для передачи их в соответствии с функцией PC (PCF).

11.2.1.6 Процедура приема для STA в режиме PS в течение периода соединения

STA в PS режиме должна работать так, как описано ниже для приема MSDU или фрейма управления от AP, когда не работает PC, и в течение периода соединения, когда PC работает.

a) STA должна просыпаться настолько рано, чтобы она могла принять следующий маяк по графику после ListenInterval от последнего TBTT.

b) Если STA обнаруживает, что бит, соответствующий ее AID, установлен в TIM, она должна передать PS-Poll для запроса MSDU или фрейма управления. Если в TIM установлено более одного бита, PS-Poll должен передаваться после случайной задержки, равномерно распределенной между нулем и CWMin.
c) STA должна оставаться в разбуженном состоянии до тех пор, пока она не примет ответ на этот poll, либо другой маяк, TIM которого указывает, что у AP нет каких-либо MSDU или фреймов управления, буферизированных для данной STA. Если бит, соответствующий AID STA, установлен в последующем TIM, STA должна передать еще один PS-Poll для запроса MSDU или фрейма(ов) управления.
d) Если поле More Data принятых MSDU или фреймов управления указывает, что для данной STA есть дополнительный буферизированный трафик, STA, по собственному усмотрению, должна осуществлять опрос (Poll) до тех пор, пока не останется никаких буферизированных MSDU или фреймов управления для данной STA.
e) Если ReceiveDTIMs равен true, STA должна просыпаться настолько рано, чтобы она могла принимать каждый DTIM. STA, принимающая broadcast/multicast MSDU, должна оставаться разбуженной до тех пор, пока поле More Data принятых broadcast/multicast MSDU не укажет на отсутствие дополнительных буферизированных broadcast/multicast MSDU, либо до тех пор, пока на это не укажет принятый TIM.

11.2.1.7 Процедура приема для STA в режиме PS в течение CFP

STA в PS режиме, которая ассоциирована как CF-опрашиваемая, должна работать в BSS с активным PC так, как описано ниже, для приема MSDU или фреймов управления от AP в течение CFP.

a) STA должна входить в разбуженное состояние для приема маякового фрейма (который содержит DTIM) в начале каждого CFP.
b) Чтобы принять broadcast/multicast MSDU, STA должна просыпаться настолько рано, чтобы она могла принять каждый DTIM, который может быть послан в течение CFP. STA, принимающая broadcast/multicast MSDU, должна оставаться разбуженной до тех пор, пока поле More Data принятых broadcast/multicast MSDU не укажет на отсутствие дополнительных буферизированных broadcast/multicast MSDU, либо до тех пор, пока на это не укажет принятый TIM.
c) Если STA обнаруживает, что бит, соответствующий ее AID, установлен в DTIM в начале CFP (либо в последующем TIM в течение CFP), она должна оставаться в разбуженном состоянии, по крайней мере, на такую часть времени CFP, чтобы принять MSDU или фрейм управления от AP с полем More Data поля Frame Control, указывающим на отсутствие дополнительного буферизированного трафика.
d) Если поле More Data поля Frame Control последних принятых MSDU или фрейма управления указывает, что для данной STA есть дополнительный буферизированный трафик, то, по окончании CFP, STA может остаться в разбуженном состоянии и передать PS-Poll фреймы в течение периода соединения для запроса доставки дополнительных MSDU или фреймов управления, либо может войти в сонное состояние в течение периода соединения (за исключением TBTT для DTIM, ожидаемых в период соединения), ожидая начала следующего CFP.

11.2.1.8 Работа STA в Активном режиме

У STA, работающей в этом режиме, приемник должен быть включен постоянно; она не должна интерпретировать информационную трафиковую часть маяка.

11.2.1.9 Функция старения AP

AP должна иметь функцию старению, предназначенную для стирания буферизированного трафика, который был буферизирован для истекшего периода времени. Эта функция должна основываться на значении aListenInterval STA, для которой буферизирован трафик. Функция старения AP не должна приводить к стиранию трафика до истечения любого периода времени, который короче, чем aListenInterval

STA, для которой буферизирован трафик. Точное определение функции старения лежит за пределами описания данного стандарта.

11.2.2 Управление питанием в IBSS

Данный раздел определяет механизм управления питанием для использования внутри IBSS.

11.2.2.1 Основной подход

Базовый подход аналогичен инфраструктурному случаю в том, что STA являются синхронизированными, а multicast MSDU и те MSDU, которые предназначены для передачи STA, работающим в режиме пониженного потребления, сначала анонсируются в периоды пробуждения этих STA. Анонсирование производится с помощью специального трафикового сообщения индикации (ATIM). STA в режиме PS должна слушать эти анонсы для определения того, нужно ли ей оставаться в разбуженном состоянии. Для передачи MSDU на STA, работающую в режиме PS, передающая STA сначала передает фрейм ATIM в течение окна ATIM, в котором все STA (включая работающие в режиме PS) находятся в разбуженном состоянии. Окно ATIM определяется как особый период времени с величиной ATIMWindow, следующий за TBTT, в течение которого должны передаваться только маяковые фреймы или фреймы ATIM. Время передачи ATIM является случайным, после передачи или приема маякового фрейма STA, что достигается процедурой backoff с окном соединения, равным aCWMinx. Направленные ATIM должны быть подтверждены. Если STA, передающая направленный ATIM, не получает подтверждения, она должна выполнить процедуру backoff для повторной передачи ATIM. Multicast ATIM не должны подтверждаться.

