

FreeNIBS

Модуль авторизации, аутентификации и учета объема услуг для радиус-сервера FreeRADIUS.

Документация.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
FreeNIBS. Что это?.....	3
Инсталляции, отличия версий и общий план разработки.....	3
Как помочь в разработке FreeNIBS?.....	4
ФУНКЦИОНАЛЬНЫЕ ОСОБЕННОСТИ.....	5
КАК РАБОТАЕТ FreeNIBS?.....	6
Авторизация и аутентификация.....	6
Эккаунтинг.....	8
УСТАНОВКА FreeNIBS.....	9
Структура базы данных FreeNIBS (для версий 1 и 2).....	10
Структура базы данных FreeNIBS (для версии 3).....	14
Содержимое файла NIBS.CONF.....	23
НАСТРОЙКА ОТДЕЛЬНЫХ ПАРАМЕТРОВ.....	27
Настройка профессионального режима (до версии 3).....	27
Зависшие сессии.....	28
Описание функционирования системы черных/белых списков.....	29
ОБНОВЛЕНИЕ FreeNIBS.....	30
Приложение 1. Сокращения в SQL-запросах.....	31
Приложение 2. Конфигурация DialUp-сервера на базе CISCO.....	32
Приложение 3. Установка MPD+FreeNIBS под FreeBSD.....	34
FAQ.....	39

ВВЕДЕНИЕ.

Документация на FreeNIBS основана на документации FreeNIBS версии 0.0.3, содержит оригинальный текст и личное мнение ее авторов. Данная документация будет дорабатываться по мере возможности и по мере появления новых функциональных возможностей FreeNIBS.

Автор данного документа не принимает на себя никаких обязательств по развитию, совершенствованию, пополнению и т.п. Как данного документа, так и всего модуля FreeNIBS.

FreeNIBS. Что это?

FreeNIBS - модуль авторизации, аутентификации и учета объема услуг (биллинга) для FreeRADIUS. Модуль работает с базами данных MySQL, PostgreSQL и Oracle. Возможно написание драйверов к другим СУБД.

Модуль осуществляет биллинг (препайд и постпайд) в режиме реального времени. Задание правил биллинга и сервиса как для групп пользователей, серверов доступа, так и для отдельных пользователей. Модуль обладает развитыми функциями настройки.

Модуль работает на всех платформах и операционных системах, на которых работает радиус-сервер FreeRADIUS.

FreeRADIUS - радиус-сервер для UNIX-подобных операционных систем, распространяемый, в том числе, в исходных текстах под лицензией GPL. Официальная страница - <http://www.freeradius.org>

Модуль, как и радиус-сервер, является свободно распространяемым и бесплатным программным обеспечением. Модуль распространяется в исходных текстах под лицензией GPL.

Данный модуль, начнем отсчет с версии FreeNIBS 0.1.0, порожден от FreeNIBS-0.0.3-bf3 и появился на свет в результате попытки применить последний для биллинга dialin-пользователей (PPP/PPTP/PPPOE) в небольшом ИСП. Был почищен код, переписаны отдельные места, исправлены некоторые ошибки. Результат успешно применяется более полутора лет.

Официальная страница оригинального проекта FreeNIBS - <http://nibs.net.ua>

В настоящий момент нумерация версий изменена следующим образом:

вместо 0.1.x – используется нумерация 1.x.x

вместо 0.2.x – используется нумерация 2.x.x.

Также в разработке находится версия 3 модуля FreeNIBS.

Официальная веб-страница проекта <http://freesnibs.sourceforge.net>

Инсталляции, отличия версий и общий план разработки.

FreeNIBS является популярным продуктом на территории бывшего СССР и имеет, как минимум, несколько сотен инсталляций (по оценке автора документации). FreeNIBS широко применяется в малых Интернет-сервис-провайдерах и ethernet-сетях. Кроме того, FreeNIBS может применяться на

корпоративном уровне для организации управляемого доступа к отдельным сервисам (например, доступ в Интернет) и учета доступа.

По своим функциональным возможностям FreeNIBS превосходит множество коммерческих продуктов.

Недостатками FreeNIBS в сравнении с коммерческими продуктами являются слабость документации (до последнего времени), отсутствие коммерческой поддержки продукта и отсутствие линейки программных средств, в которую бы входил FreeNIBS, реализующих полноценный биллинг (до последнего времени).

В настоящий момент доступны 3 ветви FreeNIBS.

- Версии 1.x.x являются полностью совместимыми с оригинальным FreeNIBS (от Неона) и отличаются переработкой кода в некоторых местах, устраненными ошибками и новой (либо улучшенной) функциональностью, которую удалось внести не изменяя структуру данных. В настоящий момент разработка версии 1.x.x завершена. Новые релизы будут включать лишь исправления возможных ошибок и функционал из старших версий, который можно портировать без изменения структуры данных.
- Версии 2.x.x полностью включает в себя функционал версий 1.x.x и имеет незначительные изменения структуры данных для обеспечения нового. Основной упор в версии делался на реализацию биллинга пользователей по трафику с различной ценой, зависящей от объема последнего (Так называемый «биллинг с градациями трафика»). Кроме того, появились возможности задания различных прайс-листов для различных серверов доступа. В настоящий момент разработка версии 1.x.x завершена. Новые релизы будут включать лишь исправления возможных ошибок и функционал из старших версий, который можно портировать без изменения структуры данных.
- Версии 3.x.x полностью включает в себя функционал версий 2.x.x и имеет полностью переработанную структуру данных. Изменение структуры данных потребовалось как для реализации новых функциональных возможностей (авторизация по мак-адресу, АОНу и т.п., постпайд-биллинг), так и для нормальной совместной работы с модулем FreeVOIP.

Как помочь в разработке FreeNIBS?

Нам очень нужна Ваша помощь в разработке FreeNIBS. Помощь может иметь различные формы:

- Совместная разработка модуля.
- Компиляция и работа модуля на 64-х разрядных архитектурах.
- Тестирование и отладка.
- Разработка схем для различных СУБД.
- Портирование драйверов для СУБД.
- Написание скриптов и служебных утилит.
- Написание веб-интерфейсов.
- Написание документации, примеров конфигурирования различных серверов доступа.
- Концептуализация модуля. Разработка и планирование функционала модуля.
- Перевод документации на английский язык.
- Материальная помощь в денежной форме.

Коллектив разработчиков и пользователей FreeNIBS с благодарностью примет Вашу помощь в любой форме.

ФУНКЦИОНАЛЬНЫЕ ОСОБЕННОСТИ.

- Все пользователи и группы (пакеты), прайс-листы, сеансы, а также сведения об оказанных услугах хранятся в базе данных.
- Возможность использовать различные базы данных (MySQL, Postgresql, Oracle).
- Работа со счетами.
- Создание неограниченного числа групп различной конфигурации.
- Работа в режимах подсчета денег за трафик, время и трафик + время.
- Подсчет любого трафика (входящий, исходящий, максимальный, минимальный, превалирующий).
- Возможность работы в сетях VPN.
- Задание пропускной способности для групп/пользователей.
- Работа в режимах ограничений по времени и/или трафику.
- Ограничения на день, неделю, месяц и полное ограничение.
- Работа в режиме фиксированной суточной оплаты.
- Работа в режиме фиксированной месячной оплаты.
- Работа в смешанных режимах.
- Установка стоимости каждого часа соединения в любой день недели и в праздник.
- Установка стоимости трафика.
- Посекундная тарификация или поминутная тарификация.
- Праздничные дни.
- Активация с первого подключения на заданный период.
- Установка даты окончания действия логина.
- Установка даты начала действия логина.
- Возможность работы в кредит.
- Корпоративные счета (*).
- Блокировка логина.
- Выделение фиксированных IP адресов для каждого логина.
- Выделение номера для обратного звонка для каждого логина.
- Полная информация о подключениях (статистика).
- Предоплаченная система расчетов.
- Постпэйд система расчетов.
- Работа с карточками оплаты.
- Ограничение на время и день входа.
- Ограничения на доступ через определенный сервер доступа.
- Поддержка "живой" статистики (**).
- Отключение (сброс) пользователя при различных обстоятельствах (**).
- Любые методы проверки пароля (за счет других модулей).
- Индивидуальные прайс-листы для пользователей (начиная с версии 0.2.0)
- Разные прайс-листы для групп и пользователей по разным серверам доступа (начиная с версии 0.2.0).
- Тарификация трафика с градациями цены от объема потребления.
- Авторизация пользователя на АОНу.
- Авторизация пользователя по ip-адресу.
- Авторизация пользователя по mac-адресу.

