

Network configuration (Русский)

Ссылки по теме

- [Jumbo-кадры](#)
- [Firewalls](#)
- [Настройка беспроводной сети](#)
- [Сетевой мост](#)
- [Список приложений/Интернет#Управление подключениями](#)



Эта страница нуждается в сопроводителе



Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. [Команда переводчиков ArchWiki](#)

Состояние перевода: На этой странице представлен перевод статьи [Network configuration](#). Дата последней синхронизации: 2014-10-16. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#).



Эта статья или раздел нуждается в [переводе](#)



Примечания: Необходимо полностью переписать, очень устарела (обсуждение: [Talk:Network configuration \(Русский\)#](#))

На этой странице объясняется, как настроить **проводное** подключение к сети. Если вам необходимо настроить **беспроводное** подключение, смотрите статью [Настройка беспроводной сети](#).

Contents

[hide]

- [1Проверка подключения](#)
- [2Установка имени узла](#)
- [3Драйвер устройства](#)
 - [3.1Проверка состояния](#)
 - [3.2Загрузка модуля](#)
- [4Сетевые интерфейсы](#)
 - [4.1Имена устройств](#)
 - [4.1.1Получение текущих имен устройств](#)
 - [4.1.2Изменение имени устройства](#)
 - [4.2Установка MTU и длины очереди для устройства](#)
 - [4.3Включение и отключение сетевых интерфейсов](#)
- [5Настройка IP-адреса](#)
 - [5.1Динамический IP-адрес](#)
 - [5.1.1systemd-networkd](#)
 - [5.1.2dhcpcd](#)
 - [5.2Статический IP-адрес](#)
 - [5.2.1Ручное распределение](#)
 - [5.2.2Постоянная конфигурация при загрузке с использованием systemd](#)
 - [5.2.3Вычисление адресов](#)
- [6Загрузка конфигурации](#)
- [7Дополнительные настройки](#)
 - [7.1ifplugd для ноутбуков](#)
 - [7.2Объединение сетевых интерфейсов \(bonding\) или LAG](#)
 - [7.3Создание алиасов для IP-адресов](#)

- 7.3.1 Пример
 - 7.4 Изменение MAC/аппаратного адреса
 - 7.5 Раздача интернета
 - 7.6 Настройка маршрутизатора
 - 7.7 Разрешение имени узла (hostname resolution) локальной сети
 - 7.8 "Неразборчивый" режим
- 8 Решение проблем
 - 8.1 Смена компьютера при использовании кабельного модема
 - 8.2 Проблема масштабирования TCP window
 - 8.2.1 Диагностика
 - 8.2.2 Способы решения проблемы
 - 8.2.2.1 Плохой
 - 8.2.2.2 Хороший
 - 8.2.2.3 Лучший
 - 8.2.3 Дополнительная информация
 - 8.3 Realtek: нет соединения / проблема WOL
 - 8.3.1 Способ 1: включение сетевой платы непосредственно в Linux
 - 8.3.2 Способ 2: откат/замена драйвера Windows
 - 8.3.3 Способ 3: включение WOL в драйвере Windows
 - 8.3.4 Способ 4: более новый драйвер Realtek для Linux
 - 8.3.5 Способ 5: включение LAN Boot ROM в BIOS/CMOS
 - 8.4 Для чипсетов Atheros отсутствует интерфейс
 - 8.5 Broadcom BCM57780

Проверка подключения

Примечание: Если вы получаете сообщение вида `ping: icmp open socket: Operation not permitted` (операция не позволена) при вызове `ping`, попробуйте переустановить пакет [iputils](#)

В большом количестве случаев базовая процедура установки создает работающую конфигурацию сети. Чтобы это проверить, используйте следующую команду:

Примечание: Опция `-c 3` задает количество отправляемых запросов (три). Для получения дополнительной информации смотрите страницу справочного руководства [ping\(8\)](#)

```
$ ping -c 3 yandex.ru
```

```
PING yandex.ru (93.158.134.11) 56(84) bytes of data.
64 bytes from 93.158.134.11: icmp_req=1 ttl=54 time=15 ms
64 bytes from 93.158.134.11: icmp_req=2 ttl=54 time=17.4 ms
64 bytes from 93.158.134.11: icmp_req=3 ttl=54 time=19.4 ms

--- yandex.ru ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 15.076/17.320/19.444/1.791 ms
```

Если сеть работает, вы можете лишь настроить ее под свои потребности при помощи опций, описанных ниже.