Если STA принимает направленный фрейм ATIM в течение окна ATIM, она должна подтвердить его и находиться в разбуженном состоянии в течение всего интервала маяка для приема анонсированных MSDU. Если STA не принимает ATIM, она может войти в сонное состояние в конце окна ATIM. Передачи MSDU, анонсированных посредством ATIM, рандомизированы после окна ATIM с использованием процедуры backoff, описанной в разделе 9.

Существует возможность того, что ATIM может быть принят от нескольких STA, и STA, которая принимает ATIM, может принимать несколько MSDU от передающей STA. Фреймы ATIM адресованы только той STA, для которой предназначены MSDU.

ATIM для broadcast или multicast MSDU должен иметь адрес назначения, идентичный тому, что и для MSDU.

После истечения интервала ATIM на STA, работающие в режиме PS, должны передаваться только те направленные MSDU или broadcast/multicast MSDU, которые были успешно анонсированы ATIM с подтверждением. Передача этих фреймов должна осуществляться с использованием нормальной процедуры доступа DCF.

На Рис. 36 показана основная последовательность работы в режиме пониженного потребления.

Расчетное состояние управления питанием другой STA может основываться на соответствующей информации, передаваемой данной STA, и на дополнительной информации, доступной локально, такой, как история неудачных попыток передачи. Использование RTS/CTS в IBSS может уменьшить количество передач для STA, работающей в режиме PS. Если RTS послан, а CTS не принят, то передающая STA может предположить, что соответствующая STA находится в режиме PS. Метод оценки состояния управления питанием другой STA в IBSS находится за пределами обзора данного стандарта.

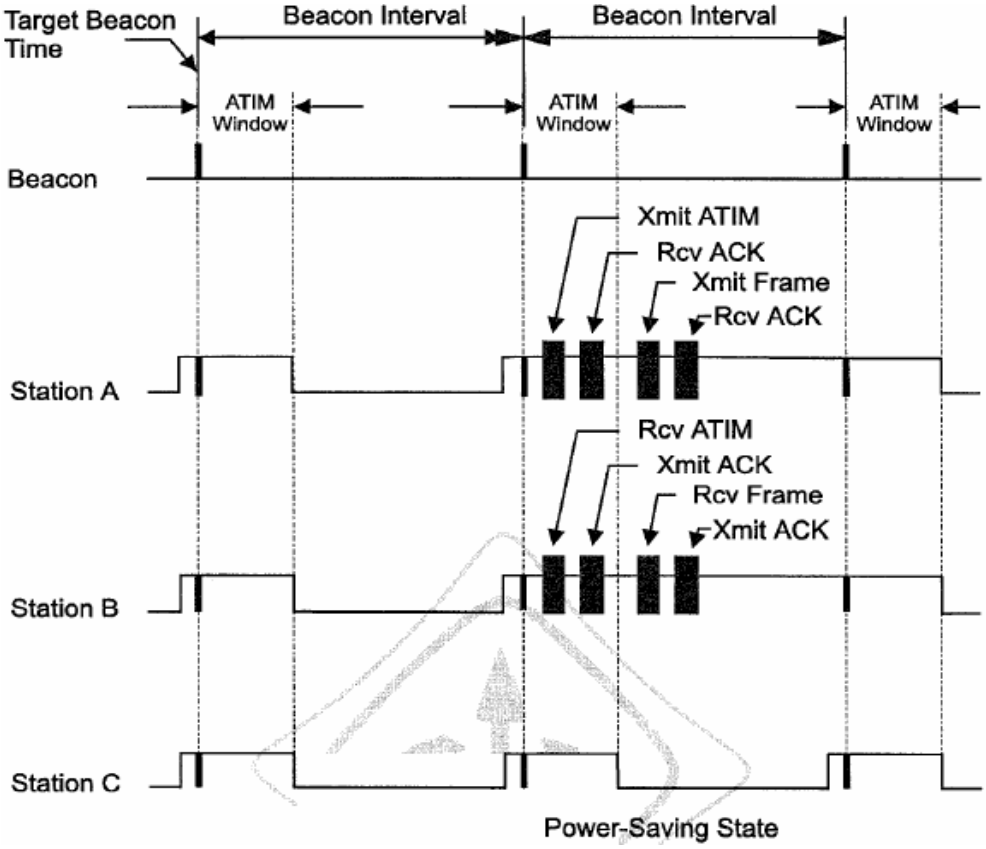


Рис. 36. Управление питанием в IBSS – основной случай.

11.2.2.2 Инициализация управления питанием в IBSS

Следующая процедура должна использоваться для инициализации управления питанием внутри новой IBSS, либо для получения информации о текущем режиме управления питанием внутри существующей IBSS.

a) STA, присоединяющаяся к существующей IBSS согласно процедуре 11.1.3.3, должна обновить свое окно ATIM значением, содержащимся в поле ATIM Window элемента IBSS Parameter Set внутри маяка или управляющего фрейма Ответа Доступа, принятого в течение процедуры сканирования.

b) STA, создающая новый IBSS согласно процедуре 11.1.3.3, должна установить значение поля ATIM Window элемента IBSS Parameter Set внутри передаваемых ею управляющих фреймов маяка в значение своего окна ATIM.

c) Началом окна ATIM должен быть TBTT, определенный в 11.1.2.2. конец окна ATIM определяется как

$$\text{TSF timer MOD BeaconInterval} = \text{ATIMWindow}$$

d) Период окна ATIM должен быть статическим в течение существования IBSS.

e) Нулевое значение окна ATIM должно указывать на то, что управление питанием внутри IBSS не используется.

