

# Беспроводная сеть в Linux

Оригинал: "[Linux Wireless Networking](#)"

Автор: Петер Харрисон (Peter Harrison)

Перевод: Н.Ромоданов

Дата перевода: 10.09.2009 г.

## Введение

Моим самым первым веб сервером на Linux был древний десктоп, который я купил в магазине подержанных товаров и который был охарактеризован как "очень устаревший". Он был дешевым и работал, но был некрасивым и шумным, причем настолько шумным, что сразу стало невозможным с этим мириться. Потратив денег больше, чем было затрачено на этот антиквариат, я сделал его беспроводным и смог убрать его из спальни комнаты, куда в моей квартире был подключен вход соединения DSL. Если оглядываться назад, то я действительно занялся этим с тем, чтобы разобраться самому, а также из-за того, что мы все иногда поступаем неразумно. Я думал, что беспроводный Linux должен быть прост, но в то время это было не так. У меня с ним было столько головной боли, что я решил - одна из моих самых первых веб-страничек должна быть о моем маленьком кошмаре и должна предупреждать людей о том, как все это делать правильно. Именно так появился сайт [www.linuxhomenetworking.com](http://www.linuxhomenetworking.com). Эта глава о том, с чего все начиналось.

Беспроводные сети, использующие стандарт 802.11, имеют много преимуществ, причем не только эстетических, о которых я уже упоминал. Оборудование доступно, развертывать беспроводные сети относительно легко и недорого, а защита сетей постепенно улучшается. Однако прежде, чем мы рассмотрим, на что способен Linux сервер, убедитесь в том, что Вы купили сетевую плату, которая совместима с Linux. Если это вызывает у вас затруднения, не волнуйтесь, я все объясню ниже.

## Беспроводные сетевые платы, совместимые с Linux

Не все беспроводные сетевые платы работают с Linux. Поэтому выполните следующее: поищите списки устройств, совместимых с пакетом Wireless Tools. С помощью популярных поисковых систем сделать это сравнительно легко.

Производители беспроводных сетевых плат печально известны тем, что заменяют на своих платах наборы микросхем, добиваясь снижения стоимости комплектующих. Затем они поставляют другие драйвера с каждой новой платой с тем, чтобы сделать ее работоспособной. Можно купить платы с тем же самым номером модели одного и того же производителя и с очень разными схемами. Часто для новых плат отсутствуют драйвера для Linux. Всегда проверяйте списки совместимости прежде, чем купить беспроводное железо.

Беспроводная плата Linksys WMP11 является показательным примером таких проблем. В исходной версии платы, которая работала под Linux, использовался набор микросхем Intersil Prisms, но новая версия 2.7 (набор микросхем Broadcom) и версия 4 (набор микросхем InProComm) – не работают. Даже более – исходная плата WMP не работает без обновления прошивки.

В последние годы стало возможным использовать для сетевых плат под Linux обычные драйвера Windows. Это подробно обсуждается в разделе "Конфигурирование Linux с несовместимыми беспроводными сетевыми платами". Это метод требует понимания пакета Wireless Tools для Linux, который будет рассмотрен перед этим, но, прежде всего, для того, чтобы обеспечить некоторый фундамент, давайте рассмотрим основы беспроводных сетей.

Замечание: Не обманывайте себя. Тот факт, что система Linux смогла обнаружить вашу сетевую плату, не означает, что плата совместима. Всегда смотрите в Интернете списки совместимости с Linux с тем чтобы знать, как действовать дальше.

## Терминология, используемая в беспроводных сетях

Изучение плюсов и минусов беспроводных сетей для Linux будет проще, если мы будем говорить с вами на одном и том же языке. Прежде, чем продолжить, потратим время, чтобы ознакомиться с тремя ключевыми терминами беспроводных сетей: точкой беспроводного доступа, идентификатором сети Service Set ID и совместно используемым ключом шифрования. Разберемся с ними сейчас, поскольку на протяжении всей главы мы будем иметь с ними дело.

## Точки беспроводного доступа WAP

Беспроводная точка доступа (*прим. пер.:* по англ.: wireless access point – WAP) – это устройство, которое действует как центральный хаб, через который передаются все данные беспроводной сети. В наиболее часто используемом режиме функционирования (режим инфраструктуры - Infrastructure mode) все беспроводные сервера взаимодействуют друг с другом через точку WAP, которая для связи с Интернет обычно подключена к внешнему или интегрированному маршрутизатору. Следовательно, точки доступа WAP аналогичны свичам обычных проводных сетей.

Сервера могут взаимодействовать друг с другом и без точки доступа WAP, если их сетевые платы сконфигурированы в режим Ad Hoc, однако при этом они не смогут взаимодействовать с какими-либо другими устройствами. Для этого вам нужна точка доступа WAP в вашей сети.

## Идентификатор сети Service Set ID

Все беспроводные сети стандарта 802.11a/b, обычно используемые дома, работают в одном и том же частотном диапазоне, так что ваш компьютер имеет возможность слышать трафик, предназначенный для кого-нибудь из иной, соседней сети. Идентификатор сети (Extended Service Set ID - ESSID) помогает предотвратить возникновение мешанины из сообщений. Для каждой сети следует назначить идентификатор ESSID, который бы не совпадал с идентификаторами любых других соседних сетей, попадающих в зону действия нашей сети. Назначаем ESSID и устанавливаем его как на сетевых платах, так и на точках доступа WAP, что позволит игнорировать весь трафик, идущий от сетей с другими идентификаторами.

Большинство пакетов программ, работающих с беспроводными сетями, позволят вам просмотреть все идентификаторы ESSID сетей, находящихся в зоне доступа, и предоставляют возможность выбрать беспроводную сеть, к которой Вы желаете подключиться. К сожалению, это дает возможность легко прослушивать соседнюю сеть, и поэтому всякий раз, когда это возможно, лучше не только изменить заводское значение идентификатора ESSID, выбираемое по умолчанию, но также зашифровать весь свой беспроводный трафик.

Многие производители часто заменяют название ESSID на SSID (Service Set ID). Я буду использовать ESSID, если в приложении не будет указываться SSID. (*Прим. пер.:* Сети с ESSID – один из типов сетей с SSID. Это - сети с точкой доступа. Другой вид сети с SSID – сети с BSSID. Эти сети работают в режиме Ad Hoc).

## Шифрование

Шифрование является методом кодирования или скремблирования данных, с тем, чтобы просматривать исходные данные могли только лица, имеющие секретный ключ для разблокировки. Очевидно, что для того, чтобы взаимодействие было успешным, вам нужно на всех устройствах использовать одну и ту же схему шифрования.

## Схема WEP

Первой широко используемой схемой шифрования данных в домашних / офисных беспроводных сетях была схема Wired Equivalent Privacy (WEP). Однако в этой схеме шифрования была обнаружена ошибка, и вскоре стали свободно доступны пакеты типа "WEP crack" и утилита aircrack-ng, позволяющая расшифровывать ключи шифрования WEP в течение нескольких минут.

## Схема WPA

Более новая схема Wi-Fi Protected Access (WPA) свободна от уязвимостей, имеющих в схеме WEP. В ней используются следующие режимы:

- **Режим Pre Shared Key (PSK) или персональный режим**

На всех устройствах беспроводной сети используется вручную вводимый ключ шифрования.

- **Режим Enterprise**

Обычно используется как механизм аутентификации, так и схема шифрования со многими имеющимися в наличии возможностями. Одним из распространенных методов аутентификации является Extensible Authentication Protocol (протокол EAP). Протокол EAP основывается на использовании имени и пароля пользователя в LDAP или Active Directory, применяемых для

получения доступа к компьютеру. Это делается незаметно для пользователя. Как только пользователь регистрируется в своей системе, автоматически происходит обращение к EAP и предоставляется доступ к беспроводной сети. Для того, чтобы обеспечить дополнительную безопасность, протокол EAP часто объединяется с такими схемами шифрования, как TLS (Transport Layer Security – безопасность транспортного уровня, которая в настоящее время рассматривается в качестве преемника SSL) и TKIP (Temporal Key Integrity Protocol – схема быстрой генерации новых ключей шифрования).