Если предыдущая команда выдает ошибку о неизвестном узле (unknown host), это значит, что ваша машина не может преобразовать имя домена в IP-адрес. Это может быть связано с вашим провайдером или маршрутизатором/шлюзом. Вы можете попробовать отправить запросы на статический IP-адрес, чтобы проверить, имеет ли ваша машина доступ в интернет:

```
$ ping -c 3 93.158.134.11
```

```
PING 93.158.134.11 (93.158.134.11 56(84) bytes of data.  
64 bytes from 93.158.134.11: icmp_req=1 ttl=53 time=15.1 ms  
64 bytes from 93.158.134.11: icmp_req=2 ttl=54 time=17.7 ms  
64 bytes from 93.158.134.11: icmp_req=3 ttl=53 time=14.8 ms  
  
--- 93.158.134.11 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 14.840/15.917/17.788/1.335 ms
```

Примечание: 93.158.134.11 - это IP-адрес сервера yandex.ru. Вы можете использовать статический адрес первичного сервера DNS google.com, который очень легко запомнить: 8.8.8.8. Он считается весьма надежным и, как правило, не блокируется системами фильтрации содержимого и прокси-серверами

Если запросы на адрес 93.158.134.11 проходят, а на yandex.ru - нет, проверьте ваши настройки DNS. Для получения дополнительной информации смотрите статью resolv.conf.

Установка имени узла

Имя узла - уникальное имя, создаваемое для идентификации машины в сети. Оно прописывается в файле `/etc/hostname`. Этот файл может также содержать имя домена, если это необходимо. Чтобы установить имя узла, выполните:

```
# hostnamectl set-hostname моеимяузла
```

Эта команда создаст запись `моеимяузла` в файле `/etc/hostname`.

Для получения дополнительной информации смотрите страницы справочного руководства [hostname \(5\)](#) и [hostnamectl \(1\)](#).

Добавьте то же самое имя узла в `/etc/hosts`:

```
/etc/hosts  
  
#<ip-address> <hostname.domain.org> <hostname>  
127.0.0.1 localhost.localdomain localhost  
127.0.1.1 мое_имя_узла.localdomain мое_имя_узла  
::1 localhost.localdomain localhost
```

Примечание: Если вы используете постоянный IP-адрес, его необходимо использовать вместо 127.0.1.1

Чтобы установить имя узла временно (до перезагрузки), используйте команду `hostname` из пакета [inetutils](#):

```
# hostname моеимяузла
```

Драйвер устройства

Проверка состояния

[udev](#) должен определить вашу сетевую карту (смотрите статью в [Википедии](#): [Сетевая плата](#)) и автоматически загрузить необходимый модуль при старте системы. Посмотрите на пункт "Ethernet controller" (или похожий) в выводе команды `lspci -v`. Там должна быть информация о том, какой модуль ядра содержит драйвер для вашего сетевого устройства. Например:

```
$ lspci -v

02:00.0 Ethernet controller: Attansic Technology Corp. L1 Gigabit Ethernet
Adapter (rev b0)

...
Kernel driver in use: atl1
Kernel modules: atl1
```

Затем проверьте, был ли загружен драйвер, при помощи команды `dmesg | grep module_name`. Например:

```
$ dmesg | grep atl1

...
atl1 0000:02:00.0: eth0 link is up 100 Mbps full duplex
```

Пропустите следующий раздел, если драйвер был успешно загружен. В противном случае необходимо узнать, какой модуль требуется для вашей конкретной модели.

Загрузка модуля

Найдите в интернете необходимый для вашего чипсета модуль/драйвер. Вот некоторые общие модули: `8139too` для плат с чипсетом Realtek, `sis900` для плат с чипсетом SiS. Когда вы узнаете, какой модуль необходимо использовать, попробуйте [загрузить его вручную](#) [\[broken link: invalid section\]](#). Если вы увидите сообщение об ошибке, говорящее, что модуль не найден (the module was not found), возможно, драйвер не включен в состав ядра Arch. Вы можете осуществить поиск имени модуля в [AUR](#).