11.2.2.3 Переходы STA по состояниям питания

STA может войти в режим PS тогда и только тогда, когда значение окна ATIM, используемого в IBSS, больше ноля. STA должна устанавливать подполе Power Management поля Frame Control в MSDU так, чтобы оно передавалось в соответствии с процедурой 7.1.3.1.7.

STA в режиме PS должна осуществлять переход между разбуженным и сонным состояниями в соответствии со следующими правилами:

a) Если STA работает в режиме PS, она должна входить в разбуженное состояние перед каждым TBTT.
b) Если STA принимает направленный фрейм управления ATIM, содержащий ее собственный адрес, либо multicast фрейм управления ATIM в течение окна ATIM, она должна оставаться в разбуженном состоянии до окончания следующего окна ATIM.
c) Если STA передает маяк или фрейм управления ATIM, она должна оставаться в разбуженном состоянии до окончания следующего окна ATIM независимо от того, получено ли подтверждение для ATIM.
d) Если STA не передает ATIM и не принимает направленный фрейм управления ATIM, содержащий ее собственный адрес, либо multicast фрейм управления ATIM в течение окна ATIM, она может вернуться в сонное состояние по окончании текущего окна ATIM.

11.2.2.4 Передача ATIM и фреймов

Если внутри IBSS используется управление питанием, все STA должны буферизировать MSDU, предназначенные для STA, работающих в режиме PS. Алгоритм, используемый для оценки состояния управления питанием внутри IBSS, находится за пределами обзора данного стандарта. MSDU для STA в активном режиме могут посылаться в любое действительное время.

a) Вслед за приемом или передачей маяка в течение окна ATIM STA должна передать направленный фрейм управления ATIM для каждой STA, для которой имеются буферизированные MSDU. Если у STA есть буферизированные multicast MSDU с очищенным битом Strictly Ordered, она должна передать соответствующий адресованный multicast фрейм ATIM. STA, передающая фрейм управления ATIM, должна оставаться разбуженной в течение всего текущего интервала маяка.
b) Все STA должны использовать процедуру backoff, описанную в 9.2.5.2, для передачи первого ATIM вслед за маяком. Все остальные ATIM должны передаваться с помощью традиционной процедуры доступа DCF.
c) Фреймы управления ATIM должны передаваться только в течение окна ATIM.
d) В течение окна ATIM STA не должна передавать никакие фреймы, кроме управляющих фреймов RTS, CTS, ACK, маяков и фреймов управления ATIM.
e) Направленные фреймы управления ATIM должны быть подтверждены. Если не принято подтверждение, ATIM должен быть передан повторно с помощью традиционной процедуры доступа DCF. Multicast фреймы управления ATIM не должны подтверждаться.
f) Если STA не может передать ATIM в течение окна ATIM, например, из-за соединения с другой STA, она должна сохранить буферизированные MSDU и попытаться передать ATIM в течение следующего окна ATIM.
g) Сразу после окна ATIM STA должна начать передачу буферизированных broadcast/multicast фреймов, для которых был предварительно передан ATIM. После передачи broadcast/multicast фреймов должны быть переданы MSDU и фреймы управления, адресованные тем STA, от которых было получено подтверждение на предварительно переданный фрейм ATIM. Все STA должны использовать процедуру backoff, описанную в 9.2.5.2, для передачи первого фрейма вслед за маяком. Все оставшиеся фреймы должны передаваться с помощью традиционной процедуры доступа DCF.
h) Буферизированные MSDU могут передаваться с помощью фрагментации. Если MSDU были частично переданы до посылки следующего маякового фрейма, STA должна сохранить буферизированные MSDU и анонсировать оставшиеся фрагменты путем передачи ATIM в течение следующего окна ATIM.
i) Если STA не может передать буферизированные MSDU в течение интервала маяка, в котором они были анонсированы, например, из-за соединения с другой STA, она должна сохранить буферизированные MSDU и анонсировать их снова путем передачи ATIM в течение следующего окна ATIM.
j) После передачи всех буферизированных MSDU STA может передать MSDU без анонсирования для тех STA, которые точно находятся в разбуженном состоянии на текущем интервале маяка, поскольку соответствующий фрейм управления ATIM или маяк уже передан.

k) STA может удалить фреймы, буферизированные для передачи на PS STA, если она определяет, что фрейм был буферизирован для истекшего момента времени либо по другим внутренним соображениям, зависящим от реализации STA (например, переполнение буфера). Фрейм не должен удаляться, если он был буферизирован для времени, меньшего aBeaconPeriod. Алгоритм управления такой буферизацией находится за пределами обзора данного стандарта.

11.3 *Ассоциация и реассоциация*

Данный раздел определяет то, как STA ассоциируется и реассоциируется с AP.

11.3.1 Процедура ассоциации STA

После приема MLME-ASSOCIATE.request STA должна ассоциироваться с AP, используя следующую процедуру:

a) STA должна передать запрос ассоциации на AP, с которой эта STA аутентифицирована.
b) Если фрейм ответа ассоциации принят со значением состояния “successful”, то STA является ассоциированной с AP, и MLME должен передать MLME-ASSOCIATE.confirm для индикации успешного завершения операции.
c) Если фрейм ответа ассоциации принят со значением состояния, отличным от “successful”, либо истекает AssociateFailureTimeout, то STA не является ассоциированной с AP, и MLME должен передать MLME-ASSOCIATE.confirm для индикации провала операции.