**Замечание:** Прежде, чем активировать систему безопасности, обычно лучше проверить вашу сеть в незашифрованном состоянии. Это позволит вам ограничиться проблемами, связанными с основными настройками беспроводной сети, без дополнительных сложностей, связанных с шифрованием.

## Сети с Wireless-Tools для Linux

Пакет Wireless Tools для Linux, установленный по умолчанию, вероятно, наиболее соответствует требованиям стандарта 802.11a/b.

### Использование команды iwconfig для конфигурации пакета wireless-tools

После того, как ваша сетевая плата, совместимая с Linux, будет установлена физически, вам потребуется сконфигурировать IP платы и выполнить настройки беспроводной сети прежде, чем можно будет использовать пакет Wireless Tools.

Вы можете сконфигурировать IP вашей сетевой платы также, как если бы это было обычное Ethernet устройство. После выполнения команды ifup сетевая плата станет активной, но не будет работать должным образом, поскольку еще не были сконфигурированы настройки беспроводной сети.

Наиболее часто используемая команда в пакете Wireless Tools – iwconfig, которую Вы можете использовать для конфигурирования большинства параметров беспроводной сети, в том числе задания SSID и выбора режима работы. Выбор режима работы Managed означает, что в сети имеется беспроводная точка доступа WAP, а Ad-hoc указывает, что ее нет.

Например, если беспроводная сетевая плата имеет имя eth0, а ESSID вашей сети – homenet, то команды будут следующими:

```
iwconfig eth0 mode Managed  
iwconfig eth0 essid homenet
```

Теперь ваша сетевая плата должна стать полностью функциональной. Вам нужно будет запускать эти команды iwconfig всякий раз, когда Вы используете команду ifup. Однако, возникает проблема – не забывать это делать. В следующем разделе будет показано, как сделать изменения, внесенные командой iwconfig, постоянными.

### Сохранение конфигурации wireless-tools

После того, как Вы проверили свою конфигурацию ad-hoc, вам потребуется сохранить сделанные изменения. Способы сохранения изменений слегка варьируются в зависимости от используемого дистрибутива.

#### Fedora / RedHat

Конфигурация беспроводной сети в Fedora / RedHat потребует внесения нескольких дополнительных строк в конфигурационные файлы вашей сетевой платы.

1. Сконфигурируйте файл /etc/sysconfig/network-scripts/ifcfg-eth0 обычным образом, как если

это была обычная сетевая плата Ethernet.

DHCP Version	Fixed IP Version
=====	=====
DEVICE=eth0	DEVICE=eth0
USERCTL=yes	IPADDR=192.168.1.100
ONBOOT=yes	NETMASK=255.255.255.0
BOOTPROTO=dhcp	ONBOOT=yes
	BOOTPROTO=static

*Прим. пер.: DHCP Version - вариант для динамически распределяемых ip адресов; Fixed IP Version – вариант с фиксированными ip адресами.*

2. Для того, чтобы указать, что сетевая плата беспроводная, в конце добавьте приведенные ниже инструкции: укажите используемый ESSID (в нашем случае homenet) и в качестве режима работы выберите Managed (точка доступа WAP присутствует на сети) или Ad-hoc (точки доступа нет). "Managed" является наиболее вероятным вариантом в случае, если в вашей сети имеется беспроводный маршрутизатор или точка доступа.

Если Вы используете беспроводный маршрутизатор и сетевую плату стандарта 802.11g, то можно указать максимальную скорость, равную 54Мбит/сек – максимальной скорости передачи данных, обеспечиваемую этим протоколом. Если нет, то по умолчанию будет использоваться скорость, равная 11 Мбит/сек – максимальная скорость более медленных протоколов. Сетевая плата автоматически использует тип протокола с точкой доступа WAP. Вам нужно просто указать максимальную скорость.

```
#
# Wireless configuration
#
TYPE=Wireless
MODE=Managed
ESSID=homenet
RATE=54Mb/s
```

Эти команды нужны только в главном интерфейсном файле. Они не нужны для IP алиасов. При использовании команд ifup и ifdown ваша беспроводная сетевая плата должна функционировать точно также, как и обычная сетевая плата Ethernet.

## Debian / Ubuntu

Конфигурация в системах Debian / Ubuntu требует добавления в файл /etc/network/interfaces правильного параметра wireless-ssid.

```
#
# File: /etc/network/interfaces
#
# The primary network interface
auto eth1
iface eth1 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    wireless-ssid homenet
```

```
auto eth0
iface eth0 inet dhcp
    wireless-essid jamrock
```

В этом примере интерфейс eth1 использует для идентификатора ESSID значение homenet, тогда как интерфейс eth0 использует для идентификатора ESSID значение jamrock.

### Конфигурирование схемы шифрования WEP

В Linux поддерживаются обе схемы шифрования WEP и WPA. Ниже рассказывается, как Вы можете сконфигурировать их на своей системе.

### Генерация ключа WEP

Для шифрования WEP требуется ключ шифрования, который Вы можете выбрать самостоятельно или можете сгенерировать случайным образом с помощью команды dd так, как это показано ниже:

```
[root@bigboy tmp]# dd if=/dev/random bs=1 count=5 2>/dev/null | xxd -ps
c276246d65
[root@bigboy tmp]#
```

По умолчанию WEP для Linux использует 40 битовый ключ, форматированный в шестнадцатеричной нотации, т.е. посредством цифр от 0 до 9 и букв от A до F. Следовательно, в приведенном выше примере требуется задать количество байтов равным 5, и в результате будет сгенерировано в два раза больше (десять) шестнадцатеричных символов. В таблице 13.1 показано число байтов, требуемое для генерации ключей различной длины, и соответствующее число шестнадцатеричных символов, которые должны быть в ключе.

Таблица 13-1 Соотношение числа байтов и длин ключа WEP

Длина ключа (биты)	Количество байт	Количество шестнадцатеричных символов
40	5	10
64	8	16
104	13	26
128	16	32
152	19	28
232	29	58
256	32	64

Если Вы решили сделать свой собственный ключ, то используйте правильное количество шестнадцатеричных цифр.

### Конфигурирование ключа WEP для Fedora / RedHat

С помощью команды iwconfig ваш ключ WEP может быть временно добавлен в

конфигурацию вашей сетевой платы. Убедитесь в том, что среди символов ключа нет запятых или иных шестнадцатеричных символов. Всего должно быть десять символов:

```
iwconfig eth0 key 967136deac
```

Те же самые правила (отсутствие запятых или других шестнадцатеричных символов среди десяти символов ключа) применяются, когда для шифрования используется файл /etc/sysconfig/network-scripts :

```
#
# File: ifcfg-eth0
#
DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=static
TYPE=Wireless
MODE=Managed
ESSID=homenet
KEY=967136deac
```

Замечание: Использование файлов ключей в директории /etc/sysconfig/network-scripts поддерживается только в новых версиях Fedora . Формат файла тот же самый, как и в старых вариантах конфигурационных файлов интерфейса. Помните, в конфигурационном файле интерфейса не поддерживается использование инструкции KEY.

```
#
# File: /etc/sysconfig/network-scripts/keys-eth0
#
KEY=967136deac
```

### Конфигурирование ключа WEP для Debian / Ubuntu

Конфигурация в системах Debian / Ubuntu требует добавления в файл /etc/network/interfaces правильного параметра wireless-ssid.

```
#
# File: /etc/network/interfaces
#
# The primary network interface
auto eth1
iface eth1 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    wireless-key 967136deac

    wireless-ssid homenet
```

В этом примере указывается, что должны использоваться значение нашего ключа WEP, равное 967136deac, и ESSID, равный homenet, и они будут использоваться сразу, как только будет активирован беспроводный интерфейс eth1.