Если `udev` не определяет и не загружает нужный модуль автоматически во время старта системы, обратитесь к разделу [Модули ядра#Загрузка](#) [\[broken link: invalid section\]](#).

Сетевые интерфейсы

Имена устройств

Для компьютеров, имеющих несколько сетевых плат, важно иметь фиксированные имена устройств. Множество проблем с настройкой вызвано изменением имен интерфейсов.

[udev](#) отвечает за то, какое имя дается каждому устройству. В `systemd v197` появились [предсказуемые имена сетевых интерфейсов](#), благодаря чему сетевым устройствам автоматически присваиваются статические имена. У них теперь есть префиксы: `en` (ethernet), `wl` (WLAN) или `ww` (WWAN), после которых следует автоматически сгенерированный идентификатор. В итоге получаются имена наподобие `enp0s25`.

Такое поведение может быть отключено путем добавления `net.ifnames=0` в командную строку ядра.

Совет: Вы можете выполнить команду `ip link` или `ls /sys/class/net`, чтобы увидеть список всех доступных интерфейсов

Примечание: При изменении схемы именования интерфейсов не забудьте обновить все конфигурационные файлы, имеющие отношение к сети, и пользовательские файлы юнитов

systemd. Конкретно, если у вас включены [статические профили netctl](#), выполните `netctl reenable профиль`, чтобы обновить сгенерированный файл службы

Получение текущих имен устройств

Текущие имена сетевых плат можно узнать при помощи `sysfs`:

```
$ ls /sys/class/net  
  
lo eth0 eth1 firewire0
```

Изменение имени устройства

Вы можете изменить имя устройства, установив его вручную при помощи правила udev. Например:

```
/etc/udev/rules.d/10-network.rules  
  
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="aa:bb:cc:dd:ee:ff",  
NAME="net1"  
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="ff:ee:dd:cc:bb:aa",  
NAME="net0"
```

Кое-что на заметку:

- Чтобы увидеть MAC-адрес каждой платы, используйте команду `cat /sys/class/net/имя_устройства/address`
- Убедитесь, что в ваших правилах udev используются шестнадцатиричные значения со строчным написанием. Буквы не должны быть прописными

Если сетевая плата имеет динамический MAC-адрес, вы можете использовать `DEVPATH`, например:

```
/etc/udev/rules.d/10-network.rules  
  
SUBSYSTEM=="net", DEVPATH=="/devices/platform/wemac.*", NAME="int"
```

Примечание: При выборе статических имен **вы должны избегать использования формата "ethX" и "wlanX"**, поскольку это может привести к "гонке" между ядром и udev во время загрузки системы. Вместо этого лучше взять имена интерфейсов, которые не используются по умолчанию в ядре, например: `net0`, `net1`, `wifi0`, `wifi1`. Для получения дополнительной информации, пожалуйста, смотрите документацию по [systemd](#)

Установка MTU и длины очереди для устройства

Вы можете изменить MTU и длину очереди для устройства, определив их вручную в правиле udev. Например:

```
/etc/udev/rules.d/10-network.rules  
  
ACTION=="add", SUBSYSTEM=="net", KERNEL=="wl*", ATTR{mtu}="1480",  
ATTR{tx_queue_len}="2000"
```

Включение и отключение сетевых интерфейсов

Вы можете включать или отключать сетевые интерфейсы при помощи:

```
# ip link set eth0 up
# ip link set eth0 down
```

Для проверки результата:

```
$ ip link show dev eth0

2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master br0 state UP mode DEFAULT qlen 1000
...
```

Настройка IP-адреса

У вас есть два варианта: динамически присваиваемый при помощи [DHCP](#) или неизменяемый "статический" адрес.

Динамический IP-адрес

systemd-networkd

Легкий способ настройки DHCP без особых требований - использовать службу [systemd-networkd](#), которую предоставляет systemd. Смотрите раздел [Основная сеть DHCP](#).

dhcpcd

[dhcpcd](#) используется для настройки DHCP в установочном образе Arch Linux в качестве клиента по умолчанию. Это более мощный инструмент, который позволяет указать большее количество опций клиента DHCP. Чтобы узнать, как его активировать для интерфейса, смотрите раздел [Запуск](#).

Статический IP-адрес

Есть несколько причин, по которым вы можете захотеть присвоить статический IP-адрес. Например, можно получить определенную степень предсказуемости с неизменными адресами, или у вас недоступен сервер DHCP.