* - реализуется дополнительными компонентами.

** - реализовано для некоторых аппаратных серверов доступа.

КАК РАБОТАЕТ FreeNIBS?

Авторизация и аутентификация.

Приходит пакет AUTH REQUEST. Radius запускает процесс authorize.

...

Вызывается модуль rlm_nibs.

Выполняется функция rlm_nibs_authorize.

Производится проверка на существование пользователя если найден, то происходит добавление специфических атрибутов (функция nibs_add_attrs):

- Добавление Framed-IP-Address, если нужно (groupss.framed_ip или users.framed_ip).
- Добавление Framed-IP-Netmask, если нужно (groups.framed_mask или users.framed_mask).
- Добавление Callback-Number, если нужно (users.callback_number).
- Добавление Idle-Timeout, если нужно (groups.idle_timeout).
- Добавление Simultaneous-Use, если нужно (groups.simultaneous_use).
- Добавление Session-Timeout, если нужно (groups.session_timeout или рассчитанное время).
- Установка лимита на превалирующий трафик за сессию в 4Гб (для предотвращения переполнения счетчиков трафика на сервере доступа).
- Добавление Port-Limit, если нужно (groups.port_limit).
- Добавление Huntgroup-Name, если нужно (groups.huntgroup_name).
- Добавление других атрибутов (groups.other_params).

Проверка на установку (Auth-Type == Nibs). Если нет - Auth-Type установится в Nibs (Auth-Type = Nibs). Если Auth-Type установлен в Nibs - модуль вернет OK.

Если пользователь не найден в базе:

- Если Auth-Type установлен в Nibs (Auth-Type == Nibs), то доступ запрещен - модуль вернет REJECT.
- Если Auth-Type не равен Nibs, то модуль вернет NOOP (продолжить проверку другими модулями)

...

Если процесс authorize вернул OK, то Radius запускает процесс authenticate. Если Auth-Type установлен в Nibs вызывается модуль rlm_nibs. Выполняется функция rlm_nibs_authenticate:

Производится проверка на существование пользователя, на вхождение пользователя в группу. Если пользователь найден и входит в какую-либо группу, то проверяется пароль, ip-адрес, мас-адрес или АОН (в зависимости от типа аутентификации) пользователя. Если не совпадает - доступ запрещен, и модуль вернет REJECT. Иначе модуль вернет OK

...

Если процесс authenticate вернул OK, то будет запущен процесс post-auth. Запускается фильтр nibs_filter – который производит множество проверок:

- Если пользователь использует счет (packets.tos > 0), то есть деньги на счету ((users.deposit + users.credit) > 0). Если нет - (error 1).

- Если пользователь не активирован (`users.activated == 0`), то если время добавления пользователя больше текущего (`users.add_date > now()`) - (error 31)
- Если время добавления пользователя равно нулю (`users.add_date == 0`) – то время добавления выставится в текущую дату (`users.add_date = now()`).
- Если время активации равно нулю (`packet.activation_time == 0`), время окончания действия логина устанавливается в ноль и игнорируется (`users.expired = 0`).
- Если время активации больше нуля (`packet.activation_time > 0`) и если дата для активации меньше даты истечения действия логина (`(now() + packets.activation_time) < users.expired`), то дата окончания заменяется новой датой (`users.expired = (now() + packets.activation_time)`).

Активация пользователя (`users.activated = 1`):

- Если дата окончания действия логина не равна нулю и если дата окончания действия логина меньше либо равно текущей дате (`users.expired <= now()`), то (error 2).
- Если пользователь заблокирован (`users.blocked = 1`), то (error 3).
- Если общее ограничение по времени больше нуля (`groups.total_time_limit > 0`) и если общая сумма времени пользователя больше либо равна лимиту (`users.total_time >= groups.total_time_limit`), то (error 11).
- Если месячное ограничение по времени больше нуля (`groups.month_time_limit > 0`), то производится расчет времени за текущий календарный месяц по таблице actions. Если рассчитанное время больше либо равняется лимиту (`%month_time% >= groups.month_time_limit`) - (error 12).
- Если недельное ограничение по времени больше нуля (`groups.week_time_limit > 0`), то производится расчет времени за текущую неделю по таблице actions. Если рассчитанное время больше, либо равняется лимиту (`%week_time% >= groups.week_time_limit`) - (error 13).
- Если суточное ограничение по времени больше нуля (`groups.day_time_limit > 0`), то производится расчет времени за текущие сутки по таблице actions. Если рассчитанное время больше, либо равняется лимиту (`%day_time% >= groups.day_time_limit`) - (error 14).
- Если общее ограничение по трафику больше нуля (`groups.total_traffic_limit > 0`) и если общая сумма трафика пользователя больше, либо равна лимиту (`users.total_traffic >= groups.total_traffic_limit`) - (error 21).
- Если месячное ограничение по трафику больше нуля (`groups.month_traffic_limit > 0`), то производится расчет времени за текущий календарный месяц по таблице actions. Если рассчитанный трафик больше, либо равняется лимиту (`%month_traffic% >= groups.month_traffic_limit`) - (error 22).
- Если недельное ограничение по трафику больше нуля (`groups.week_traffic_limit > 0`), то производится расчет времени за текущую неделю по таблице actions. Если рассчитанный трафик больше, либо равняется лимиту (`%week_traffic% >= groups.week_traffic_limit`) - (error 23).
- Если суточное ограничение по трафику больше нуля (`groups.day_traffic_limit > 0`), то производится расчет времени за текущие сутки по таблице actions. Если рассчитанный трафик больше, либо равняется лимиту (`%day_traffic% >= groups.day_traffic_limit`) - (error 24).
- Если установлено ограничение на время входа (`groups.login_time[0] != 0`) и если разрешенное время входа не равно текущему (`groups.login_time != now()`) - (error 40).

- Проверка на запрещенный телефонный номер. Если не прошла, то (error 50).
- Проверка на возможность входа с данного сервера доступа.
- Если возникла ошибка в ходе проверки, то (error -1).
- Если все в порядке, то (error 0).

Если фильтр вернул не 0, то вход запрещен - модуль вернет REJECT. Иначе модуль вернет OK.

Если процесс post-auth также вернет OK то пользователя впустит

***Примечание.** Лимиты установленные пользователю имеют приоритет перед лимитами, установленными группой. Лимиты установленные пользователю аналогично групповым лимитам используются в функции nibs_filter.*

Эккаунтинг.

Приходит пакет ACCT START. Radius запускает процесс accounting.

...

Вызывается модуль rlm_nibs. Выполняется функция rlm_nibs_accounting.

Производится проверка на существование пользователя:

- Если найден, то если у пакета отключена статистика (packet.no_acct == 1) - модуль вернет OK.
- В таблицу actions заносится запись о начале сессии. Модуль возвращает OK.
- Если пользователя нет в базе, то модуль вернет NOOP.

...

Приходит пакет ACCT ALIVE. Radius запускает процесс accounting.

...

Вызывается модуль rlm_nibs. Выполняется функция rlm_nibs_accounting.

Производится проверка на существование пользователя:

- Если найден, то если у пакета отключена статистика (packet.no_acct == 1) - модуль возвращает OK.
- Если для этого ALIVE пакета нет записи в таблице actions о начале сессии, то в таблицу actions добавляется запись о начале сессии.
- Выполняется функция nibs_billing - подсчет статистики и денег. Если сервер доступа не вернул Acct-Terminate-Cause, то:
- Acct-Terminate-Cause устанавливается в User-Request.
- В таблице users корректируются поля deposit, total_traffic и total_time (если это необходимо).
- В таблице actions корректируются поля in_bytes, out_bytes, time_on и framed_ip.
- Если (%kill_by_alive% == 1), то выполняется функция nibs_filter (работа аналогична описанной в AUTH части).
- Если (error != 0), запускается функция nibs_user_kill, которая вызовет %kill_program%.
- Модуль возвращает OK.
- Если пользователя нет в базе, то модуль возвращает NOOP.