## Шифрование WPA

В Linux шифрование WPA основывается на демоне- суппликанте, который от имени операционной системы запрашивает проверку подлинности доступа, а также выполняет шифрование данных. Он запускается независимо от демона сети и, поэтому, для шифрования WPA сетевые интерфейсы вообще не конфигурируются.

*Прим.перев.: Слово "supplicant" переводится как "проситель". Подходящего компьютерного термина для перевода я не нашел, поэтому оставляю название в виде транслитерации английского термина. Домашняя страница проекта WPA Supplicant - [http://hostap.epitest.fi/wpa\\_supplicant/](http://hostap.epitest.fi/wpa_supplicant/)*

### Установка суппликанта шифрования WPA

Установка проста. Установите пакет RPM `wpa_supplicant` или пакет DEB `wpa_supplicant`. Если вам нужно вспомнить, как это делается, смотрите Главу 6 "[Установка программного обеспечения Linux](#)", где это подробно описано.

### Файл `wpa_supplicant.conf`

Основным конфигурационным файлом суппликанта WPA является файл `/etc/wpa_supplicant/wpa_supplicant.conf`. Его конфигурирование хорошо задокументировано, с примерами, в страницах `man`.

```
[root@bigboy tmp]# man wpa_supplicant.conf
```

Замечание: В Debian / Ubuntu файл может не создаваться во время установки и вам следует его создать вручную следующим образом:

```
root@u-server:/tmp# mkdir -p /etc/wpa_supplicant
root@u-server:/tmp# vi /etc/wpa_supplicant/wpa_supplicant.conf
```

В этой главе мы сфокусируемся только на простом методе шифрования PSK WPA, другие методы выходят за рамки рассмотрения настоящей книги.

В этом примере мы установим идентификатор SSID равным `homenet` и будем использовать WPA-PSK шифрование с ключом шифрования `"ketchup_and_mustard"`.

```
#
# File: wpa_supplicant.conf
#
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=root
network={
    ssid="homenet"
    key_mgmt=WPA-PSK
    psk="ketchup_and_mustard"
}
```

Если Вы обеспокоены, что люди могут прочитать ваш файл `wpa_supplicant.conf`, то зашифруйте ключ PSK с помощью команды `wpa_passphrase`, которая сгенерирует пример конфигурации. Она потребует в качестве аргументов идентификатор SSID и незашифрованный ключ. В этом примере мы видим, что незашифрованную строку

psk="ketchup\_and\_mustard" можно заменить зашифрованным эквивалентом, в котором не используются кавычки.

```
[root@bigboy tmp]# wpa_passphrase homenet ketchup_and_mustard
network={
    ssid="homenet"
    #psk="ketchup_and_mustard"
    psk=aeaa365d1703f88afc11715cd997b71038ce5798907510bd1b1c6786d33c8c3a
}
[root@bigboy tmp]#
```

Замечание: Единственное место, где должен быть определен ключ шифрования, это конфигурационный файл WPA.

Дальнейшие шаги по шифрованию WPA - Fedora / RedHat

Суппликант WPA также использует файл /etc/sysconfig/wpa\_supplicant, в котором определяется, какие интерфейсы должны контролироваться и какие для этого драйвера должны использоваться.

В этом примере WPA следует применять к интерфейсу eth0 и по умолчанию использовать драйвер "wext".

```
#
# File: /etc/sysconfig/wpa_supplicant
#
INTERFACES="-ieth0"
DRIVERS="-Dwext"
```

Здесь мы видим, что WPA сконфигурирован для wlan0, создаваемого с помощью драйвера ndiswrapper.

```
#
# File: /etc/sysconfig/wpa_supplicant
#
INTERFACES="-iwlan0"
DRIVERS="-Dndiswrapper"
```

Дальнейшую подсказку, касающуюся файла wpa\_supplicant, можно получить из страниц man.

```
[root@bigboy tmp]# man wpa_supplicant
```

После того, как Вы завершили редактирование конфигурационных файлов, нужно сразу стартовать демон суппликанта WPA с тем, чтобы настройки стали активными. Не забудьте с помощью команды chkconfig сделать эту активацию постоянной.

```
[root@bigboy tmp]# service wpa_supplicant restart
[root@bigboy tmp]# chkconfig wpa_supplicant on
```

Наконец сконфигурируйте свою сетевую плату также, как и для беспроводной сети, но без идентификатора SSID и ключа шифрования, поскольку эта информация будет подаваться через суппликант WPA.

```
File: /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Wireless
MODE=Managed
```

При решении любых проблем, с которыми Вы можете встретиться, пожалуйста, обращайтесь в раздел по устранению проблем, имеющийся в настоящей главе.

### Дальнейшие шаги по шифрованию WPA - Debian / Ubuntu

Суппликант WPA можно вызвать из командной строки. В системах Debian / Ubuntu нужно модифицировать файл `/etc/network/interfaces` и включить в него параметр `pre-up`, за которым будет следовать допустимый набор команд суппликанта WPA. В данном примере с помощью опции `"-c"` дается ссылка на файл `/etc/wpa_supplicant/wpa_supplicant.conf`, а нужный интерфейс определяется с помощью опции `"-i"`. Затем используется параметр `post-down`, в котором определяются команды, останавливающие работу демона суппликанта WPA при остановке интерфейса `eth1`.

```
#
# File: /etc/network/interfaces
#
# The primary network interface
auto eth1
iface eth1 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    wireless-essid homenet

    pre-up wpa_supplicant -Bw -Dwext -ieth1 -c/etc/wpa_supplicant/wpa_supplicant.conf
    post-down killall -q wpa_supplicant
```

При решении любых проблем, с которыми Вы можете встретиться, пожалуйста, обращайтесь в раздел по устранению проблем, имеющийся в настоящей главе.

### Конфигурирование Linux с несовместимыми беспроводными сетевыми платами

Не все беспроводные платы работают с Linux, особенно новые модели плат стандарта 54 Mbps 802.11g/n. К счастью имеется ряд способов преодолеть это очевидное ограничение. Они рассматриваются ниже.

### Использование `ndiswrapper`

Windows в качестве стандартизированного метода взаимодействия операционной системы с драйверами сетевых плат различных производителей использует спецификацию интерфейса сетевого драйвера (Network Driver Interface Specification - NDIS). Пакет `ndiswrapper` для Linux, который можно получить с сайта [ndiswrapper.sourceforge.net](http://ndiswrapper.sourceforge.net), позволит вам запускать под Linux драйвера Windows вашей сетевой платы. Для этого вокруг драйвера Windows создается программная обертка, заставляющая его думать, что он общается с Windows, а не с Linux. Диапазон совместимости таким образом расширяется, а для случаев, когда вам может потребоваться перекомпиляция ядра, на сайте проекта имеются ссылки на RPM

пакеты стандартных ядер с поддержкой ndiswrapper. Инструкции по установке на сайте проекта достаточно ясные и профессиональный пользователь Linux в состоянии сделать свою сетевую плату работоспособной в течение часа или двух от момента первой попытки.

Пакет ndiswrapper также имеет некоторые ограничения. Он работает только на архитектуре, поддерживающей Windows, не поддерживается очень полезная команда iwspy (см. обсуждение ниже), а обертка добавляет слой сложности программного взаимодействия, который в нормальных условиях отсутствует. У пакета ndiswrapper имеется коммерческий конкурент, называемый DriverLoader, созданный корпорацией Linuxant, который, возможно, Вы также захотите рассмотреть.