11.3.2 Процедура ассоциации AP

AP должна использовать следующую процедуру для того, чтобы поддерживать ассоциацию STA:

a) Когда бы ни был принят фрейм запроса ассоциации от STA и если STA является аутентифицированной, AP должна передать ответ ассоциации с кодом состояния, как определено в 7.3.1.9. Если значение состояния равно “successful”, в ответ должен быть включен Association ID, назначенный STA. Если STA не аутентифицирована, AP должна передать на нее фрейм деаутентификации.
b) После того, как ответ ассоциации со значением состояния “successful” будет подтвержден STA, эта STA считается ассоциированной с AP.
c) AP должна информировать систему распределения (DS) об ассоциации, и MLME должен передать MLME-ASSOCIATE.indication.

11.3.3 Процедура реассоциации STA

После приема MLME-REASSOCIATE.request STA должна реассоциироваться с AP, используя следующую процедуру:

a) STA должна передать фрейм запроса реассоциации на AP.
b) Если фрейм ответа реассоциации принят со значением состояния “successful”, то STA является ассоциированной с AP, и MLME должен передать MLME-REASSOCIATE.confirm для индикации успешного завершения операции.
c) Если фрейм ответа реассоциации принят со значением состояния, отличным от “successful”, либо истекает ReassociateFailureTimeout, то STA не является ассоциированной с AP, и MLME должен передать MLME-REASSOCIATE.confirm для индикации провала операции.

11.3.4 Процедура реассоциации AP

AP должна использовать следующую процедуру для того, чтобы поддерживать реассоциацию STA:

a) Когда бы ни был принят фрейм запроса реассоциации от STA и если STA является аутентифицированной, AP должна передать ответ реассоциации с кодом состояния, как определено в 7.3.1.9. Если значение состояния равно “successful”, в ответ должен быть включен Association ID, назначенный STA. Если STA не аутентифицирована, AP должна передать на нее фрейм деаутентификации.
b) После того, как ответ реассоциации со значением состояния “successful” будет подтвержден STA, эта STA считается ассоциированной с AP.

c) AP должна информировать DS о реассоциации, и MLME должен передать MLME-REASSOCIATE.indication.

11.4 *Определения информационной базы управления (MIB)*

12 Спецификация службы физического уровня (PHY)

12.1 *Обзор*

В данном разделе описаны службы PHY, обеспечиваемые беспроводным LAN MAC IEEE 802.11. Различные PHY определены как часть одного стандарта IEEE 802.11. Каждый PHY может состоять из следующих двух протокольных функций:

a) Функция конвергенции физического уровня, которая адаптирует возможности системы, зависящей от физической среды (PMD), к службе PHY. Данная функция поддерживается процедурой конвергенции физического уровня (PLCP), которая определяет метод маппирования протокольных данных подуровня MAC (MPDU) во фреймы, пригодные для посылки и приема пользовательских данных и управляющей информации между двумя или более STA, использующих соответствующую систему PMD.
b) Система PMD, функция которой определяет характеристики и методы приема/передачи данных через беспроводную среду (WM) между двумя или более STA.

Каждый подуровень PMD может потребовать определения уникального PLCP. Если подуровень PMD уже обеспечивает указанные службы PHY, то функция конвергенции физического уровня может быть нулевой.

12.2 *Функции PHY*

Базовая протокольная модель архитектуры IEEE 802.11 показана на Рис. 11. Большинство определений PHY содержат три функциональных устройства: функцию PMD, функцию конвергенции физического уровня и функцию менеджмента уровня.

Служба PHY предоставляется устройству MAC на STA через точку доступа обслуживания (SAP), называемую PHY-SAP, как показано на Рис. 11. Кроме того, можно определить набор примитивов для описания интерфейса между протокольным подуровнем конвергенции физического уровня и подуровнем PMD, называемым PMD-SAP.

12.3 *Детальные спецификации служб PHY*

12.3.1 Обзор

В данном разделе описаны службы PHY, обеспечиваемые для IEEE 802.11 MAC. Данные службы описаны абстрактно и не подразумевают какого-либо конкретного исполнения.

12.3.2 Обзор служб

Как показано на Рис. 11, функция PHY разделяется на два подуровня: подуровень PLCP и подуровень PMD. Функцией подуровня PLCP является обеспечение механизма передачи протокольных данных MAC (MPDU) между двумя или более STA через подуровень PMD.

12.3.3 Обзор сигналов взаимодействия

Примитивы, предназначенные для связи подуровней MAC и PHY, подразделяются на две категории:

a) Служебные примитивы, которые поддерживают взаимодействие равноправных подуровней MAC
b) Служебные примитивы, которые имеют локальное назначение и поддерживают взаимодействие подуровень-подуровень.

12.3.4 Базовые службы и опции

Все служебные примитивы, описанные здесь, являются обязательными, пока не указано иначе.

12.3.4.1 Служебные примитивы равноправных подуровней PHY-SAP

В Таблица 56 показаны примитивы, предназначенные для взаимодействия равноправных (peer-to-peer) подуровней.