Приходит пакет ACCT STOP. Radius запускает процесс accounting.

...

Вызывается модуль rlm_nibs. Выполняется функция rlm_nibs_accounting.

Производится проверка на существование пользователя если найден, то:

- Если у пакета отключена статистика (`packet.no_acct == 1`) - модуль возвращает ОК.
- Если для этого STOP пакета нет записи в таблице actions о начале сессии, то в таблицу actions добавляется запись о начале сессии.

Выполняется функция `nibs_billing` - подсчет статистики и денег.

- Если сервер доступа не вернул Acct-Terminate-Cause, то Acct-Terminate-Cause устанавливается в User-Request.
- В таблице users корректируются поля `deposit`, `total_traffic` и `total_time` (если это необходимо).
- В таблице actions запись о начале сессии заменяется записью о конце сессии. Модуль возвращает ОК.
- Если пользователя нет в базе, то модуль вернет NOOP.

УСТАНОВКА FreeNIBS.

Скачать архив последней версии FreeRadius

Скачать архив последней версии FreeNIBS

Распаковать архива с FreeRadius

```
# tar -zxf freeradius-X.Y.Z.tar.gz
```

Содержимое архива FreeNIBS распаковать в каталог с FreeRadius Все существующие файлы перезаписать.

Перейти в каталог `freeradius` и выполнить следующие команды

```
# cd freeradius-X.Y.Z
```

Внимание! Следующие 10 строк только для версий 0.1.0 и выше.

Перейти в каталог FreeNIBS.

```
# cd src/modules/rlm_nibs
```

Отредактировать файл `conf.h`. А именно, укажите версию FreeRADIUS-сервера в директиве

```
#ifndef RADIUS_VERSION
#define RADIUS_VERSION 1010
#endif
```

для версии радиус-сервера 1.0.1.

Вернитесь в корневой каталог радиуса

```
# cd ../../..
# ./configure --prefix=/usr/local/freeradius
# make
# make intsall
```

Если все прошло удачно, вы получите установленную версию FreeRadius + FreeNIBS в каталоге `/usr/local/freeradius`

Настроить радиус-сервер под свои нужды:

`/usr/local/freeradius/etc/raddb/clients.conf` - здесь нужно прописать все ваши сервера доступа (NAS) или IP машин, на которых будет запускаться `pppd_radius` и/или `portslave` и/или `pptpd`.

`/usr/local/freeradius/etc/raddb/huntgroups` – здесь можно вписать группы серверов доступа и порты, для ограничения доступа некоторым группам с определенных серверов доступа или портов.

/usr/local/freeradius/etc/raddb/nibs.conf. Установить по вкусу:

```
server = "localhost"
login = "root"
password = "*password_for_your_mysql*"
```

/usr/local/freeradius/etc/raddb/radiusd.conf. Просмотреть - может чего захочется поменять

Теперь пришла пора настроить базу данных

Пример для MySQL:

- перейти в каталог freeradius/src/modules/rlm_nibs/drivers/rlm_nibs_mysql
- отредактировать в файле nibs_mysql.sql строку с паролем
- дать команду `# mysql -uroot -p < nibs_mysql.sql`

Структура базы данных FreeNIBS (для версий 1 и 2).

Основная база системы состоит из 6 таблиц:

1. `users` - информация о пользователях (логин, пароль, депозит и т.д.)
2. `actions` - информация о подключениях (статистика)
3. `prices` - цены на время и трафик
4. `packets` - информация о пакетах (группах)
5. `holidays` - список праздничных дат
6. `blacklist` - список запрещенных и разрешенных телефонных номеров

Таблица `users`

`user` varchar(64) - Логин пользователя

- * Должен состоять из маленьких букв английского алфавита
- * Может содержать цифры, знаки '-' и '_'
- * Не должен начинаться с цифры

`passwd` varchar(254) - Пароль

`crypt_method` tinyint(1) - Тип шифрования пароля

- 0 - не зашифрованный пароль
- 1 - зашифрованный Crypt
- 2 - зашифрованный MD5
- 3 - зашифрованный SHA1

`uid` integer - Уникальный номер пользователя

`gid` integer - Номер группы(пакета) по умолчанию

`deposit` double - Счет

`credit` double - Кредит

Если Счет + Кредит ≤ 0 при packet.tos > 0 то доступ будет запрещен

`fio` varchar(128) - ФИО (*)

`phone` varchar(128) - Контактный телефон (*)

`address` varchar(128) - Адрес (*)

`prim` varchar(254) - Примечание (*)

`add_date` date(YYYY-MM-DD) - Время добавления/активации пользователя

- * Если при подключении пользователя поле равно 0000-00-00 то оно будет установлено в текущую дату.
- * Если значение поля больше текущей даты, то пользователя не впустит до наступления указанной даты

`blocked` tinyint(1) - Если равно 1 - доступ запрещен

`activated` tinyint(1) - Если равно 0, то при входе будет установлено в 1 и если packets.activation_time больше 0 то произойдет коррекция поля expired. expired = текущая дата + количество секунд в packets.activation_time.

`expired` date(YYYY-MM-DD) - Дата окончания действия логина.

* Если равно 0 - игнорируется

``total_time` int(10)` - Счетчик времени в секундах

* Используется для сравнения с полем `packets.total_time_limit` и, если становится больше последнего - доступ запрещается.

``total_traffic` bigint(15)` - Счетчик трафика в байтах

* Используется для сравнения с полем `packets.total_traffic_limit` и, если становится больше последнего - доступ запрещается

``total_money` double` - Счетчик потраченных денег

* Используется для сравнения с полем `packets.total_money_limit` и, если становится больше последнего - доступ запрещается

``last_connection` date(YYYY-MM-DD)` - Дата последнего подключения

* Используется для функционирования механизма снятия фиксированной суммы денег

``framed_ip` varchar(16)` - Выделенный IP

``framed_mask` varchar(16)` - Выделенная маска

``callback_number` varchar(64)` - Номер для обратного звонка, если разрешен в `packets.allow_callback`

* Если поле пустое и `packets.allow_callback` для этой группы установлен в 1, то пользователю будет выдано приглашение ввести любой номер

Таблица ``actions``

``user` varchar(64)` - Логин пользователя

``gid` integer` - Группа, в которой в данный момент пользователь

``id` varchar(32)` - Идентификационный номер соединения

``unique_id` varchar(64)` - Уникальный номер соединения

``time_on` int(12)` - Время соединения в секундах

``start_time` datetime(YYYY-MM-DD HH:MM:SS)` - Дата и время окончания соединения

``stop_time` datetime(YYYY-MM-DD HH:MM:SS)` - Дата и время начала соединения

``in_bytes` bigint(15)` - принятые байты

``out_bytes` bigint(15)` - переданные байты

``ip` varchar(15)` - IP пользователя

``server` varchar(15)` - IP сервера доступа

``client_ip` varchar(15)` - Реальный IP сервера доступа

``port` int(12)` - Порт на сервере доступа

``call_to` varchar(64)` - Телефонный номер, на который был произведен звонок

``call_from` varchar(64)` - Телефонный номер пользователя

``connect_info` varchar(254)` - Информация о соединении

``protocol` varchar(32)` - Используемый протокол

``terminate_cause` varchar(64)` - Причина разрыва соединения

``hour_in_bytes` bigint(15)` - принятые байты за последний (текущий) час

``hour_out_bytes` bigint(15)` - переданные байты за последний (текущий) час

``hour_traffic_money` double` - Деньги за трафик последнего (текущего) часа

``last_change` int(10)` - Дата (timestamp) последнего обновления записи

``before_billing` double` - Состояние счета на момент начала соединения

``billing_minus` double` - Количество снятых денег за соединение

Таблица ``packets``

``num` Integer` - Номер пакета (*)

``packet` varchar(128)` - Название пакета (*)