## Установка и конфигурирование ndiswrapper

1. Пакет ndiswrapper использует многие возможности ядра. В новых версиях of Fedora вам следует сначала установить RPM пакет ядра, предназначенный для разработки (kernel-devel). Пакет RPM должен быть на ваших установочных компакт дисках. Если Вы новичок в установке программного обеспечения Linux, не волнуйтесь. Скачать и установить пакеты RPM несложно. Если вам нужно вспомнить, как это делается, смотрите Главу 6 "Установка программного обеспечения Linux", где это подробно описано.

2. Остановите систему, установите вашу сетевую плату и перезагрузитесь. Скачайте tar файл ndiswrapper и распакуйте его содержимое. Войдите в директорию ndiswrapper и в файле INSTALL прочитайте инструкции по установке конкретной версии. В версии ndiswrapper-1.16, используемой в этом примере, для завершения процесса установки требуется выполнить команды make distclean, make и make install.

```
[root@bigboy tmp]# tar -xvzf ndiswrapper-1.16.tar.gz
[root@bigboy tmp]# cd ndiswrapper-1.16
[root@bigboy ndiswrapper-1.16]# make distclean
[root@bigboy ndiswrapper-1.16]# make
[root@bigboy ndiswrapper-1.16]# make install
```

Замечание: В дистрибутивах, основанных на Debian, таких как Ubuntu, ndiswrapper можно установить с помощью команды apt-get.

3. Далее мы должны определить PCI ID вашей вновь установленной сетевой платы. Сначала используйте команду lspci для того, чтобы найти номер IRQ сетевой платы. Значения IRQ будут указаны в первом столбце. В нашем случае IRQ равно 01:08.0.

```
[root@bigboy ndiswrapper-1.16]# lspci
...
...
01:08.0 Network controller: Intersil Corporation Prism 2.5 Wavelan
chipset (rev 01)
...
...
```

```
[root@bigboy ndiswrapper-1.16]#
```

4. Затем можно использовать команду lspci -n для того, чтобы получить PCI ID, который имеет формат xxxx:xxxx. Наша сетевая плата имеет ID, равный 1260:3873.

```
[root@bigboy ndiswrapper-1.16]# lspci -n
...
```

```
...
01:08.0 Class 0280: 1260:3873 (rev 01)
```

```
...
```

```
...
[root@bigboy ndiswrapper-1.16]#
```

5. На сайте `ndiswrapper` по ссылке, приведенной ниже, имеется таблица идентификаторов PCI ID и подходящих для них драйверов Windows.

<http://ndiswrapper.sourceforge.net/mediawiki/index.php/List>

Замечание: Используйте эту информацию для того, чтобы загрузить правильный драйвер для вашей сетевой платы. Не используйте драйвера Windows с компакт диска вашей сетевой платы, поскольку они, возможно, не были проверены разработчиками пакета `ndiswrapper`. В списке на сайте указаны названия драйверов, о которых известно, что они работают.

6. После загрузки извлеките файлы драйверов. Внутри главной директории, как правило, имеются поддиректории, соответствующие различным версиям Windows. Войдите в подкаталог, соответствующий самой последней версии.

```
[root@bigboy tmp]# unzip mzq345v25_xp_certd.zip
Archive: mzq345v25_xp_certd.zip
  inflating: mzq345v25_xp_certd_no_doc/autorun.exe
  inflating: mzq345v25_xp_certd_no_doc/autorun.inf
```

```
...
```

```
...
```

```
...
```

```
  inflating: mzq345v25_xp_certd_no_doc/winxp/NETMZQ345.INF
  inflating: MZQ345v25_Release_Note.TXT
```

```
[root@bigboy tmp]# cd mzq345v25_xp_certd_no_doc/winxp
[root@bigboy winxp]#
```

7. Основной файл драйвера Windows будет иметь расширение `.INF`. Установите этот драйвер при помощи команды `ndiswrapper` с опцией `-i`, за которой укажите имя файла драйвера.

```
[root@bigboy winxp]# ls
mzq345n51.sys NETMZQ345.INF
[root@bigboy winxp]# ndiswrapper -i NETMZQ345.INF
Installing netmzq345
[root@bigboy winxp]#
```

8. Используйте команду `ndiswrapper` снова с опцией `-l` для того, чтобы проверить, что установка была выполнена успешно.

```
[root@bigboy winxp]# ndiswrapper -l
Installed drivers:
netmzq345          driver installed, hardware present
[root@bigboy winxp]#
```

Замечание: Если Вы получили другое сообщение о драйвере, подобное тому, что мы видим ниже, то нужно будет выполнить дополнительные шаги, поскольку сообщение указывает, что Linux загрузил свой собственный драйвер для вашего устройства и это мешает работе `ndiswrapper`.

```
[root@bigboy winxp]# ndiswrapper -l
bcmwl5 : driver installed
        device (14E4:4320) present (alternate driver: bcm43xx)
[root@bigboy winxp]#
```

- Сначала вам нужно удалить из памяти драйвер Linux. В данном случае драйвер ndiswrapper обнаружил bcm43xx и его можно удалить с помощью команды rmmod. В некоторых случаях Вы можете получить сообщение о том, что драйвер зависит от другого драйвера – удалите оба драйвера с помощью команды rmmod.

```
[root@bigboy winxp]# rmmod bcm43xx
```

- Далее вам нужно предотвратить загрузку Linux-версии драйвера, когда Вы будете перезагружать систему. Добавьте запись для этого драйвера в черный список в файлы /etc/modprobe.d/blacklist-compat and /etc/modprobe.d/blacklist .

```
#
#
# File: /etc/modprobe.d/blacklist AND
# /etc/modprobe.d/blacklist-compat
#
blacklist bcm43xx
```

Если вам потребуется использовать команду rmmod более одного раза, то не забудьте добавить в черные списки все драйвера, которые Вы удалили.

- Теперь вам нужно с помощью команды rmmod с флагами -r и -I переустановить драйвер Windows так, как это показано ниже.

```
[root@bigboy winxp]# ndiswrapper -r Bcmwl5.inf
[root@bigboy winxp]# ndiswrapper -i Bcmwl5.inf
```

9. Далее следует модифицировать таблицы модулей ядра Linux с тем, чтобы добавить модуль ndiswrapper. Это делается с помощью команды depmod с флагом -a.

```
[root@bigboy winxp]# depmod -a
[root@bigboy winxp]#
```

10. Когда ndiswrapper будет загружен, вашей сетевой плате нужно будет назначить имя. Это делается с помощью команды ndiswrapper с флагом -m. Ниже показано, что имя нового устройства будет wlan0.

```
[root@bigboy winxp]# ndiswrapper -m
Adding "alias wlan0 ndiswrapper" to /etc/modprobe.d/ndiswrapper
[root@bigboy winxp]#
```

11. Теперь настало время с помощью команды modprobe загрузить модуль ядра ndiswrapper. Вы также можете проверить успешность выполнения этой операции, увидев в конце файла /var/log/messages сообщение о правильном выполнении команды.

```
[root@bigboy winxp]# modprobe ndiswrapper
[root@bigboy winxp]# tail /var/log/messages
```

```
...
...
```

```
Mar 17 23:25:21 bigboy kernel: ndiswrapper version 1.6
loaded (preempt=no,smp=no)
[root@bigboy winxp]#
```

Команда `dmesg` выдаст сообщения о статусе загрузки как драйвера вашей сетевой платы, так и для модуля `ndiswrapper`. Ошибок быть не должно. Если они имеются, то, возможно, Вы использовали драйвер, не рекомендованный на сайте `ndiswrapper`, возможно, ваша сетевая плата неисправна, возможно, ваша сетевая плата совместима с Linux, либо, возможно, ваша установка `ndiswrapper` или ядра были сделаны неправильно. Пожалуйста, смотрите подробности в разделе "Устранение проблем в вашей беспроводной сети" в этой главе.

```
[root@bigboy tmp]# dmesg
...
...
ndiswrapper version 1.16 loaded (preempt=no,smp=no)
ndiswrapper: driver mzq345 (Broadcom,04/21/2005, 3.100.65.1) loaded
ACPI: PCI Interrupt 0000:01:08.0[A] -> Link [LNKB] -> GSI 10
```

```
(level, low) -> IRQ 10
ndiswrapper: using irq 10
wlan0: vendor:
wlan0: ndiswrapper ethernet device 00:06:25:1b:b2:a9 using
driver mzq345, 14E4:4301.5.conf
wlan0: encryption modes supported: WEP; TKIP with WPA, WPA2,
```

```
WPA2PSK, WPA2, WPA2PSK
[root@bigboy tmp]#
```