Таблица 56. Служебные примитивы равноправных подуровней PHY-SAP

Тип примитива	Запрос (request)	Индикация (indicate)	Подтверждение (confirm)	Ответ (response)
PHY-DATA	X	X	X	

12.3.4.2 Служебные примитивы подуровень-подуровень PHY-SAP

В Таблица 57 показаны примитивы, предназначенные для взаимодействия подуровень-подуровень.

Таблица 57. Служебные примитивы подуровень-подуровень PHY-SAP

Тип примитива	Запрос (request)	Индикация (indicate)	Подтверждение (confirm)	Ответ (response)
PHY-TXSTART	X		X	
PHY-TXEND	X		X	
PHY-CCARESET	X		X	
PHY-CCA		X		
PHY-RXSTART		X		
PHY-RXEND		X		

12.3.4.3 Параметры служебных примитивов PHY-SAP

В Таблица 58 показаны параметры, используемые одним или более служебными примитивами PHY-SAP.

Таблица 58. Параметры служебных примитивов PHY-SAP

Параметр	Соответствующие примитивы	Значение
DATA	PHY-DATA.request PHY-DATA.indication	Байтовое значение 0x00 – 0xFF.
TXVECTOR	PHY-TXSTART.request	Набор параметров.
STATUS	PHY-CCA.indication	BUSY, IDLE
RXVECTOR	PHY-RXSTART.indication	Набор параметров.
RXERROR	PHY-RXEND.indication	NoError, FormatViolation, CarrierLost, UnsupportedRate

12.3.4.4 Описание векторов

Многие служебные примитивы используют параметрический вектор. Этот вектор является списком параметров, которые могут сильно зависеть от типа PHY. В Таблица 59 перечислены значения параметров, требуемые MAC или PHY в каждом из параметрических векторов.

Таблица 59. Описание векторов

Параметр	Соответствующий вектор	Значение
DATARATE	TXVECTOR, RXVECTOR	Зависит от PHY. Название поля, используемого для указания скорости приема и передачи данных, может отличаться для раз-

Параметр	Соответствующий вектор	Значение
		личных PHY.
LENGTH	TXVECTOR, RXVECTOR	Зависит от PHY.

12.3.5 Подробное описание служб PHY-SAP

В данном подразделе приведено описание служб, обеспечиваемых каждым примитивом подуровня PHY.

12.3.5.1 PHY-DATA.request

Данный примитив определяет передачу байта данных от подуровня MAC к локальному устройству PHY.
Семантика:

PHY-DATA.request (DATA)

Параметр DATA имеет байтовое значение от 0x00 до 0xFF.
Данный примитив генерируется подуровнем MAC для передачи байта данных на устройство PHY. Данный примитив может быть подан только после ответа об инициализации передатчика (PHY-TXSTART.confirm) от подуровня PHY.
Прием данного примитива устройством PHY обуславливает передачу байта данных конечным аппаратом PLCP. Когда устройство PHY получает байт, оно передает PHY-DATA.confirm на подуровень MAC.

12.3.5.2 PHY-DATA.indication

Данный примитив индицирует передачу данных от подуровня PHY на локальное устройство MAC.
Семантика:

PHY-DATA.indication (DATA)

Параметр DATA имеет байтовое значение от 0x00 до 0xFF.
PHY-DATA.indication генерируется приемным устройством PHY для передачи принятого байта данных на локальное устройство MAC. Время между приемом последнего бита поставляемого байта данных из беспроводной среды до приема данного примитива устройством MAC будет суммой aRXRFDelay + aRxPLCPDelay.

12.3.5.3 PHY-DATA.confirm

Данный примитив передается подуровнем PHY на локальное устройство MAC для подтверждения передачи данных от устройства MAC на подуровень PHY.
Семантика:

PHY-DATA.confirm

У данного примитива нет параметров.
Данный примитив будет передаваться подуровнем PHY на устройство MAC всякий раз, когда PLCP завершит передачу данных от устройства MAC на подуровень PHY. Подуровень PHY будет передавать этот примитив в ответ на каждый примитив PHY-DATA.request, переданный подуровнем MAC.
Прием данного примитива подуровнем MAC обусловит начало следующего запроса от устройства MAC.

12.3.5.4 PHY-TXSTART.request

Данный примитив является запросом подуровня MAC на локальное устройство PHY начать передачу MPDU.
Семантика:

PHY-TXSTART.request (TXVECTOR)

TXVECTOR представляет собой список параметров, которые подуровень MAC поставляет на локальное устройство PHY для того, чтобы передать MPDU. Этот вектор содержит параметры менеджмента как для PLCP, так и для PHY. Необходимые параметры PHY приведены в 12.3.4.4.
Данный примитив будет передаваться подуровнем MAC на устройство PHY всякий раз, когда подуровню MAC необходимо начать передачу MPDU.
После приема данного примитива устройство PHY запустит локальный конечный аппарат.

12.3.5.5 PHY-TXSTART.confirm

Данный примитив передается подуровнем PHY на локальное устройство MAC для подтверждения начала передачи. Подуровень PHY будет передавать этот примитив в ответ на каждый примитив PHY-TXSTART.request, переданный подуровнем MAC.

Семантика:

PHY-TXSTART.confirm

У данного примитива нет параметров.

Данный примитив будет передаваться подуровнем PHY на устройство MAC всякий раз, когда PHY примет PHY-TXSTART.request от устройства MAC и буде готов начать принимать байты данных.

Прием данного примитива устройством MAC обусловит начало передачи байт данных от MAC.