``prefix` varchar(4)` - Префикс пакета

``gid` integer` - Номер группы

``deposit` double` - Начальное значение счета (*)

``credit` double` - Начальное значение кредита (*)

``tos` tinyint(1)` - За что считать деньги:
0 - не считать

1 - за время
 2 - за трафик
 3 - за время + трафик
 `do_with_tos` tinyint(1) - Снимать деньги с депозита и проверять их наличие:
 0 - нет, только записать в таблицу статистики (используется для работы только с денежными лимитами)
 1 - да и записать в таблицу статистики
 `direction` tinyint(1) - Если работа с трафиком, то учитывать трафик:
 0 - не учитывать
 1 - исходящий
 2 - входящий
 3 - суммарный
 4 - больший (за все соединение)
 5 - меньший (за все соединение)
 `fixed` tinyint(1) - Фиксированную оплату снимать:
 0 - не снимать
 1 - раз в сутки, если было подключение
 2 - за каждые сутки, в т.ч. и за предыдущие
 3 - при каждом подключении
 `fixed_cost` double - количество денег для снятия
 `activated` tinyint(1) - Начальное значение activated (*)
 `activation_time` int(10) - Время активации в секундах
 `blocked` tinyint(1) - если равно 1, то весь пакет заблокирован
 `total_time_limit` int(10) - Полное ограничение на время (**)
 `month_time_limit` int(10) - Месячное ограничение на время (**)
 `week_time_limit` int(10) - Недельное ограничение на время (**)
 `day_time_limit` int(10) - Дневное ограничение на время (**)
 `total_traffic_limit` bigint(15) - Полное ограничение на трафик (**)
 `month_traffic_limit` bigint(15) - Месячное ограничение на трафик (**)
 `week_traffic_limit` bigint(15) - Недельное ограничение на трафик (**)
 `day_traffic_limit` bigint(15) - Дневное ограничение на трафик (**)
 `total_money_limit` double - Полное ограничение на деньги (**)
 `month_money_limit` double - Месячное ограничение на деньги (**)
 `week_money_limit` double - Недельное ограничение на деньги (**)
 `day_money_limit` double - Дневное ограничение на деньги (**)
 `login_time` varchar(254) - Строка специального формата, для определения разрешенных временных интервалов подключения
 `huntgroup_name` varchar(64) - название IPгруппы (etc/raddb/huntgroups)
 `simultaneous_use` smallint(5) - Количество одновременно подключаемых пользователей
 `port_limit` smallint(5) - Ограничение по портам на один логин
 `session_timeout` int(10) - Ограничение на время сессии
 `idle_timeout` int(10) - Максимальное время бездействия
 `allowed_prefixes` varchar(64) - Разрешенные префиксы для смены тарифа
 `framed_ip` varchar(16) - IP для всего пакета (рекомендуется со знаком + в конце IP например 192.168.100.0+)
 `framed_mask` varchar(16) - Сетевая маска для пакета (например 255.255.255.255)
 `no_pass` tinyint(1) - если равно 1, то пускать всех пользователей этого пакета без пароля
 `no_acct` tinyint(1) - если равно 1, то не вести статистику по пользователям этого пакета (деньги также учитываться не будут)
 `allow_callback` tinyint(1) - Разрешить обратный звонок для данного пакета
 `other_params` varchar(254) - Текстовое поле, для передаваемых серверу доступа значений
 формат: параметр = значение, параметр = значение
 `create_system_user` tinyint(1) - Создавать системного пользователя, при создании пользователя в базе (*)

Таблица `prices`

`gid` integer - Номер группы (должен совпадать с номером группы в таблице users)
`week_day` tinyint(1) - День недели:
0 - воскресенье
1 - понедельник
2 - вторник
3 - среда
4 - четверг
5 - пятница
6 - суббота
7 - праздник
`h0` double(8,6) - Цена за час при работе с 00:00:00 до 00:59:59
`input0` double(8,6) - Цена за входящий трафик при работе с 00:00:00 до 00:59:59
`output0` double(8,6) - Цена за исходящий трафик при работе с 00:00:00 до 00:59:59
`h1` double(8,6) - Цена за час при работе с 01:00:00 до 01:59:59
`input1` double(8,6) - Цена за входящий трафик при работе с 01:00:00 до 01:59:59
`output1` double(8,6) - Цена за исходящий трафик при работе с 01:00:00 до 01:59:59
`h2` double(8,6) - Цена за час при работе с 02:00:00 до 02:59:59
`input2` double(8,6) - Цена за входящий трафик при работе с 02:00:00 до 02:59:59
`output2` double(8,6) - Цена за исходящий трафик при работе с 02:00:00 до 02:59:59
`h3` double(8,6) - Цена за час при работе с 03:00:00 до 03:59:59
`input3` double(8,6) - Цена за входящий трафик при работе с 03:00:00 до 03:59:59
`output3` double(8,6) - Цена за исходящий трафик при работе с 03:00:00 до 03:59:59
`h4` double(8,6) - Цена за час при работе с 04:00:00 до 04:59:59
`input4` double(8,6) - Цена за входящий трафик при работе с 04:00:00 до 03:59:59
`output4` double(8,6) - Цена за исходящий трафик при работе с 04:00:00 до 04:59:59
`h5` double(8,6) - Цена за час при работе с 05:00:00 до 05:59:59
`input5` double(8,6) - Цена за входящий трафик при работе с 05:00:00 до 05:59:59
`output5` double(8,6) - Цена за исходящий трафик при работе с 05:00:00 до 05:59:59
`h6` double(8,6) - Цена за час при работе с 06:00:00 до 06:59:59
`input6` double(8,6) - Цена за входящий трафик при работе с 06:00:00 до 06:59:59
`output6` double(8,6) - Цена за исходящий трафик при работе с 06:00:00 до 06:59:59
`h7` double(8,6) - Цена за час при работе с 07:00:00 до 07:59:59
`input7` double(8,6) - Цена за входящий трафик при работе с 07:00:00 до 07:59:59
`output7` double(8,6) - Цена за исходящий трафик при работе с 07:00:00 до 07:59:59
`h8` double(8,6) - Цена за час при работе с 08:00:00 до 08:59:59
`input8` double(8,6) - Цена за входящий трафик при работе с 08:00:00 до 08:59:59
`output8` double(8,6) - Цена за исходящий трафик при работе с 08:00:00 до 08:59:59
`h9` double(8,6) - Цена за час при работе с 09:00:00 до 09:59:59
`input9` double(8,6) - Цена за входящий трафик при работе с 09:00:00 до 09:59:59
`output9` double(8,6) - Цена за исходящий трафик при работе с 09:00:00 до 09:59:59
`h10` double(8,6) - Цена за час при работе с 10:00:00 до 10:59:59
`input10` double(8,6) - Цена за входящий трафик при работе с 10:00:00 до 10:59:59
`output10` double(8,6) - Цена за исходящий трафик при работе с 10:00:00 до 10:59:59
`h11` double(8,6) - Цена за час при работе с 11:00:00 до 11:59:59
`input11` double(8,6) - Цена за входящий трафик при работе с 11:00:00 до 11:59:59
`output11` double(8,6) - Цена за исходящий трафик при работе с 11:00:00 до 11:59:59
`h12` double(8,6) - Цена за час при работе с 12:00:00 до 12:59:59
`input12` double(8,6) - Цена за входящий трафик при работе с 12:00:00 до 12:59:59
`output12` double(8,6) - Цена за исходящий трафик при работе с 12:00:00 до 12:59:59
`h13` double(8,6) - Цена за час при работе с 13:00:00 до 13:59:59
`input13` double(8,6) - Цена за входящий трафик при работе с 13:00:00 до 13:59:59
`output13` double(8,6) - Цена за исходящий трафик при работе с 13:00:00 до 13:59:59
`h14` double(8,6) - Цена за час при работе с 14:00:00 до 14:59:59
`input14` double(8,6) - Цена за входящий трафик при работе с 14:00:00 до 14:59:59
`output14` double(8,6) - Цена за исходящий трафик при работе с 14:00:00 до 14:59:59
`h15` double(8,6) - Цена за час при работе с 15:00:00 до 15:59:59
`input15` double(8,6) - Цена за входящий трафик при работе с 15:00:00 до 15:59:59