12. Для того, чтобы приложение работало корректно, Вам всегда нужно иметь ядро, совместимое с `ndiswrapper`. Для того, чтобы сохранить текущее ядро во время модификации ядра, выполняемой с помощью команды `yum`, отредактируйте файл `/etc/yum.conf` и в опции `exclude` запретите обновление ядра.

```
#
# File: /etc/yum.conf
#
exclude=kernel
```

13. Используйте обычный пакет `wireless tools` для Linux, чтобы для интерфейса `wlan0` сконфигурировать IP адрес, идентификатор ESSID и, если это необходимо, задать шифрование. Для дистрибутива Fedora Вы можете указать скорость до 54 Мбайт/сек для 802.11g, добавив для этого в файл `/etc/sysconfig/network-scripts/ifcfg-wlan0` указанную ниже инструкцию. Оставьте файл пустым, если Вы используете стандарт 802.11b.

```
RATE=54Mb/s
```

14. Теперь Вы можете воспользоваться командой `ifup` для того, чтобы активировать сетевую плату, и командой `iwconfig`, которая покажет, что интерфейс подключен правильно к точке доступа на скорости 54 Мбит/сек.

```
[root@bigboy winxp]# ifup wlan0
```

```
[root@bigboy winxp]# iwconfig
...
wlan0 IEEE 802.11g ESSID:"johncr0w" Nickname:"bigboy"
      Mode:Managed Frequency:2.462GHz Access Point:
      00:09:5B:C9:19:22
      Bit Rate=54Mb/s Tx-Power:32 dBm
      RTS thr=2347 B Fragment thr=2346 B
      Encryption key:98D1-26D5-AC Security mode:restricted
      Power Management:off

      Link Quality:88/100 Signal level:-55 dBm Noise level:-
      256 dBm
      Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
      Tx excessive retries:96 Invalid misc:1157
      Missed beacon:0

...
[root@bigboy winxp]#
```

Мой опыт использования пакета дома был очень хорошим, но вам следует переустанавливать пакет каждый раз, когда вы модифицируете ваше ядро. Это неприемлемо в критически важных условиях ведения бизнеса, где простои, связанные с обслуживанием, должны быть сведены к минимуму и где для обеспечения стабильности работы все программное обеспечение должно быть на 100% совместимо с Linux.

Когда технология 802.11g WiFi станет более зрелой, она, несомненно, будет непосредственно поддерживаться пакетом Wireless Tools для Linux и не потребуется использовать дополнительное программное обеспечение, однако всегда будут сетевые платы, которые не поддерживаются Linux, и знание пакета ndiswrapper будет всегда ценно.

#### Устранение проблем в вашей беспроводной сети

Средства устранения проблем в беспроводной сети, имеющиеся в Linux, довольно обширны и предоставляют достаточно полезной информации, которая поможет сделать вашу сеть работоспособной. В этом разделе рассмотрены многие важные стратегии, облегчающие традиционные процедуры, такие как просмотр файла /var/log/messages.

#### Проверьте состояние сетевой платы

Когда используется технология беспроводных сетей, команды iwconfig, iwlist и iwspy могут предоставить вам полезную информацию о состоянии вашей сети. Рассмотрим их подробнее.

#### Команда iwconfig

В дополнение к обычной команде ifconfig, которая проверяет состояние вашей сетевой платы, Вы можете воспользоваться командой iwconfig для просмотра состояния вашей беспроводной сети – просто не указывайте каких-либо параметров. В частности, Вы можете увидеть такую важную информацию, как качество соединения, MAC адрес точки доступа WAP, скорость передачи данных, ключи шифрования, что позволит убедиться в том, что эти

параметры одинаковы для всей сети. Например:

```
[root@bigboy tmp]# iwconfig
eth0 IEEE 802.11-DS ESSID:"homenet" Nickname:"bigboy"
      Mode:Managed Frequency:2.462GHz Access Point: 00:09:5B:C9:19:22
      Bit Rate:11Mb/s Tx-Power=15 dBm Sensitivity:1/3

      Retry min limit:8 RTS thr:off Fragment thr:off
      Encryption key:98D1-26D5-AC Security mode:restricted
      Power Management:off
      Link Quality:36/92 Signal level:-92 dBm Noise level:-148 dBm

      Rx invalid nwid:0 Rx invalid crypt:2 Rx invalid frag:0
      Tx excessive retries:10 Invalid misc:0 Missed beacon:0
[root@bigboy tmp]#
```

Команда iwlist

Команда iwlist может дать дополнительную информацию, касающуюся не только сетевой платы, но и всей сети, в том числе количество активных частотных каналов, диапазон допустимых скоростей передачи данных и силу сигнала. В приведенном ниже примере команда применяется для проверки ключа шифрования, используемого сетевой платой - это очень полезно при возникновении проблем с безопасностью в вашей сети.

```
[root@bigboy tmp]# iwlist key
...
...
eth0 2 key sizes : 40, 104bits
      4 keys available :
          [1]: 9671-36DE-AC (40 bits)
          [2]: off
          [3]: off

          [4]: off
      Current Transmit Key: [1]
      Security mode:open
...
...
[root@bigboy tmp]#
```

Команда iwlist может проверять скорость, с которой работает сетевая плата, в нашем случае – 11Мбит/сек. Это полезно при определении возможных причин замирания сети, поскольку плохое качество сигнала может быть причиной низкой скорости взаимодействия сетевой платы с точкой доступа WAP.

```
[root@bigboy tmp]# iwlist rate
...
...
eth0 4 available bit-rates :
      1Mb/s
      2Mb/s
      5.5Mb/s
```

11Mb/s  
Current Bit Rate:11Mb/s

...  
...

```
[root@bigboy tmp]#
```

Дальнейшую информацию о команде iwlist смотрите на страницах man.

Команда iwspy

Команда iwspy предоставляет статистику по качеству соединения между вашей сетевой платой и другими беспроводными устройствами в сети. Она не работает все время; Вы должны сначала активировать команду iwspy для вашего интерфейса. Если команда iwspy не активирована, то она выдаст сообщение "no statistics to collect" ("статистика не собрана").

```
[root@bigboy root]# iwspy eth0  
eth0 No statistics to collect  
[root@bigboy root]#
```

Для активации требуется, чтобы Вы указали целевой IP адрес и беспроводный интерфейс сетевой платы, через который этот адрес может быть найден.

```
[root@bigboy tmp]# iwspy eth0 192.168.1.1
```

Если Вы используете команду iwspy без указания IP адреса, то она выдаст статистику сети с указанием типичного значения и значения, относительно которого делается сравнение. В примере, приведенном ниже, сигнал с качеством 64/92 можно рассматривать как сильный в сравнении с типичным значением 36/92, но он может быть слабым в другое время. Из-за возможности флуктуации сигнала следует периодически повторять такие проверки.

```
[root@bigboy tmp]# iwspy eth0  
eth0 Statistics collected:  
00:09:5B:C9:19:22 : Quality:0 Signal level:0 Noise level:0  
Link/Cell/AP : Quality:64/92 Signal level:-51 dBm Noise level:-149 dBm (updated)  
  
Typical/Reference : Quality:36/92 Signal level:-62 dBm Noise level:-98 dBm  
[root@bigboy tmp]#
```

Для того чтобы отключить мониторинг iwspy, добавьте аргумент off.

```
[root@bigboy root]# iwspy eth0 off
```

Проверка конфликтов прерываний

Устройствам, вставленным в слоты PCI шины вашего компьютера, система обычно назначает номер прерывания, который она использует для подачи сигнала в случае, если ей нужно взаимодействовать с этим устройством. Несколько устройств на шине могут иметь один и тот же номер прерывания, но система для того, чтобы избежать путаницы, будет получать доступ к каждому из них с различных адресов памяти. Иногда такое автоматическое выделение номеров прерываний (IRQ) и распределение памяти выполняется неверно, возникают накладки и происходит отказ устройства.