Статический адрес можно настроить при помощи большинства сетевых инструментов, стандартных для Arch Linux, например, при помощи [netctl](#), [systemd-networkd](#) или [dhcpcd](#).

В следующем списке описано, что необходимо указать для **ручной** настройки статического IP-адреса:

- Статический IP-адрес
- [Маска подсети](#)
- [Широковещательный адрес](#)
- IP-адрес [шлюза](#)

Если вы работаете в частной сети, можно с уверенностью использовать IP-адреса из диапазона 192.168.*.*, маску подсети 255.255.255.0 и широковещательный адрес 192.168.*.255. Шлюзу обычно присваивается значение 192.168.*.1 или 192.168.*.254.

Важно: Удостоверьтесь, что IP-адрес, присвоенный вручную, не конфликтует с теми, что присвоены при помощи DHCP. Смотрите [ветку форума](#)

Совет: Возможно, вам понадобится вручную указать серверы DNS. Для получения дополнительной информации смотрите статью [resolv.conf](#)

Примечание: Если вы получаете интернет-соединение от машины с Windows без использования маршрутизатора, во избежание проблем с LAN убедитесь в том, что обоим компьютерам присвоены статические IP-адреса

Ручное распределение

Вы можете присвоить статический IP-адрес через консоль:

```
# ip addr add IP-адрес/маска_подсети broadcast широковещательный_адрес dev
интерфейс
```

Например:

```
# ip addr add 192.168.1.2/24 broadcast 192.168.1.255 dev eth0
```

Примечание: Маска подсети была указана при помощи [бесклассовой адресации](#)

Чтобы увидеть список всех опций, смотрите страницу справочного руководства [ip\(7\)](#).

Добавьте ваш IP-адрес шлюза, как здесь:

```
# ip route add default via шлюз_по_умолчанию
```

Например:

```
# ip route add default via 192.168.1.1
```

Если вы получите сообщение об ошибке "Такого процесса не существует" (No such process), это значит, что вы запустили `ip link set dev eth0 up` от имени суперпользователя.

Постоянная конфигурация при загрузке с использованием systemd

Для начала создайте конфигурационный файл службы [systemd](#), при этом замените `интерфейс` на ваше имя сетевого интерфейса:

```
/etc/conf.d/net-conf-интерфейс
```

```
address=192.168.1.2
netmask=24
broadcast=192.168.1.255
gateway=192.168.1.1
```

Создайте скрипт для запуска сети:

```
/usr/local/bin/net-up.sh
```

```
#!/bin/bash
ip link set dev "$1" up
ip addr add ${address}/${netmask} broadcast ${broadcast} dev "$1"

[[ -z ${gateway} ]] || {
    ip route add default via ${gateway}
}
```

Скрипт для остановки сети:

```
/usr/local/bin/net-down.sh
```

```
#!/bin/bash
ip addr flush dev "$1"
ip route flush dev "$1"
ip link set dev "$1" down
```

Сделайте оба скрипта исполняемыми:

```
# chmod +x /usr/local/bin/net-{up,down}.sh
```

Файл службы *systemd*:

```
/etc/systemd/system/network@.service

[Unit]
Description=Network connectivity (%i)
Wants=network.target
Before=network.target
BindsTo=sys-subsystem-net-devices-%i.device
After=sys-subsystem-net-devices-%i.device

[Service]
Type=oneshot
RemainAfterExit=yes
EnvironmentFile=/etc/conf.d/net-conf-%i
ExecStart=/usr/local/bin/net-up.sh %i
ExecStop=/usr/local/bin/net-down.sh %i

[Install]
WantedBy=multi-user.target
```

Включите и запустите юнит `network@интерфейс`, указав ваше имя интерфейса.

Совет: Если вы предпочитаете обходиться без скриптов, добавьте необходимые команды в качестве дополнения к строкам `ExecStart=` и `ExecStop=`. Смотрите [пример для беспроводной сети](#)

Вычисление адресов

Вы можете использовать `ipcalc`, предоставляемый пакетом [ipcalc](#), для вычисления широковещательного IP, сети, маски и рангов узлов, чтобы получить более сложную конфигурацию. Пример - использование Ethernet и Firewire для подключения машины с Windows к Linux. Для улучшения организации сети и ее безопасности каждая машина имеет свою собственную сеть с соответствующим образом настроенными маской сети и широковещательным адресом.