``output15` double(8,6)` - Цена за исходящий трафик при работе с 15:00:00 до 15:59:59
``h16` double(8,6)` - Цена за час при работе с 16:00:00 до 16:59:59
``input16` double(8,6)` - Цена за входящий трафик при работе с 16:00:00 до 16:59:59
``output16` double(8,6)` - Цена за исходящий трафик при работе с 16:00:00 до 16:59:59
``h17` double(8,6)` - Цена за час при работе с 17:00:00 до 17:59:59
``input17` double(8,6)` - Цена за входящий трафик при работе с 17:00:00 до 17:59:59
``output17` double(8,6)` - Цена за исходящий трафик при работе с 17:00:00 до 17:59:59
``h18` double(8,6)` - Цена за час при работе с 18:00:00 до 18:59:59
``input18` double(8,6)` - Цена за входящий трафик при работе с 18:00:00 до 18:59:59
``output18` double(8,6)` - Цена за исходящий трафик при работе с 18:00:00 до 18:59:59
``h19` double(8,6)` - Цена за час при работе с 19:00:00 до 19:59:59
``input19` double(8,6)` - Цена за входящий трафик при работе с 19:00:00 до 19:59:59
``output19` double(8,6)` - Цена за исходящий трафик при работе с 19:00:00 до 19:59:59
``h20` double(8,6)` - Цена за час при работе с 20:00:00 до 20:59:59
``input20` double(8,6)` - Цена за входящий трафик при работе с 20:00:00 до 20:59:59
``output20` double(8,6)` - Цена за исходящий трафик при работе с 20:00:00 до 20:59:59
``h21` double(8,6)` - Цена за час при работе с 21:00:00 до 21:59:59
``input21` double(8,6)` - Цена за входящий трафик при работе с 21:00:00 до 21:59:59
``output21` double(8,6)` - Цена за исходящий трафик при работе с 21:00:00 до 21:59:59
``h22` double(8,6)` - Цена за час при работе с 22:00:00 до 22:59:59
``input22` double(8,6)` - Цена за входящий трафик при работе с 22:00:00 до 22:59:59
``output22` double(8,6)` - Цена за исходящий трафик при работе с 22:00:00 до 22:59:59
``h23` double(8,6)` - Цена за час при работе с 23:00:00 до 23:59:59
``input23` double(8,6)` - Цена за входящий трафик при работе с 23:00:00 до 23:59:59
``output23` double(8,6)` - Цена за исходящий трафик при работе с 23:00:00 до 23:59:59

Таблица ``holidays``

``holiday_date` varchar(5)` - Дата праздника в формате ММ-DD
``comment` varchar(64)` - Описание праздника (*)

Таблица ``blacklist``

``id` int(10)` - ID записи (для обеспечения их упорядоченности) (не используется)
``uid` integer` - UID пользователя, к которому применяется правило может быть NULL
``gid` integer` - GID пользователя, к которому применяется правило может быть NULL
``phone` varchar(64)` - номер или маска номеров, к которым относится правило
``permit` char(1)` - если 'y', то правило описывает разрешенный (белый) номер(а), иначе - запрещенный (черный)
``comment` varchar(128)` - любой комментарий, описывающий правило (не используется)

(*) - Используется только WEB интерфейсом

(**) - Значение для каждого пользователя, не для всей группы

Если вам нужен режим, в котором можно тонко изменять каждый параметр для каждого пользователя, а также подключить некоторые дополнительные возможности – смотри ниже, иначе можно перейти к следующей главе.

На этом этапе установка и начальная настройка закончены, теперь пора наполнять систему пакетами и пользователями.

Структура базы данных FreeNIBS (для версии 3).

```
# nibs_mysql.schema
# STA.billing / FreeNIBS Version 3.x
# Schema release 4
```

Codepage: Windows-1251

#

```
CREATE DATABASE freenibs;
use freenibs;
```

```
GRANT USAGE ON freenibs.* to freenibs@localhost;
GRANT SELECT,UPDATE,INSERT,DELETE,CREATE,DROP,ALTER,INDEX ON freenibs.* to
freenibs@localhost;
SET PASSWORD FOR "freenibs"@"localhost"=PASSWORD("*** pass ***");
```

```
CREATE TABLE `users` (
  `user` varchar(64) binary NOT NULL default "", # Имя пользователя / логин
  `password` varchar(64) binary NOT NULL default "", # Пароль пользователя.
                                                    # Не используется, если auth_type отличен
от '0' или no_pass установлен в '1'
  `auth_info` varchar(64) binary NOT NULL default "", # АОН пользователя, если auth_type
= '1',
                                                    # MAC-адрес устройства, если auth_type =
'2',
                                                    # IP-адрес, если auth_type = '3'
  `auth_type` tinyint(1) unsigned NOT NULL default '0', # Способ авторизации пользователя
                                                    # 0 - пароль, 1 - АОН, 2 - MAC-адрес, 3 - IP-
адрес
  `uid` bigint(20) unsigned NOT NULL auto_increment, # Уникальный идентификатор (номер)
записи
  `master_uid` bigint(15) unsigned NOT NULL default '0',# Уникальный идентификатор (номер)
владельца записи
  `gid` bigint(15) unsigned NOT NULL default '1', # Уникальный идентификатор (номер)
группы
  `deposit` double(16,6) NOT NULL default '0.000000', # Сумма на счете в валюте биллинга
  `credit` double(16,6) NOT NULL default '0.000000', # Максимальная сумма кредита для
записи (в валюте биллинга)
  `add_date` datetime NOT NULL default 'now()', # Дата добавления записи
  `tos` tinyint(1) unsigned NULL, # за что считать деньги:
# 0 - не считать
# 1 - за время
# 2 - за трафик
# 3 - за время + трафик
# 4 - за трафик с градациями объема
  `do_with_tos` tinyint(1) unsigned NULL, # снимать деньги с депозита и проверять их
наличие:
# 0 - нет, только записать в таблицу
статистики (используется для работы только с денежными лимитами)
# 1 - да и записать в таблицу статистики
  `direction` tinyint(1) unsigned NULL, # учитывать трафик:
# 0 - не учитывать
# 1 - исходящий
# 2 - входящий
# 3 - суммарный
# 4 - больший (за все соединение)
# 5 - меньший (за все соединение)
  `fixed` tinyint(1) unsigned NULL, # фиксированную оплату:
# 0 - не снимать
# 1 - раз в сутки, если было подключение
# 2 - за каждые сутки, в т.ч. и за предыдущие
# 3 - при каждом подключении
```

```

`fixed_cost` double(16,6), # количество денег для снятия
`activation_time` bigint(15), # время активации (получения доступа) в
секундах (на какое время пользователь будет активирован(получит доступ))
# Ограничения для данного пользователя
на время:
`total_time_limit` bigint(15), # полное
`month_time_limit` bigint(15) NULL, # месячное
`week_time_limit` bigint(15) NULL, # недельное
`day_time_limit` bigint(15) NULL, # сутки
# Ограничения для данного пользователя
на трафик:
`total_traffic_limit` bigint(15) NULL, # полное
`month_traffic_limit` bigint(15) NULL, # месячное
`week_traffic_limit` bigint(15) NULL, # недельное
`day_traffic_limit` bigint(15) NULL, # сутки
# Ограничения для данного пользователя
на деньги:
`total_money_limit` double(16,6) NULL, # полное
`month_money_limit` double(16,6) NULL, # месячное
`week_money_limit` double(16,6) NULL, # недельное
`day_money_limit` double(16,6) NULL, # сутки
`login_time` varchar(254) NULL, # строка для определения разрешенных
временных интервалов подключения, формат:
# разделенные запятыми интервалы, состоящие из дня и времени (DDHHMM-NNMM) или
просто дня (DD).
# Вот поддерживаемые дни:
# Su - воскресенье
# Mo - понедельник
# Tu - вторник
# We - среда
# Th - четверг
# Fr - пятница
# Sa - суббота
# Wk - рабочие дни (= Mo,Tu,We,Th,Fr)
# Al - все дни
# Hl - праздники
#
# вот пример:
# Wk0000-0800,Sa,Su,Hl2200-1000
#
# Впускать с понедельника по пятницу только ночью с 0 и до 8 утра, суббота и воскресенье
круглосуточно, в праздники с 10 вечера и до 10 утра.
#
`huntgroup_name` varchar(64) NULL, # название huntgroup для данного
тарифа, группа должна быть записана в etc/raddb/huntgroups
`simultaneous_use` smallint(3) unsigned NULL, # количество одновременно
подключаемых пользователей на один логин
`port_limit` smallint(3) unsigned NULL, # ограничение по портам NAS на одного
пользователя
`session_timeout` bigint(15), # ограничение на время сессии
пользователя
`idle_timeout` bigint(15) NULL, # максимальное время бездействия
пользователя на линии
`no_pass` tinyint(1) unsigned NULL, # если равно 1, то пускать
пользователя без пароля
`no_acct` tinyint(1) unsigned NULL, # если равно 1, то не вести
статистику по пользователю (деньги также учитываться не будут)