Прежде, чем конфигурировать программное обеспечение вашей беспроводной сети, Вы должны убедиться, что номер прерывания, присвоенный беспроводной сетевой плате, не конфликтует с другим устройством на вашем компьютере. Вставьте плату в пустой слот вашего Linux компьютера согласно инструкциям, изложенным в руководстве по компьютеру, перезагрузитесь и снова проверьте файл /var/log/messages.

```
[root@bigboy tmp]# tail -300 /var/log/messages
```

Внимательно поищите любые признаки, указывающие на то, что плата влияет на номера прерываний IRQ для ранее вставленных плат. Если конфликт существует, то обычно выдается предупреждение или сообщение вида "IRQ also used by ..." ("прерывание IRQ используется также ..."). В этом случае переставьте плату в другой слот, либо ликвидируйте конфликт вынув конфликтующее устройство, если оно в действительности вам не нужно.

Вам следует также проверить файл /proc/interrupts на наличие устройств, имеющих одно и то же прерывание.

```
[root@bigboy tmp]# cat /proc/interrupts
11: 4639 XT-PIC wlan0, eth0 (potentially bad)
```

```
[root@bigboy tmp]# cat /proc/interrupts
11: 4639 XT-PIC wlan0 (good)
```

Прим. пер.: potentially bad – потенциально плохо; good - хорошо

Конфликты по прерываниям обычно более проблематичны со старыми шинами PC-AT; более новые системы PCI обрабатывают конфликты лучше. Первый вариант (потенциально плохой) взят из функционирующего Linux компьютера, использующего шину PCI. Он работал потому, что хотя прерывание было то же самое, базовые адреса памяти, которые Linux использует для связи с платами, были различными. Вы можете проверить как прерывания, так и базовые адреса ваших сетевых плат с помощью команды ifconfig -a:

```
[root@bigboy tmp]# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:08:C7:10:74:A8
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:11 Base address:0x1820
wlan0 Link encap:Ethernet HWaddr 00:06:25:09:6A:B5
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:215233 errors:0 dropped:0 overruns:0 frame:0
TX packets:447594 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:39394014 (37.5 Mb) TX bytes:126738425 (120.8 Mb)

Interrupt:11 Memory:c887a000-c887b000
[root@bigboy tmp]#
```

## Ошибки ядра

Сообщения, касающиеся совместимости вашей беспроводной платы с вашей версией главной программы Linux – ядра, обычно можно найти одним из двух способов: увидеть в файле `/var/log/messages` или показать с помощью команды `dmesg`.

### Использование файла `/var/log/messages`

Если Вы найдете в файле `/var/log/messages` ошибку ядра `r80211`, то это обычно указывает на некорректно сконфигурированный идентификатор SSID, либо причиной ошибки может быть сетевая плата с устаревшей версией прошивки. Например:

```
Nov 13 22:24:54 bigboy kernel: p80211knetdev_hard_start_xmit: Tx attempt prior to association, frame dropped.
```

### Использование команды `dmesg`

Другой хороший источник информации – команда `the dmesg`, которая показывает ошибки, возникшие в ядре. В приведенном ниже примере указывается, что не найден микрокод для сетевой платы Broadcom 43XX. Эту ошибку можно устранить с помощью применения пакета `ndiswrapper`, как это описано в настоящей главе.

```
[root@bigboy tmp]# dmesg
...
...
bcm43xx: PHY connected
b43-phy0 debug: Adding Interface type 2
b43-phy0 ERROR: Microcode "bcm43xx_microcode5.fw" not available or load failed.
b43-phy0 ERROR: You must go to http://linuxwireless.org/en/users/Drivers/b43#devicefirmware
and download the correct firmware (version 4)

bcm43xx: core_up for active 802.11 core failed (-2)
[root@bigboy tmp]# dmesg
```

## Невозможно пропинговать шлюз, используемый по умолчанию

Если Вы не можете пропинговать шлюз, используемый по умолчанию, то сначала проверьте журнал ошибок ядра. Если в файле `/var/log/messages` ошибок нет, но Вы не можете пропинговать ваш шлюз или получить IP адрес, то проверьте конфигурационные файлы `/etc/sysconfig/network-scripts/` на предмет правильной конфигурации IP и проверьте таблицу маршрутизации с тем, чтобы убедиться, что там все правильно. С помощью команды `iwconfig` Вы можете также проверить, не выходит ли ваш Linux компьютер из диапазона адресов, обслуживаемых точкой доступа WAP.

## Ошибки "Unknown Device" - "Неизвестное устройство"

Следите за тем, чтобы во время установки или конфигурирования в журнальных файлах или на экране не появилось сообщение "unknown device" или "no such device" ("неизвестное устройство" или "нет такого устройства"). Причины таких сообщений могут быть следующие:

- Сетевая плата неправильно вставлена в PCI слот
- Несовместимое железо

Вы можете увидеть сообщения об ошибках, связанных с несовместимостью железа, в файле /var/log/messages:

```
00:0c.0 Network controller: BROADCOM Corporation: Unknown device 4301 (rev01)
Subsystem: Unknown device 1737:4301
Flags: bus master, fast devsel, latency 64, IRQ 5
Memory at f4000000 (32-bit, non-prefetchable) [size=3D8K]
```

Capabilities: [40] Power Management version 2

Либо Вы можете увидеть сообщения об ошибках на экране:

```
Dec 1 01:28:14 bigboy insmod: /lib/modules/2.4.18-14/net/<WBR>prism2_pci.o: init_module: No
such device Dec 1 01:28:14 bigboy insmod: Hint: insmod errors can be caused by incorrect module
parameters, including invalid IO or IRQ parameters. You may find more information in syslog or
the output from dmesg Dec 1 01:28:14 bigboy insmod: /lib/modules/2.4.18-
14/net/<WBR>prism2_pci.o: insmod wlan0 failed
```

### Ошибки чипсета Hermes

Я встречался со случаями, когда сетевая плата на чипсете Hermes, совместимая с Linux, отказывалась реагировать на запросы после того, как она нормально проработала несколько дней, а в файле /var/log/messages выдавалась следующая диагностика:

```
May 7 22:26:26 bigboy kernel: hermes @ e0854000: BAP0 offset timeout: reg=0x8044 id=0xfc80
offset=0x0 May 7 22:26:26 bigboy kernel: eth1: Error -110 setting multicast list. May 7 22:26:26
bigboy avahi-daemon[1701]: Withdrawing address record for 216.10.119.243 on eth1. May 7
22:26:26 bigboy avahi-daemon[1701]: Leaving mDNS multicast group on interface eth1.IPv4 with
address 216.10.119.243. May 7 22:26:26 bigboy avahi-daemon[1701]: IP_DROP_MEMBERSHIP
failed: No such device May 7 22:26:26 bigboy avahi-daemon[1701]: iface.c:
interface_mdns_mcast_join() called but no local address available. May 7 22:26:26 bigboy avahi-
daemon[1701]: Interface eth1.IPv4 no longer relevant for mDNS. May 7 22:26:27 bigboy kernel:
hermes @ e0854000: Timeout waiting for command 0x0002 completion. May 7 22:26:27 bigboy
kernel: eth1: Error -110 disabling MAC port May 7 22:26:31 bigboy kernel: hermes @ e0854000:
ng Error -16 issuing command 0x0021. May 7 22:26:31 bigboy kernel: hermes @ e0854000: Error -
16 issuing command 0x0021. May 7 22:26:31 bigboy kernel: eth1: Error -110 setting MAC address
May 7 22:26:31 bigboy kernel: eth1: Error -110 configuring card
```

Взаимодействие с платой обычно восстанавливалось только после перезагрузки. Наилучшем решением в этом случае было либо использование ndiswrapper, либо замена сетевой платы на действительно совместимое устройство.