12.3.5.6 PHY-TXEND.request

Данный примитив является ответом подуровня MAC локальному устройству PHY, говорящим о том, что текущая передача MPDU завершена.

Семантика:

PHY-TXEND.request

У данного примитива нет параметров.

Данный примитив будет генерироваться всякий раз, когда подуровень MAC примет последний примитив PHY-DATA.confirm от локального устройства PHY для передаваемых в текущий момент MPDU.

После приема данного примитива локальное устройство PHY остановит конечный аппарат.

12.3.5.7 PHY-TXEND.confirm

Данный примитив передается подуровнем PHY на локальное устройство MAC для подтверждения окончания передачи. Подуровень PHY передает этот примитив в ответ на каждый примитив PHY-TXEND.request, переданный подуровнем MAC.

Семантика:

PHY-TXEND.confirm

У данного примитива нет параметров.

Данный примитив будет передаваться подуровнем PHY на устройство MAC всякий раз, когда PHY примет PHY-TXEND.request, сразу после окончания передачи последнего бита последнего байта данных, индицируя окончание передачи последнего байта данных.

Прием данного примитива устройство MAC обеспечит временную метку для протокола contention back-off.

12.3.5.8 PHY-CCARESET.request

Данный примитив является запросом подуровня MAC на локальное устройство PHY о сбросе явного назначения канала (CCA) для конечного аппарата.

Семантика:

PHY-CCARESET.request

У данного примитива нет параметров.

Данный примитив генерируется подуровнем MAC для локального устройства PHY по истечении таймера NAV. Данный запрос может использоваться некоторыми реализациями PHY для синхронизации антенного разноса со слотовыми временами.

После приема данного примитива устройство PHY сбросит таймера назначения PLCP CS/CCA в соответствующее состояние для окончания приема фрейма.

12.3.5.9 PHY-CCARESET.confirm

Данный примитив передается подуровнем PHY на локальное устройство MAC для подтверждения того, что PHY сбросил конечный аппарат CCA.

Семантика:

PHY-CCARESET.confirm

У данного примитива нет параметров.

Данный примитив передается подуровнем PHY на устройство MAC всякий раз, когда PHY примет PHY-CCARESET.request.

12.3.5.10 PHY-CCA.indication

Данный примитив является индикацией подуровня PHY для локального устройства MAC о текущем состоянии среды.

Семантика:

PHY-CCA.indication (STATE)

Параметр STATE может иметь одно из двух значений – BUSY или IDLE. Параметр имеет значение BUSY, если подуровень PHY определяет, что назначенный канал недоступен. В противном случае параметр имеет значение IDLE.

Данный примитив генерируется каждый раз, когда состояние канала меняется с busy на idle, и наоборот. При этом учитывается период времени, когда подуровень PHY принимает данные. Подуровень PHY поддерживает индикацию о занятости канала (busy) до тех пор, пока не истечет период времени, определяемый полем длины в действительном заголовке PLCP.

12.3.5.11 PHY-RXSTART.indication

Данный примитив является индикацией подуровня PHY локальному устройству MAC о том, что PLCP принял действительный разделитель начала фрейма (SFD) и заголовок PLCP.

Семантика:

PHY-RXSTART.indication (RXVECTOR)

RXVECTOR представляет собой список параметров, который подуровень PHY предоставляет локальному устройству MAC после приема действительного заголовка PLCP. Данный вектор может содержать как параметры MAC, так и параметры менеджмента MAC. Необходимые параметры приведены в 12.3.4.4.

Данный примитив генерируется локальным устройством PHY для подуровня MAC всякий раз, когда PHY успешно проверяет CRC заголовка PLCP в начале новых PLCP PDU.

12.3.5.12 PHY-RXEND.indication

Данный примитив является индикацией подуровня PHY локальному устройству MAC о том, что завершен прием текущих MPDU.

Семантика:

PHY-RXSTART.indication (RXERROR)

Параметр RXERROR может нести одно или более из следующих значений: NoError, FormatViolation, CarrierLost, UnsupportedRate. Ниже приведено описание параметров при условии возникновения различных ошибок.

- NoError. Данное значение используется для индикации об отсутствии ошибок в процессе приема на PLCP.
- FormatViolation. Данное значение используется для индикации того, что формат принятых PLCP PDU был ошибочным.
- CarrierLost. Данное значение используется для индикации того, что в течение приема MPDU была потеряна несущая, и никакая последующая обработка MPDU не может быть выполнена.
- UnsupportedRate. Данное значение используется для индикации того, что в течение приема PLCP PDU была обнаружена неподдерживаемая скорость данных.

Данный примитив генерируется подуровнем PHY для локального устройства MAC, чтобы сообщить о том, что приемный конечный аппарат завершил прием (с ошибками или без них).

13 Менеджмент PHY

В данном разделе описаны атрибуты и шаблоны менеджмента PHY. Сюда включены как PHY-зависимые, так и PHY-независимые части информационной базы управления (MIB) физическим уровнем. Не все атрибуты, перечисленные в данном разделе, поддерживаются каждым PHY. Каждый PHY содержит Управляемый Объектный (Managed Object) список, в котором определены специфические значения, необходимые для каждой реализации PHY. ASN.1 кодирование MIB приведено в Приложении D. При любом расхождении между определениями данного раздела и теми, что приведены в Приложении D, первые имеют преимущество.