```

```

`allow_callback` tinyint(1) unsigned NULL,      # разрешить callback для данного
пользователя
`other_params` varchar(254) NULL,                # текстовое поле, для передаваемых
NAS значений, формат: параметр = значение, параметр = значение
`allowed_servers` varchar(254) NULL,            # Список адресов разрешенных для
данного пользователя серверов доступа
`blocked` tinyint(1) unsigned NOT NULL default '1', # если равно '1', то пользователь
блокирован
`activated` tinyint(1) unsigned NOT NULL default '0', # начальное значение активации для
нового пользователя (будет ли пользователь активирован (получит доступ) сразу или нет)
# 0 - с первого подключения
# 1 - при добавлении
`expired` date NOT NULL default '0000-00-00',   # Дата истечения срока действия
логина
`total_time` bigint(15) NOT NULL default '0',    # Общее время соединений
`total_traffic` bigint(15) NOT NULL default '0', # Общий трафик
# Общая сумма снятых денег
`total_money` double(16,6) NOT NULL default '0.000000',
# Время последнего соединения
`last_connection` datetime NOT NULL default '0000-00-00 00:00:00',
`framed_ip` varchar(16) NOT NULL default "",     # IP адрес назначаемый
пользователям данного тарифа.
# Для выдачи группы адресов необходимо
писать знак '+' после IP адреса, например '192.168.0.100+' (пользователям будут выданы
адреса 192.168.0.100, 192.168.0.101, 192.168.0.102 и т.д.)
`framed_mask` varchar(15) NOT NULL default "",   # сетевая маска назначаемая
пользователям данного тарифа, например 255.255.255.252
`callback_number` varchar(64) NOT NULL default "", # Номер для колбэка
`shape` int(5) NULL,                             # Пропускная способность соединения в
килобитах
PRIMARY KEY (`uid`),
KEY `user` (`user`)
) TYPE=MyISAM;
# -----

```

```

CREATE TABLE `actions` (
`user` varchar(64) NOT NULL default "",          # Имя пользователя / логин
`uid` bigint(20) unsigned NOT NULL default '0', # Уникальный идентификатор (номер)
записи
`gid` bigint(15) unsigned NOT NULL default '0', # Уникальный идентификатор (номер)
группы
`id` varchar(64) NOT NULL default "",           # Идентификатор сессии
`unique_id` varchar(64) NOT NULL default "",     # Уникальный идентификатор
сессии
`time_on` bigint(15) unsigned NOT NULL default '0', # Время соединения
`start_time` datetime NOT NULL default '2005-01-01 00:00:00', # Начало соединения
`stop_time` datetime NOT NULL default '0000-00-00 00:00:00', # Окончание соединения
`in_bytes` bigint(15) unsigned NOT NULL default '0', # Байты, входящего к
пользователю трафика за время соединения
`out_bytes` bigint(15) unsigned NOT NULL default '0', # Байты, исходящие от
пользователя трафика за время соединения
`ip` int(10) unsigned NOT NULL default '0',
`server` int(10) unsigned NOT NULL default '0', # IP-адрес сервера доступа
`client_ip` int(10) unsigned NOT NULL default '0', # IP-адрес пользователя
`port` int(5) unsigned NOT NULL default '0',    # Номер порта на сервере,
занятый пользователем

```

```

`call_to` varchar(64) NOT NULL default "", # Набранный пользователем номер
(Номер сервера доступа)
`call_from` varchar(64) NOT NULL default "", # Номер, с которого
пользователь вышел на сервер доступа (АОН)
`protocol` varchar(12) default NULL, # Протокол
`terminate_cause` varchar(64) NOT NULL default "", # Причина окончания сеанса.
Для текущих - 'Online'
`hour_in_bytes` bigint(15) unsigned NOT NULL default '0', # Байты, входящие к
пользователю за текущий час соединения
`hour_out_bytes` bigint(15) unsigned NOT NULL default '0', # Байты, исходящие от
пользователя за текущий час соединения
`hour_traffic_money` double(16,6) NOT NULL default '0.000000',# Деньги начисленные за
трафик в течении текущего часа соединения
`last_change` int(10) unsigned NOT NULL default '0', # Дата и время последнего
изменения записи
`before_billing` double(16,6) NOT NULL default '0.000000', # Сумма на счете
пользователя перед началом сессии
`billing_minus` double(16,6) NOT NULL default '0.000000', # Сумма, снятая со счета
PRIMARY KEY (`unique_id`),
KEY `user` (`user`,`id`,`time_on`,`start_time`,`stop_time`)
) TYPE=MyISAM;
# -----

```

```

CREATE TABLE `d_groups` (
`gid` bigint(15) unsigned NOT NULL auto_increment, # Уникальный
идентификатор (номер) группы
`packet` varchar(128) NOT NULL default "", # Название группы
`deposit` double(16,6) NOT NULL default '0.000000', # Начальная сумма на
счете пользователей группы в валюте биллинга
`credit` double(16,6) NOT NULL default '0.000000', # Начальная сумма
кредита для всех пользователей из группы (в валюте биллинга)
`tos` tinyint(1) unsigned NOT NULL default '0', # за что считать деньги:
# 0 - не считать
# 1 - за время
# 2 - за трафик
# 3 - за время + трафик
# 4 - за трафик с градациями
объема
`do_with_tos` tinyint(1) unsigned NOT NULL default '1', # снимать деньги с депозита и
проверять их наличие:
# 0 - нет, только записать в
таблицу статистики (используется для работы только с денежными лимитами)
# 1 - да и записать в таблицу
статистики
`direction` tinyint(1) unsigned NOT NULL default '0', # учитывать трафик:
# 0 - не учитывать
# 1 - исходящий
# 2 - входящий
# 3 - суммарный
# 4 - больший (за все
соединение)
# 5 - меньший (за все
соединение)
`fixed` tinyint(1) unsigned NOT NULL default '0', # фиксированную оплату:
# 0 - не снимать