### Ошибки Broadcom SoftMac

Если ваша конфигурация правильная, а ваша сетевая плата отказывается работать и выдает в файл /var/logs/messages повторяющиеся сообщения об отказе запроса на аутентификацию SoftMAC, так как это показано ниже, то, возможно, это проблема несовместимости вашей сетевой платы с Linux.

```
May 15 20:02:04 bigboy kernel: bcm43xx: set security called, .level = 0, .enabled = 0, .encrypt = 0
May 15 20:02:04 bigboy kernel: bcm43xx: set security called, .level = 0, .enabled = 0, .encrypt = 0
May 15 20:02:04 bigboy kernel: bcm43xx: set security called, .level = 0, .enabled = 0, .encrypt = 0
May 15 20:02:04 bigboy kernel: bcm43xx: set security called, .level = 0, .enabled = 0, .encrypt = 0
May 15 20:02:04 bigboy kernel: bcm43xx: set security called, .level = 0, .enabled = 0, .encrypt = 0
```

```
May 15 20:02:04 bigboy kernel: SoftMAC: Scanning finished: scanned 14 channels starting with channel 1
May 15 20:02:04 bigboy kernel: SoftMAC: Queueing Authentication Request to 00:18:39:ea:5c:ac
May 15 20:02:04 bigboy kernel: SoftMAC: Cannot associate without being authenticated, requested authentication
May 15 20:02:04 bigboy kernel: SoftMAC: Sent Authentication Request to 00:18:39:ea:5c:ac.
May 15 20:02:04 bigboy kernel: SoftMAC: generic IE set to dd160050f20101000050f202010000<WBR>50f20201000050f202
May 15 20:02:04 bigboy kernel: SoftMAC: Already associating or associated to 00:18:39:ea:5c:ac
May 15 20:02:04 bigboy kernel: SoftMAC: Open Authentication completed with 00:18:39:ea:5c:ac
May 15 20:02:04 bigboy kernel: SoftMAC: sent association request!
May 15 20:02:04 bigboy kernel: SoftMAC: associated!
May 15 20:02:04 bigboy kernel: SoftMAC: Associate: Scanning for networks first.
```

Попробуйте использовать `ndiswrapper` в качестве быстрого решения этой проблемы.

## Ошибки `ndiswrapper`

Имеется ряд общих ошибок, которые возникают при использовании `ndiswrappers`. Ниже приведены некоторые общие примеры.

## Ошибки `CONFIG_4KSTACKS` во время установки пакета

Иногда из-за несовместимости с ядром при установке `ndiswrapper` выдаются ошибки `CONFIG_4KSTACKS`, похожие на те, что приведены ниже:

```
*** WARNING: Kernel seems to have 4K size stack option (CONFIG_4KSTACKS) removed;
many Windows
drivers will need at least 8K size stacks. You should read wiki about 4K size stack issue. Don't
complain about crashes until you resolve this.
```

...

...

```
[root@bigboy ndiswrapper-1.16]#
```

Предупреждение гласит: Кажется, что в ядре удалена опция размера стека в 4К; многие драйвера Windows требуют стек размером как минимум 8К. Почитайте в `wiki` раздел о стеке размером 4К. Не жалуйтесь на сбои, пока не решите эту проблему.

Это обычная ситуация в случае, когда дистрибутив Fedora устанавливается по умолчанию, и `ndiswrapper` может отлично работать в этой ситуации. Если у вас нет ошибок типа `CONFIG_4KSTACKS` или хотите протестировать пакет `ndiswrapper` даже при наличии этих ошибок, то можно продолжить установку в обычном режиме. Ниже перечислены шаги, показывающие как избавиться от этих ошибок.

1. На сайте `ndiswrapper` приведены ссылки на сайты, откуда можно загрузить ядра с размером стека больше 16К. Ниже приведен приме такой ссылки. Это будет быстрее, чем создавать свое собственное ядро. <http://ndiswrapper.sourceforge.net/mediawiki/index.php/Fedora>

Помните, что нужно загрузить ядро, которое соответствует архитектуре вашей системы и версии ядра. Проверить это можно с помощью команды `uname -a`. Ниже видно, что наша система Fedora Core 5 с версией ядра 2.6.16-1.2122 для платформы i686.

```
[root@bigboy linux]# uname -rp
2.6.16-1.2122_FC5 i686
[root@bigboy linux]#
```

Если Вы скачали подходящее ядро, то загрузите его. Установите RPM, перезагрузитесь и затем продолжайте в соответствии с разделом "Установка и конфигурирование ndiswrapper".

Если Вы решили создать свое собственное ядро, то выполните следующие рекомендации.

2. Если Вы перешли к этому шагу, то это означает, что Вы решили перекомпилировать ядро. Это нетрудный процесс, состоящий из нескольких шагов, но время компиляции может быть большим. Первый шаг – установить файлы исходных кодов ядра. Он описывается в Главе 33 "[Модификация ядра для улучшения производительности](#)".

3. После установки исходных кодов ядра, Вы должны подготовить новое ядро, настроенное для использования с ndiswrapper, для компиляции. Первым шагом будет очистка всех временных файлов, которые могут существовать с предыдущих компиляций. Это можно сделать с помощью команды `make mrproper`. Затем вам нужно использовать команду `make oldconfig` для создания используемого по умолчанию варианта файла `.config`, который потребуется Linux при компиляции нового настроенного ядра.

```
[root@bigboy tmp]# cd /usr/src/linux
[root@bigboy linux]# make mrproper
[root@bigboy linux]# make oldconfig
```

4. Отредактируйте файл `.config` и установите переменную `CONFIG_4KSTACKS` равной "n".

```
[root@bigboy linux]# vi .config
#
# File: /usr/src/linux/.config
#
#CONFIG_4KSTACKS=y
CONFIG_4KSTACKS=n
[root@bigboy linux]#
```

5. Процесс компиляции ядра читает также файл `Makefile` для определения нового имени, которое будет использовать ядро. Переменная `EXTRAVERSION` в этом файле добавляет суффикс к имени ядра, это позволит вам отслеживать номера версий. Отредактируйте `Makefile` и задайте переменной `EXTRAVERSION` значение `-ndis-stk16` с тем, чтобы новое ядро легко идентифицировалось как версия, поддерживающая работу с ndiswrapper.

```
[root@bigboy linux]# vi Makefile
#
# File: /usr/src/linux/Makefile
#
EXTRAVERSION = -ndis-stk16
[root@bigboy linux]#
```

6. Откомпилируйте ядро и его модули с помощью следующих команд `make`. Убедитесь в том, что они закончились без ошибок и не забудьте, что это длительный процесс.

```
[root@bigboy linux]# make; make modules_install; make install
```

7. Если Вы установили новую версию ядра, то теперь вам надо сделать так, чтобы ваша система при перезагрузке выбирала правильную версию ядра. Для этого вам потребуется отредактировать файл `/etc/grub.conf` так, как это описано в Главе 33 "[Модификация ядра для улучшения производительности](#)".

8. Остановите систему, установите сетевую плату и перезагрузитесь. Система теперь загрузит

ваше новое ядро, это можно проверить с помощью команды `uname`.

```
[root@bigboy linux]# uname -r
2.6.16-ndis-stk16
[root@bigboy linux]#
```

9. Если Вы установили новую версию ядра и ваша система не смогла правильно загрузиться, то ищите подсказку в разделе "Восстановление ядра после его краха" в Главе 33 "Модификация ядра для улучшения производительности". Когда Вы заставите систему перезагрузиться правильно, снова пересмотрите шаги инсталляции и удостоверьтесь, что Вы изначально установили правильную версию.