Чтобы вычислить соответствующие маску сети и широковещательные адреса, можно использовать `ipcalc`, задавая IP-адрес сетевой платы Linux `10.66.66.1` и количество узлов (в данном примере два):


```
$ ipcalc -nb 10.66.66.1 -s 1
```

```
Address: 10.66.66.1
```

```
Netmask: 255.255.255.252 = 30
```

```
Network: 10.66.66.0/30
```

```
HostMin: 10.66.66.1
```

```
HostMax: 10.66.66.2
```

```
Broadcast: 10.66.66.3
```

```
Hosts/Net: 2
```

```
Class A, Private Internet
```

Загрузка конфигурации

Чтобы протестировать ваши настройки либо перезагрузите компьютер, либо перезапустите соответствующие службы systemd. Затем попробуйте отправить запросы (ping) на ваши шлюз, DNS-сервер, провайдера интернет-услуг и другие сайты в интернете, именно в этом порядке, чтобы определить, в чем заключается проблема подключения. Вот пример:

```
$ ping -c 3 yandex.ru
```

Дополнительные настройки

ifplugd для ноутбуков

Совет: [dhcpcd](#) предоставляет ту же возможность "из коробки"

[ifplugd](#) из [официальных репозиториев](#) - это демон, который автоматически настроит ваше устройство Ethernet при подключении кабеля и удалит конфигурацию при его отключении. Это полезно для ноутбуков со встроенными сетевыми адаптерами, поскольку интерфейс будет настроен лишь при реальном подключении кабеля. Другой вариант использования - когда вам необходимо лишь перезапустить сеть, но не компьютер, без необходимости делать это в оболочке.

По умолчанию ifplugd настроен на работу с устройством `eth0`. Эта и другие настройки, такие как время задержки, можно изменить в файле `/etc/ifplugd/ifplugd.conf`.

Примечание: Пакет [netctl](#) содержит службу `netctl-ifplugd@.service`, в ином случае вы можете использовать `ifplugd@.service` из пакета [ifplugd](#). Выполните, например, `systemctl enable ifplugd@eth0.service`

Объединение сетевых интерфейсов (bonding) или LAG

Смотрите раздел [netctl \(Русский\)#Объединение сетевых интерфейсов \(бондинг\)](#).

Создание алиасов для IP-адресов



Эта статья или раздел нуждается в [переводe](#)



Примечания: должен быть добавлен ручной метод с использованием [ip](#), а затем текущий пример, использующий `netctl`, должен быть перемещен в статью [netctl \(Русский\)](#) (обсуждение: [Talk:Network configuration \(Русский\)#](#))

Создание алиасов для IP-адресов - это процесс добавления больше, чем одного IP-адреса, к сетевому интерфейсу. Благодаря этому один узел сети может иметь несколько подключений, каждое из которых используется в своих целях. Типичное использование этой возможности - виртуальный хостинг серверов Web и FTP или реорганизация серверов без необходимости

обновления каких-либо других машин (это особенно полезно для серверов имен (nameservers)).

Пример

Вам понадобится пакет [netctl](#) из [официальных репозиториев](#).

Подготовьте конфигурацию:

```
/etc/netctl/моясеть

Connection='ethernet'
Description='Five different addresses on the same NIC.'
Interface='eth0'
IP='static'
Address=('192.168.1.10' '192.168.178.11' '192.168.1.12' '192.168.1.13'
'192.168.1.14' '192.168.1.15')
Gateway='192.168.1.1'
DNS=('192.168.1.1')
```

Затем просто выполните:

```
$ netctl start моясеть
```

Изменение MAC/аппаратного адреса

Смотрите статью [Смена MAC адреса](#).

Раздача интернета

Смотрите статью [Раздача интернета](#).

Настройка маршрутизатора

Смотрите статью [Маршрутизатор](#).