13.1 *PHY MIB*

13.1.1 Атрибуты MIB

13.1.1.1 agPhyOperationGroup

aPHYType,
aRegDomainsSupported,
aCurrentRegDomain,
aSlotTime,
aCCATime,
aRxTxTurnaroundTime,
aTxPLCPDelay,
aRxTxSwitchTime,
aTxRampOnTime,
aTxRFDelay,
aSIFSTime,
aRxRFDelay,
aRxPLCPDelay,
aMACProcessingDelay,
aTxRampOffTime,
aPreambleLength,
aPLCPHeaderLength,
aMPDUDurationFactor,
aAirPropagationTime,
aTempType,
aCWmin,
aCWmax;

13.1.1.2 agPhyRateGroup

aSupportedDataRateTx,
aSupportedDataRateRx,
aMPDUMaxLength;

13.1.1.3 agPhyAntennaGroup

aCurrentTxAntenna,
aDiversitySupport;

13.1.1.4 agPhyTxPowerGroup

aNumberSupportedPowerLevels,
aTxPowerLevel1,
aTxPowerLevel2,

aTxPowerLevel3,
aTxPowerLevel4,
aTxPowerLevel5,
aTxPowerLevel6,
aTxPowerLevel7,
aTxPowerLevel8,
aCurrentTxPowerLevel;

13.1.1.5 agPhyFHSSGroup

aHopTime,
aCurrentChannelNumber,
aMaxDwellTime,
aCurrentSet,
aCurrentPattern,
aCurrentIndex;

13.1.1.6 agPhyDSSSGroup

aCurrentChannel,
aCCAModeSupported,
aCurrentCCAMode,
aEDThreshold;

13.1.1.7 agPhyIRGroup

aCCAWatchdogTimerMax,
aCCAWatchdogCountMax,
aCCAWatchdogTimerMin,
aCCAWatchdogCountMin;

13.1.1.8 agPhyStatusGroup

aSynthesizerLocked;

13.1.1.9 agPhyPowerSavingGroup

aCurrentPowerState,
aDozeTurnonTime;

13.1.1.10 agAntennaListGroup

aSupportedTxAntennas,
aSupportedRxAntennas,
aDiversitySelectionRx;

13.1.2 Объектный класс PHY

PHY MANAGED OBJECT CLASS
DERIVED FROM “ISO/IEC 10165-2”:top;
CHARACTERIZED BY
pPHYbase PACKAGE;
BEHAVIOUR
bPHYbase BEHAVIOUR
DEFINED AS “Объектный класс PHY обеспечивает поддержку всей рабочей PHY информации, которая может отличаться от PHY к PHY и от STA к STA, необходимую для связи с верхними уровнями.”

ATTRIBUTES	
aPHYType	GET,
aRegDomainsSupported	GET,
aCurrentRegDomain	GET-REPLACE,
aSlotTime	GET,
aCCATime	GET,
aRxTxTurnaroundTime	GET,
aTxPLCPDelay	GET,
aRxTxSwitchTime	GET,
aTxRampOnTime	GET,
aTxRFDelay	GET,
aSIFSTime	GET,
aRxRFDelay	GET,
aRxPLCPDelay	GET,
aMACProcessingDelay	GET,
aTxRampOffTime	GET,
aPreambleLength	GET,
aPLCPHeaderLength	GET,
aMPDUDurationFactor	GET,
aAirPropagationTime	GET,
aTempType	GET,
aCWmin	GET,
aCWmax	GET,
aSupportedDataRateTx	GET,
aSupportedDataRateRx	GET,
aMPDUMaxLength	GET,
aSupportedTxAntennas	GET,
aCurrentTxAntenna	GET-REPLACE,
aSupportedRxAntennas	GET,
aDiversitySupport	GET,
aDiversitySelectionRx	GET-REPLACE,
aNumberSupportedPowerLevels	GET,
aTxPowerLevel1	GET,
aTxPowerLevel2	GET,
aTxPowerLevel3	GET,
aTxPowerLevel4	GET,
aTxPowerLevel5	GET,
aTxPowerLevel6	GET,
aTxPowerLevel7	GET,
aTxPowerLevel8	GET,
aCurrentTxPowerLevel	GET-REPLACE,
aHopTime	GET,
aCurrentChannelNumber	GET-REPLACE,
aMaxDwellTime	GET,
aCurrentSet	GET-REPLACE,
aCurrentPattern	GET,
aCurrentIndex	GET-REPLACE,
aCurrentChannel	GET-REPLACE,
aCCAModeSupported	GET,
aCurrentCCAMode	GET-REPLACE,
aEDThreshold	GET-REPLACE,
aSynthesizerLocked	GET,
aCurrentPowerState	GET-REPLACE,

```
                aDozeTurnonTime                                 GET;

            ATTRIBUTE GROUPS
                agPhyOperationGroup,
                agPhyRateGroup,
                agPhyAntennaGroup,
                agPhyTxPowerGroup,
                agPhyFHSSGroup,
                agPhyDSSSGroup,
                agPhyIRGroup,
                agPhyStatusGroup,
                agPhyPowerSavingGroup,
                agAntennaListGroup;
            ACTIONS
                AcPHYreset;
            NOTIFICATIONS
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3)}
```