После того, как новое ядро заработает, переустановите и сконфигурируйте `ndiswrapper`.

### Неправильные драйвера

Использование неправильного драйвера приведет к возникновению ошибок, которые можно увидеть, запустив команду `dmesg`. Ниже приведен пример сообщения об ошибке, в котором сообщается о проблеме с инициализацией драйвера:

```
[root@bigboy tmp]#
...
...
...
wlan0: ndiswrapper ethernet device 00:06:25:1b:b2:a9 using driver
wmp11v27, 14E4:4301:1737:4301.5.conf
ndiswrapper (set_auth_mode:702): setting auth mode to 3 failed
```

(C0010015)

```
[root@bigboy tmp]#
```

Наилучший способ исправить это – получить правильный драйвер, выгрузить модуль `ndiswrapper` из памяти, деинсталлировать старый драйвер, установить новый драйвер, а затем снова загрузить модуль `ndiswrapper`. Ниже перечисляются необходимые для этого шаги со всеми нужными командами:

1. Скачайте пакет драйверов из правильного источника и выберите драйвер для вашей системы Linux
2. С помощью команды `lsmod` проверьте, что модуль `ndiswrapper` уже загружен, а затем удалите его из памяти с помощью команды `rmmod`.

```
[root@bigboy tmp]# lsmod
Module              Size  Used by
...
...
ndiswrapper         145584  0
ipv6                 225504  16
autofs4             19204   1
[root@bigboy tmp]# rmmod ndiswrapper
```

3. С помощью команды `ndiswrapper` с флагом `-l` получите список установленных драйверов, а затем с помощью команды `ndiswrapper -r` удалите старый драйвер.

```
[root@bigboy tmp]# ndiswrapper -l
Installed drivers:
wmp11v27          driver installed, hardware present
[root@bigboy tmp]# ndiswrapper -r wmp11v27
[root@bigboy tmp]#
```

4. Установите новый драйвер с помощью команды `ndiswrapper -i` и с помощью команды `ndiswrapper -l` проверьте, что драйвер загружен.

```
[root@bigboy tmp]# ndiswrapper -i bcmwl5.inf
Installing bcmwl5
[root@bigboy tmp]# ndiswrapper -l
Installed drivers:
bcmwl5           driver installed, hardware present
[root@bigboy tmp]#
```

5. Используйте команду `depmod` чтобы перезагрузить таблицу модулей операционной системы.

```
[root@bigboy tmp]# depmod -a
```

6. Используйте команду `modprobe` для того, чтобы загрузить в память модуль `ndiswrapper`.

```
[root@bigboy tmp]# modprobe ndiswrapper
```

7. Наконец, с помощью команды `dmesg` проверьте, что с загрузкой не было проблем. Если проблем не было, сконфигурируйте ваш интерфейс `wlan0` аналогично другим интерфейсам сетевых плат вашей системы Linux

8. Даже если не было ошибок, на этой стадии может потребоваться перезагрузка для того, чтобы ваша беспроводная плата заработала.

Для уменьшения риска отказов при установке всегда следует использовать правильные драйвера. К счастью описанная выше процедура восстановления должна заставить вашу систему работать правильно.

### Сетевые платы, несовместимые с `ndiswrapper`

Предполагается, что модуль `ndiswrapper` работает в случае, если операционная система Linux не распознает сетевую плату. Если Linux распознает плату, то модуль не загрузится корректно. Команда `ndiswrapper -l` будет выдавать список установленных драйверов, в файле `/var/log/messages` будут новые записи модуля `ndiswrapper`, но команда `dmesg` вообще не обмолвится о состоянии процесса загрузки модуля `ndiswrapper`, а активация интерфейса `wlan0` не произойдет.

```
[root@bigboy tmp]# ifup wlan0
ndiswrapper device wlan0 does not seem to be present, delaying initialization.
[root@bigboy tmp]# ndiswrapper -l
Installed drivers:
netma311         driver installed, hardware present
```

```
[root@bigboy tmp]# dmesg | grep ndiswrapper
[root@bigboy tmp]#
```

В предыдущем примере приведены эти симптомы для случая, когда модуль `ndiswrapper` используется совместно с сетевой платой Netgear ma311, совместимой с Linux.

Ошибки `Invalid module format` – неверный формат модуля

Сам пакет `ndiswrapper` устанавливается как модель, непосредственно работающий с ядром Linux. Если Вы обновите ядро, то модуль `ndiswrapper` может прекратить работать. В таких случаях переустановка пакета `ndiswrapper` может вызвать следующие ошибки "Invalid module format" ("неверный формат модуля"):

```
[root@bigboy tmp]# modprobe ndiswrapper
FATAL: Error inserting ndiswrapper
(/lib/modules/2.6.23.9-85.fc8/misc/ndiswrapper.ko): Invalid module format
[root@bigboy tmp]#
```

Для того, чтобы решить проблему, нужно не забывать всегда запускать команду `make distclean` перед любой другой установкой, связанной с использованием команд `make`. Это гарантирует, что модуль всегда будет совместим с вашим новым ядром.

Беспроводные сети в бизнесе

Иногда для поддержки бизнеса нужно реализовать беспроводную сеть. Приходящим менеджерам может потребоваться быстрое подключение в конференц-зале; сотрудникам, занимающимся продажами, также может потребоваться беспроводное подключение, поскольку обычных рабочих мест может не хватить. Возможно, что кто-то собирается сделать это в вашей сети, но вам нужно контролировать процесс с самого начала.

Кроме того, что мобильные пользователи могут просто загрузить завирусированные программы и приложения к электронным письмам, их ноутбуки обычно рассматриваются как источник повышенного риска вредоносной деятельности, поскольку контроля над ними меньше, чем над теми, кто использует фиксированные рабочие места. Имея это в виду, обычно лучше изолировать такую беспроводную сеть от вашей внутренней проводной сети. В некоторых случаях делается так, что беспроводной маршрутизатор имеет доступ через специально выделенную линию DSL только в Интернет и никуда больше. В этом случае беспроводные пользователи должны воспользоваться каким-нибудь клиентом VPN для того, чтобы получить доступ к офисным серверам точно так, как они бы делали это из своего дома. Для уменьшения риска проникновения в сеть обязательно зашифруйте трафик и используйте прокси сервер, например, Squid (см. Главу 32 "[Управление Веб доступом с помощью Squid](#)") с тем, чтобы обеспечить доступ к Интернету только авторизованным пользователям, вводящим свое имя и пароль. При таком построении сети в случае, если беспроводная сеть будет взломана, ваши офисные системы останутся относительно защищенными.

Во многих точках беспроводного доступа WAP имеется возможность не показывать идентификатор ESSID, что мешает пользователям, находящимся поблизости, выбрать вашу сеть как ближайшую. Активация данной функции может быть неудобна для ваших пользователей, поскольку им потребуется знать идентификатор ESSID с тем, чтобы получить доступ в сеть, однако более важно уменьшить риск проникновения в вашу беспроводную сеть, возможного путем сканирования сигналов имеющихся точек доступа WAP.

Существует много других вариантов использования беспроводной технологии. Пожалуйста, изучите различные варианты прежде, чем прийти к окончательному решению.

## Заключение

Благодаря сведениям, полученным в главах первой части настоящей книги, Вы сможете сравнительно легко сконфигурировать для небольшой сети под Linux файловый и DHCP сервер. В части 2 мы изучим, как сделать так, чтобы ваш сервер стал ядром вашего самонастраивающегося выделенного веб сервера.

[Петер Харрисон - Беспроводная сеть Linux](#)