```

```

# 1 - раз в сутки, если было
подключение
# 2 - за каждые сутки, в т.ч. и
за предыдущие
# 3 - при каждом подключении
`fixed_cost` double(16,6) NOT NULL default '0.000000', # количество денег для снятия
`activated` tinyint(1) unsigned NOT NULL default '1', # начальное значение
активации для нового пользователя (будет ли пользователь активирован (получит доступ) сразу или нет)
# 0 - с первого подключения
# 1 - при добавлении
`activation_time` int(10) NOT NULL default '2678400', # время активации
(плучения доступа) в секундах (на какое время пользователь будет активирован(получит доступ))
`blocked` tinyint(1) unsigned NOT NULL default '0', # если равно '1', то весь
тариф блокирован
# Ограничения для каждого
пользователя данного тарифа на время:
# Полное
`total_time_limit` bigint(15) unsigned NOT NULL default '0', # На месяц
`month_time_limit` bigint(15) unsigned NOT NULL default '0', # На неделю
`week_time_limit` bigint(15) unsigned NOT NULL default '0', # на сутки
`day_time_limit` bigint(15) unsigned NOT NULL default '0', # Ограничения для каждого
пользователя данного тарифа на трафик:
# Полное
`total_traffic_limit` bigint(15) unsigned NOT NULL default '0', # На месяц
`month_traffic_limit` bigint(15) unsigned NOT NULL default '0', # На неделю
`week_traffic_limit` bigint(15) unsigned NOT NULL default '0', # на сутки
`day_traffic_limit` bigint(15) unsigned NOT NULL default '0', # Ограничения для каждого
пользователя данного тарифа на деньги:
# Полное
`total_money_limit` double(16,6) NOT NULL default '0.000000', # На месяц
`month_money_limit` double(16,6) NOT NULL default '0.000000', # На неделю
`week_money_limit` double(16,6) NOT NULL default '0.000000', # на сутки
`day_money_limit` double(16,6) NOT NULL default '0.000000', # строка для определения
`login_time` varchar(254) NOT NULL default "",
разрешенных временных интервалов подключения, формат:
# разделенные запятыми интервалы, состоящие из дня и времени (DDHHMM-NNMM) или
просто дня (DD).
# Вот поддерживаемые дни:
# Su - воскресенье
# Mo - понедельник
# Tu - вторник
# We - среда
# Th - четверг
# Fr - пятница
# Sa - суббота
# Wk - рабочие дни (= Mo,Tu,We,Th,Fr)
# Al - все дни
# Hl - праздники
#
# вот пример:
# Wk0000-0800,Sa,Su,Hl2200-1000
#
# Впускать с понедельника по пятницу только ночью с 0 и до 8 утра, суббота и воскресенье
круглосуточно, в праздники с 10 вечера и до 10 утра.
#

```

```

`huntgroup_name` varchar(64) NOT NULL default "", # название huntgroup
для данного тарифа, группа должна быть записана в etc/raddb/huntgroups
`simultaneous_use` smallint(3) unsigned NOT NULL default '0', # количество
одновременно подключаемых пользователей на один логин
`port_limit` smallint(3) unsigned NOT NULL default '0', # ограничение по портам NAS
на одного пользователя
`session_timeout` bigint(15) unsigned NOT NULL default '0', # ограничение на время
сессии пользователя
`idle_timeout` bigint(15) unsigned NOT NULL default '0', # максимальное время
бездействия пользователя на линии
`framed_ip` varchar(16) NOT NULL default "", # IP адрес назначаемый
пользователям данного тарифа.
# Для выдачи группы адресов
необходимо писать знак '+' после IP адреса, например '192.168.0.100+' (пользователям
будут выданы адреса 192.168.0.100, 192.168.0.101, 192.168.0.102 и т.д.)
`framed_mask` varchar(15) NOT NULL default "", # сетевая маска назначаемая
пользователям данного тарифа, например 255.255.255.252
`no_pass` tinyint(1) unsigned NOT NULL default '0', # если равно 1, то
пускать всех пользователей этого тарифа без пароля
`no_acct` tinyint(1) unsigned NOT NULL default '0', # если равно 1, то не
вести статистику по пользователям этого тарифа (деньги также учитываться не будут)
`allow_callback` tinyint(1) unsigned NOT NULL default '0', # разрешить callback
для данного тарифа
`other_params` varchar(254) NOT NULL default 'Acct-Interim-Interval = 60, Service-Type =
Framed-User, Framed-Protocol = PPP',
# текстовое поле, для
передаваемых NAS значений, формат: параметр = значение, параметр = значение
`allowed_servers` varchar(254) NULL, # Список адресов
разрешенных для данной группы серверов доступа
`shape` int(5) NOT NULL default '0', # Пропускная
способность соединения в килобитах
PRIMARY KEY (`gid`),
UNIQUE KEY `packet` (`packet`)
) TYPE=MyISAM;
# -----

```

```

CREATE TABLE `prices` (
`gid` bigint(15) unsigned NOT NULL default '1', # Номер группы, для которой предназначен
прайс-лист.
# Если gid == 0, то прайс-лист для
конкретного пользователя
`week_day` tinyint(1) unsigned NOT NULL default '0', # День недели. Для прайс-листов
биллинга трафика в зависимости от объема
# (type == '1') - неактуален. Должен быть 0
`h0` double(10,6) NOT NULL default '0.000000', # Если type == '1', то первый объем трафика,
иначе цена за 1 час соединения с 00.00 до 01.00
`input0` double(10,6) NOT NULL default '0.000000', # Если type == '1', то цена за 1
мегабайт входящего к пользователю трафика,
# иначе цена за 1 мегабайт входящего к
пользователю трафика с 00.00 по 01.00
`output0` double(10,6) NOT NULL default '0.000000', # Если type == '1', то цена за 1
мегабайт исходящего от пользователя трафика,
# иначе цена за 1 мегабайт исходящего от
пользователя трафика с 00.00 по 01.00
`h1` double(10,6) NOT NULL default '0.000000', # Если type == '1', то второй объем трафика,
иначе цена за 1 час соединения с 01.00 до 02.00

```

```
`input1` double(10,6) NOT NULL default '0.000000',      # Если type == '1', то цена за 1
мегабайт входящего к пользователю трафика,
# иначе цена за 1 мегабайт входящего к
пользователю трафика с 01.00 по 02.00
`output1` double(10,6) NOT NULL default '0.000000',    # Если type == '1', то цена за 1
мегабайт исходящего от пользователя трафика,
# иначе цена за 1 мегабайт исходящего от
пользователя трафика с 01.00 по 02.00
`h2` double(10,6) NOT NULL default '0.000000',
`input2` double(10,6) NOT NULL default '0.000000',
`output2` double(10,6) NOT NULL default '0.000000',
`h3` double(10,6) NOT NULL default '0.000000',
`input3` double(10,6) NOT NULL default '0.000000',
`output3` double(10,6) NOT NULL default '0.000000',
`h4` double(10,6) NOT NULL default '0.000000',
`input4` double(10,6) NOT NULL default '0.000000',
`output4` double(10,6) NOT NULL default '0.000000',
`h5` double(10,6) NOT NULL default '0.000000',
`input5` double(10,6) NOT NULL default '0.000000',
`output5` double(10,6) NOT NULL default '0.000000',
`h6` double(10,6) NOT NULL default '0.000000',
`input6` double(10,6) NOT NULL default '0.000000',
`output6` double(10,6) NOT NULL default '0.000000',
`h7` double(10,6) NOT NULL default '0.000000',
`input7` double(10,6) NOT NULL default '0.000000',
`output7` double(10,6) NOT NULL default '0.000000',
`h8` double(10,6) NOT NULL default '0.000000',
`input8` double(10,6) NOT NULL default '0.000000',
`output8` double(10,6) NOT NULL default '0.000000',
`h9` double(10,6) NOT NULL default '0.000000',
`input9` double(10,6) NOT NULL default '0.000000',
`output9` double(10,6) NOT NULL default '0.000000',
`h10` double(10,6) NOT NULL default '0.000000',
`input10` double(10,6) NOT NULL default '0.000000',
`output10` double(10,6) NOT NULL default '0.000000',
`h11` double(10,6) NOT NULL default '0.000000',
`input11` double(10,6) NOT NULL default '0.000000',
`output11` double(10,6) NOT NULL default '0.000000',
`h12` double(10,6) NOT NULL default '0.000000',
`input12` double(10,6) NOT NULL default '0.000000',
`output12` double(10,6) NOT NULL default '0.000000',
`h13` double(10,6) NOT NULL default '0.000000',
`input13` double(10,6) NOT NULL default '0.000000',
`output13` double(10,6) NOT NULL default '0.000000',
`h14` double(10,6) NOT NULL default '0.000000',
`input14` double(10,6) NOT NULL default '0.000000',
`output14` double(10,6) NOT NULL default '0.000000',
`h15` double(10,6) NOT NULL default '0.000000',
`input15` double(10,6) NOT NULL default '0.000000',
`output15` double(10,6) NOT NULL default '0.000000',
`h16` double(10,6) NOT NULL default '0.000000',
`input16` double(10,6) NOT NULL default '0.000000',
`output16` double(10,6) NOT NULL default '0.000000',
`h17` double(10,6) NOT NULL default '0.000000',
`input17` double(10,6) NOT NULL default '0.000000',
`output17` double(10,6) NOT NULL default '0.000000',
`h18` double(10,6) NOT NULL default '0.000000',
```

```

`input18` double(10,6) NOT NULL default '0.000000',
`output18` double(10,6) NOT NULL default '0.000000',
`h19` double(10,6) NOT NULL default '0.000000',
`input19` double(10,6) NOT NULL default '0.000000',
`output19` double(10,6) NOT NULL default '0.000000',
`h20` double(10,6) NOT NULL default '0.000000',
`input20` double(10,6) NOT NULL default '0.000000',
`output20` double(10,6) NOT NULL default '0.000000',
`h21` double(10,6) NOT NULL default '0.000000',
`input21` double(10,6) NOT NULL default '0.000000',
`output21` double(10,6) NOT NULL default '0.000000',
`h22` double(10,6) NOT NULL default '0.000000',
`input22` double(10,6) NOT NULL default '0.000000',
`output22` double(10,6) NOT NULL default '0.000000',
`h23` double(10,6) NOT NULL default '0.000000',
`input23` double(10,6) NOT NULL default '0.000000',
`output23` double(10,6) NOT NULL default '0.000000',
`servers` varchar(254) NULL, # Сервер(а), для которых действует
данный прайс-лист. # Если поле пустое или содержит '*' -
действует для всех серверов доступа. #
`type` tinyint(1) NOT NULL default '0', #
`uid` bigint(20) unsigned NULL, #
) TYPE=MyISAM;
# -----

CREATE TABLE `holidays` (
`holiday_date` date NOT NULL default '2005-01-01', # Дата праздника
`comment` varchar(64) NOT NULL default "", # Примечание
PRIMARY KEY (`holiday_date`)
) TYPE=MyISAM;
# -----

CREATE TABLE `blacklist` (
`id` int(10) unsigned NOT NULL auto_increment, # Уникальный номер записи
`uid` bigint(20) unsigned default NULL, # Номер пользователя, для которого
выполняется блокировка
`gid` bigint(15) unsigned default NULL, # Номер группы, для которой
осуществляется блокировка
`phone` varchar(64) NOT NULL default "", # Телефоны (маска), по которым проводится
блокировка доступа
`permit` tinyint(1) unsigned NOT NULL default '1', # Если permit == 1, то блокировка
отключается
`comment` varchar(128) default "", # Комментарий
PRIMARY KEY (`id`),
KEY `phone` (`phone`)
) TYPE=MyISAM;
# -----

#
# Правила записи адресов серверов доступа и телефонных номеров:
# Адреса (номера) могут записываться в строку через разделители (символы ',', табуляцию,
пробел)
# и могут содержать символы '?' и '*'.
#