Разрешение имени узла (hostname resolution) локальной сети

Необходимым условием является [#Установка имени узла](#), после чего разрешение имени узла работает в локальной системе самостоятельно:

```
$ ping имя_узла

PING archpc (192.168.33.187) 56(84) bytes of data.
64 bytes from archpc (192.168.33.187): icmp_seq=1 ttl=64 time=0.082 ms
```

Чтобы позволить другим машинам обращаться к узлу по имени, необходимо либо вручную настроить соответствующие файлы `/etc/hosts`, либо создать службу распространения/разрешения имени.

Когда настройка сервера DNS, такого как [BIND](#) или [Unbound](#), является излишней, ручное редактирование вашего файла `/etc/hosts` слишком обременительно или вы хотите получить больше гибкости при отсоединении/присоединении узлов сети, можно оставить разрешение имени узла в вашей локальной сети, используя сеть нулевой конфигурации (zero-configuration networking). Доступны два варианта:

- [Samba](#) предоставляет разрешение имени узла при помощи **NetBIOS** от Microsoft. Требуется лишь установка пакета [samba](#) и включение службы `nmbd.service`. Компьютеры с Windows, OS X или Linux с запущенным `nmbd` будут способны найти вашу машину
- [Avahi](#) предоставляет разрешение имени узла при помощи **zeroconf**, также известного как Avahi или Bonjour. Требуется немного более сложная настройка, чем при использовании Samba: для получения дополнительной информации смотрите раздел [Разрешение имени узла](#). Компьютеры с Windows, OS X или Linux с запущенным демоном Avahi будут способны найти вашу машину. Windows не имеет встроенного клиента или демона Avahi

"Неразборчивый" режим

Включение "[неразборчивого](#)" режима заставит (беспроводную) сетевую плату перенаправлять весь трафик, который она получает, в операционную систему для дальнейшей обработки. Это противоположность "нормальному режиму", при котором сетевая плата будет терять пакеты, не предназначенные для приема. Чаще всего эта возможность используется для продвинутого решения сетевых проблем и [анализа пакетов](#).

```
/etc/systemd/system/promiscuous@.service

[Unit]
Description=Set %i interface in promiscuous mode
After=network.target

[Service]
Type=oneshot
ExecStart=/usr/bin/ip link set dev %i promisc on
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Если вы хотите включить "неразборчивый" режим для интерфейса `eth0`, выполните:

```
# systemctl enable promiscuous@eth0.service
```

Решение проблем

Смена компьютера при использовании кабельного модема

Некоторые провайдеры кабельных интернет-услуг (например, videotron) настраивают кабельный модем на работу только с одним клиентом-компьютером по MAC-адресу его сетевого интерфейса. Как только модем запомнит MAC-адрес первого подключенного компьютера или оборудования, он ни при каких обстоятельствах не будет отвечать на запросы, идущие с других MAC-адресов. Таким образом, если вы поменяете один компьютер на другой (или поставите маршрутизатор), новый компьютер (или маршрутизатор) не будет работать с кабельным модемом, поскольку он имеет MAC-адрес, отличный от предыдущего. Для сброса кабельного модема с тем, чтобы он стал работать с новым компьютером, необходимо выключить питание кабельного модема и включить его опять. Как только он перезагрузится и подключится к сети (загорятся соответствующие индикаторы), перезагрузите вновь подключенный компьютер, чтобы он выполнил запрос DHCP, или вручную заставьте его запросить новый адрес DHCP.

Если это не поможет, необходимо скопировать MAC-адрес изначальной машины. Смотрите также раздел [#Изменение MAC/аппаратного адреса](#).

Проблема масштабирования TCP window

Пакеты TCP содержат в своих заголовках значение "window", обозначающее, как много данных другие узлы могут посылать в ответ. Это значение может содержать только 16 бит информации, следовательно, размер window должен быть не более 64Кб. Пакеты TCP на некоторое время кэшируются (они должны быть перераспределены), а, поскольку память ограничена, один узел может легко перевалить за это значение.

В далеком 1992 году становилось доступно все больше и больше памяти, и для улучшения ситуации был написан [RFC 1323](#): Window Scaling. Значение "window", содержащееся во всех пакетах, будет изменено при помощи коэффициента масштабирования (Scale Factor), определяемого один раз в самом начале подключения. Этот 8-битный коэффициент масштабирования позволяет Window быть в 32 раза больше, чем изначальные 64Кб.

Похоже, некоторые нестандартные маршрутизаторы и межсетевые экраны в интернете переписывают этот коэффициент в значение 0, что вызывает недопонимание между узлами. В ядре Linux версии 2.6.17 была представлена новая схема подсчета, генерирующая максимальные коэффициенты масштабирования и виртуально делающая последующие подсчеты нестандартных маршрутизаторов и межсетевых экранов более видимыми.

В итоге соединение в лучшем случае очень медленное или часто рвется.

Диагностика

Прежде всего, необходимо разъяснить: это странная проблема. В некоторых случаях вы не сможете по-полной использовать соединения TCP (HTTP, FTP и т.д.), в других вы сможете обращаться к некоторым узлам (лишь несколькими).

Если у вас появилась такая проблема, вывод `dmesg` будет нормальным, логи - чистыми, а `ip addr` сообщит о нормальном состоянии... Все будет выглядеть нормально.

Если вы не можете просматривать никакие веб-сайты, но можете отправлять запросы ping на некоторые узлы, высока вероятность, что у вас именно эта проблема: ping использует ICMP и не затрагивается проблемами TCP.

Вы можете попробовать использовать [Wireshark](#). В итоге вы можете получить успешные соединения UDP и ICMP, но неудачные соединения TCP (только для неизвестных узлов).

Способы решения проблемы

Плохой

Плохой способ заключается в изменении значения `tcp_rmem`, на котором основывается подсчет коэффициента масштабирования. Несмотря на то, что это должно помочь для большинства узлов, это не гарантирует успеха, особенно для очень удаленных из них.