13.1.3 Шаблоны групп атрибутов PHY

13.1.3.1 agPhyOperationGroup

```
PhyOperationGroup ATTRIBUTE GROUP
        GROUP ELEMENTS
            aPHYType,
            aRegDomainsSupported,
            aCurrentRegDomain,
            aSlotTime,
            aCCATime,
            aRxTxTurnaroundTime,
            aTxPLCPDelay,
            aRxTxSwitchTime,
            aTxRampOnTime,
            aTxRFDelay,
            aSIFSTime,
            aRxRFDelay,
            aRxPLCPDelay,
            aMACProcessingDelay,
            aTxRampOffTime,
            aPreambleLength,
            aPLCPHeaderLength,
            aMPDUDurationFactor,
            aAirPropagationTime,
            aTempType,
            aCWmin,
            aCWmax;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
        PhyOperationGroup(0)}
```

13.1.3.2 agPhyRateGroup

```
PhyRateGroup ATTRIBUTE GROUP
        GROUP ELEMENTS
            aSupportedDataRateTx,
```

aSupportedDataRateRx,
aMPDUMaxLength;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8) PhyRateGroup (1)}

13.1.3.3 agPhyAntennaGroup

PhyAntennaGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aCurrentTxAntenna,
aDiversitySupport;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8) PhyAntennaGroup (2)}

13.1.3.4 agPhyTxPowerGroup

PhyTxPowerGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aNumberSupportedPowerLevels,
aTxPowerLevel1,
aTxPowerLevel2,
aTxPowerLevel3,
aTxPowerLevel4,
aTxPowerLevel5,
aTxPowerLevel6,
aTxPowerLevel7,
aTxPowerLevel8,
aCurrentTxPowerLevel;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8) PhyTxPowerGroup (3)}

13.1.3.5 agPhyFHSSGroup

PhyFHSSGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aHopTime,
aCurrentChannelNumber,
aMaxDwellTime,
aCurrentSet,
aCurrentPattern,
aCurrentIndex;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8) PhyFHSSGroup (4)}

13.1.3.6 agPhyDSSSGroup

PhyDSSSGroup ATTRIBUTE GROUP
GROUP ELEMENTS
aCurrentChannel,
aCCAModeSupported,
aCurrentCCAMode,
aEDThreshold;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8) PhyDSSSGroup (5)}

13.1.3.7 agPhyIRGroup

```
PhyIRGroup ATTRIBUTE GROUP
        GROUP ELEMENTS
                aCCAWatchdogTimerMax,
                aCCAWatchdogCountMax,
                aCCAWatchdogTimerMin,
                aCCAWatchdogCountMin;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyIRGroup (6)}
```

13.1.3.8 agPhyStatusGroup

```
PhyStatusGroup ATTRIBUTE GROUP
        GROUP ELEMENTS
                aSynthesizerLocked;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyStatusGroup (7)}
```

13.1.3.9 agPhyPowerSavingGroup

```
PhyPowerSavingGroup ATTRIBUTE GROUP
        GROUP ELEMENTS
                aCurrentPowerState,
                aDozeTurnonTime;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
PhyPowerSavingGroup (8)}
```

13.1.3.10 agAntennaListGroup

```
AntennaListGroup ATTRIBUTE GROUP
        GROUP ELEMENTS
                aSupportedTxAntennas,
                aSupportedRxAntennas,
                aDiversitySelectionRx;
REGISTERED AS { iso(1) member-body(2) us(840) ieee802dot11(10036) phy(3) attributeGroup(8)
AntennaListGroup (9)}
```

13.1.4 Шаблоны атрибутов PHY

Дальше меня не хватило.

14 FHSS PHY спецификации для индустриального, научного и медицинского диапазона (ISM) 2.4ГГц

14.1 *Обзор*

14.1.1 Обзор FHSS PHY

В данном разделе описаны службы PHY, обеспечиваемые беспроводным LAN MAC IEEE 802.11 для системы FHSS (по нашему ППРЧ). FHSS PHY состоит из следующих двух протокольных функций:

a) Функция конвергенции физического уровня, которая адаптирует возможности системы, зависящей от физической среды (PMD), к службе PHY. Данная функция поддерживается процедурой конвергенции физического уровня (PLCP), которая определяет метод маппирования протокольных данных подуровня MAC (MPDU) во фреймы, пригодные для посылки и приема пользовательских данных и управляющей информации между двумя или более STA, использующих соответствующую систему PMD.

b) Система PMD, функция которой определяет характеристики и методы приема/передачи данных через беспроводную среду (WM) между двумя или более STA.

14.1.2 Функции FHSS PHY

Архитектура FHSS PHY показана на Рис. 11. FHSS PHY содержит три функциональных устройства: функцию PMD, функцию конвергенции физического уровня и функцию менеджмента уровня. Служба FHSS PHY предоставляется устройству MAC на STA через точку доступа обслуживания (SAP), называемую PHY-SAP, как показано на Рис. 11. Кроме того, можно определить набор примитивов для описания интерфейса между протокольным подуровнем конвергенции физического уровня и подуровнем PMD, называемым PMD-SAP.

14.1.2.1 Подуровень PLCP

Для того, чтобы работа MAC в наименьшей степени зависела от подуровня PMD, определена функция конвергенции PHY. Данная функция упрощает обслуживание интерфейса PHY со службами MAC.

14.1.2.2 Устройство управления физическим уровнем (PLME)

PLME осуществляет управление локальными PHY функциями совместно с устройством менеджмента MAC.

14.1.2.3 Подуровень PMD

Подуровень PMD обслуживает передающий интерфейс, предназначенный для приема/передачи данных между одной или более STA.

14.1.3

15 DSSS PHY спецификации

16 IR PHY спецификации