```
# echo "4096 87380 174760" > /proc/sys/net/ipv4/tcp_rmem
```

Хороший

Просто отключите масштабирование Window. Поскольку оно - лишь приятная функция TCP, это может быть некомфортно, особенно, если вы не можете исправить проблему с нестандартным маршрутизатором. Есть несколько способов отключения этого масштабирования, и, кажется, наиболее "пуленепробиваемый" из них (который будет работать с большинством ядер) - добавление следующей строки в файл `/etc/sysctl.d/99-disable_window_scaling.conf` (смотрите также статью [sysctl](#)):

```
net.ipv4.tcp_window_scaling = 0
```

Лучший

Проблема вызвана нестандартными маршрутизаторами/межсетевыми экранами, поэтому замените их. Некоторые пользователи отмечали, что нестандартным маршрутизатором был их собственный маршрутизатор DSL.

Дополнительная информация

Этот раздел основывается на статьях LWN [Масштабирование window TCP и нестандартные маршрутизаторы](#) и Kernel Trap [Масштабирование Window в интернете](#).

На странице LKML есть также несколько ссылок по теме.

Realtek: нет соединения / проблема WOL

Пользователи с сетевыми платами, основанными на Realtek 8168 8169 8101 8111(C) (отдельными/встроенными) могут заметить проблему, что карта, кажется, отключена во время загрузки системы, и лампочка-индикатор не горит. Такое часто встречается на машинах с двумя операционными системами, на которых также установлена Windows. Похоже, что причиной являются обычные официальные драйверы Realtek (датирующие все после мая 2007 г.) под Windows. Эти новые драйверы отключают функцию Wake-On-LAN, отключая сетевую плату при завершении работы Windows, и она остается выключенной до следующей загрузки Windows. Вы сможете заметить это, если индикатор подключения не горит, пока не будет загружена Windows; во время ее завершения работы индикатор выключается. В нормальном состоянии лампочка всегда должна гореть, пока система работает, даже во время POST. Эта проблема также затрагивает другие операционные системы, не имеющие новейших драйверов (например, Live CD). Есть несколько способов решения этой проблемы.

Способ 1: включение сетевой платы непосредственно в Linux

Узнайте имя сетевой платы ethernet из вывода команды:

```
$ ip a
```

Поднимите устройство от имени суперпользователя, используя это имя:

```
# ip link set dev имя_сетевой_платы up
```

Например, если *имя_сетевой_платы* - enp7s0:

```
# ip link set dev enp7s0 up
```

Если это работает и сетевая карта включится, в выводе команды `ip link` вы должны увидеть `state UP` для нужного вам интерфейса.

Способ 2: откат/замена драйвера Windows

Вы можете откатить ваш драйвер сетевой платы в Windows на тот, который предоставляет Microsoft (если это возможно), или откатить/установить официальный драйвер Realtek, имеющий дату выпуска ранее мая 2007 г. (может найтись на компакт-диске, идущем в комплекте с вашим аппаратным обеспечением).

Способ 3: включение WOL в драйвере Windows

Наверное, самое лучшее и быстрое решение - изменить эту настройку в драйвере Windows. Тогда это затронет всю систему, в том числе Arch (а также live CD и другие операционные системы). В менеджере устройств Windows найдите ваш сетевой адаптер Realtek и сделайте на нем двойной щелчок мыши. Во вкладке "Дополнительно" измените значение "Wake-on-LAN после завершения работы" (Wake-on-LAN after shutdown) на "Включено".

В Windows XP (пример):

Кликните правой кнопкой мыши на "Мой компьютер" и зайдите в "Свойства"
--> Вкладка "Аппаратное обеспечение" (Hardware)
--> Менеджер устройств
--> Сетевые адаптеры
--> "двойной клик" на Realtek ...
--> Вкладка "Дополнительно"
--> Wake-On-Lan после завершения работы
--> Включено

Примечание: В новых драйверах Realtek для Windows (протестировано на *Realtek 8111/8169 LAN Driver v5.708.1030.2008*, датированном 2009/01/22, от GIGABYTE) доступ к этой опции может быть немного иным, например, *Shutdown Wake-On-LAN --> Включено*. Похоже, что ее переключение в значение *Отключено* не дает эффекта (вы увидите, что индикатор соединения по-прежнему выключается при завершении работы Windows). Довольно грязный обходной путь - загрузиться в Windows и просто перезагрузить (reset) систему (плохой restart/shutdown), что не даст драйверу Windows никаких шансов отключить LAN. Лампочка будет по-прежнему гореть, а адаптер LAN останется доступен после POST до тех пор, пока вы опять не запустите Windows и правильно завершите его работу

Способ 4: более новый драйвер Realtek для Linux

Любой более новый Linux-драйвер для этих карт Realtek можно найти на их сайте (не протестировано, но также должно решать проблему).

Способ 5: включение *LAN Boot ROM* в BIOS/CMOS

Похоже, что установка *Интегрированная периферия (Integrated Peripherals) --> Встроенный (Onboard) LAN Boot ROM --> Включено* в BIOS/CMOS возобновляет работу чипа Realtek LAN при загрузке системы, несмотря на то, что драйвер Windows делает обратное при завершении работы ОС.

Примечание: Этот способ был несколько раз протестирован на материнской плате GIGABYTE GA-G31M-ES2L с BIOS версии F8, выпущенным 2009/02/05

Для чипсетов Atheros отсутствует интерфейс

Пользователи некоторых чипов Atheros ethernet сообщают, что они не работают "из коробки" (с установочного носителя февраля 2014 г.). Помогает установка пакета [backports-patched](#)^{AUR} из AUR.

Broadcom BCM57780

Этот чипсет Broadcom иногда работает плохо, пока вы не укажете порядок загрузки модулей. Необходимые модули - `broadcom` и `tg3`, и они должны быть загружены в соответствующем порядке.

Эти шаги должны помочь, если в вашем компьютере используется этот чипсет:

- Найдите вашу сетевую плату в выводе `lspci`:

```
$ lspci | grep Ethernet
02:00.0 Ethernet controller: Broadcom Corporation NetLink BCM57780 Gigabit
Ethernet PCIe (rev 01)
```

- Если беспроводная сеть не функционирует, попробуйте отключить кабель и выполнить следующее:

```
# modprobe -r tg3
```

```
# modprobe broadcom
# modprobe tg3
```

- Подключите обратно кабель сети. Если это решает ваши проблемы, можете сделать это изменение постоянным, добавив модули `broadcom` и `tg3` (в этом порядке) в секцию `MODULES` файла `/etc/mkinitcpio.conf`:

```
MODULES=".. broadcom tg3 .."
```

- Пересоберите `initramfs`:

```
# mkinitcpio -p linux
```

- В качестве альтернативы вы можете создать файл `/etc/modprobe.d/broadcom.conf` с таким содержимым:

```
softdep tg3 pre: broadcom
```

Примечание: Этот способ может помочь и для других чипсетов, таких как BCM57760

[Categories:](#)

- [Networking \(Русский\)](#)
- [Русский](#)