```

```
# Правила сравнения номеров телефонов и адресов серверов доступа:
# Символ '?' соответствует любому символу в сравниваемой строке
# Символ '*' соответствует любым символам с текущей позиции и до конца сравниваемой строки
#
#
```

Содержимое файла NIBS.CONF

Как правило, содержимое файла NIBS.CONF (файла конфигурации модуля) требуется редактировать единственный раз – при инсталляции, да и редактировать только адрес сервера БД, имя пользователя БД, пароль и тип БД. Тем не менее ниже мы рассмотрим структуру этого файла.

NIBS.CONF.

```
nibs {

# Драйвер базы данных
# Поддерживаются: rlm_nibs_mysql, rlm_nibs_postgresql, rlm_nibs_oracle
driver = "rlm_nibs_mysql"

# Доступ в БД (сервер, порт, пользователь, пароль)
server = "localhost"
port = "3306"
login = "freenibs"
password = "*** pass ***"

# Имя БД
nibs_db = "freenibs"

# Имена используемых таблиц
auth_table = "users"
packets_table = "packets"
prices_table = "prices"
acct_table = "actions"
blacklist_table = "blacklist"
holidays_table = "holidays"

# Выводить все SQL-запросы в режиме отладки (-x)
sqltrace = no
sqltracefile = ${logdir}/nibs_mysqltrace.sql

# Количество сокетов для связи с сервером БД в пуле
num_socks = 64

# Время до повторного запроса при ошибке работы с сокетом
# Количество неудачных повторов, чтобы считать сокет «испорченным»
connect_failure_retry = 100
connect_failure_retry_delay = 3

#####
# Конфигурация запросов
#####
```

```
# Соответствующим образом обработанное имя пользователя – 'SQL-User-Name'.
# Подстановка '%{SQL-User-Name}' использована во всех запросах, где требуется
# имя пользователя. Для его получения Вы можете воспользоваться одним из правил:
#sql_user_name = "%{Stripped-User-Name:-%{User-Name:-none}}"
#
sql_user_name = "%{User-Name}"
```

```
#####
# Запрос для инициализации прайс-листов
#####
```

```
group_select_query = "SELECT gid, prefix, tos, do_with_tos, direction, fixed, fixed_cost,
activation_time, total_time_limit, month_time_limit, week_time_limit, day_time_limit,
total_traffic_limit, month_traffic_limit, week_traffic_limit, day_traffic_limit, total_money_limit,
month_money_limit, week_money_limit, day_money_limit, session_timeout, idle_timeout,
simultaneous_use, port_limit, login_time, other_params, huntgroup_name, allow_callback,
allowed_prefixes, framed_ip, framed_mask, no_acct, no_pass FROM ${packets_table}"
```

```
# Для активации проф-режима раскомментируйте следующую строку. Строка ниже –
только для версий 2.x.x
```

```
# group_select_query_prof = "SELECT allowed_servers, disabled_servers FROM
${packets_table} WHERE gid=%lu"
```

```
holyday_select_query = "SELECT holiday_date FROM ${holidays_table}"
```

```
prices_select_query = "SELECT h0, input0, output0, h1, input1, output1, h2, input2, output2, h3,
input3, output3, h4, input4, output4, h5, input5, output5, h6, input6, output6, h7, input7, output7,
h8, input8, output8, h9, input9, output9, h10, input10, output10, h11, input11, output11, h12,
input12, output12, h13, input13, output13, h14, input14, output14, h15, input15, output15, h16,
input16, output16, h17, input17, output17, h18, input18, output18, h19, input19, output19, h20,
input20, output20, h21, input21, output21, h22, input22, output22, h23, input23, output23,
week_day, gid FROM ${prices_table}"
```

```
# Если переменная "use_ext_price_attr" установлена в "yes" – используйте следующий
запрос. Только для версий 2.x.x
```

```
# prices_select_query = "SELECT h0, input0, output0, h1, input1, output1, h2, input2, output2,
h3, input3, output3, h4, input4, output4, h5, input5, output5, h6, input6, output6, h7, input7,
output7, h8, input8, output8, h9, input9, output9, h10, input10, output10, h11, input11, output11,
h12, input12, output12, h13, input13, output13, h14, input14, output14, h15, input15, output15,
h16, input16, output16, h17, input17, output17, h18, input18, output18, h19, input19, output19,
h20, input20, output20, h21, input21, output21, h22, input22, output22, h23, input23, output23,
week_day, gid, type, servers, uid FROM ${prices_table}"
```

```
blacklist_select_query = "SELECT uid, gid, phone, permit FROM ${blacklist_table} order by id"
```

```
#####
# Запрос для аутентификации
#####
```

```
authenticate_query = "SELECT user, password, gid, crypt_method FROM ${auth_table}
WHERE user = '%{SQL-User-Name}'"
```

```
#####
# Запрос для авторизации
#####
```

```
authorize_query = "SELECT user, password, crypt_method, uid, gid, deposit, credit,
unix_timestamp(add_date), blocked, activated, unix_timestamp(expired), total_time, total_traffic,
total_money, unix_timestamp(last_connection), framed_ip, framed_mask, callback_number FROM
${auth_table} WHERE user = '%{SQL-User-Name}'"
```

```
# Для включения проф.режима добавьте поля в nibs_mysql_prof_add.sql к Вашей БД
```

```
# и раскомментируйте следующую строку. Только для версий 1.x.x и 2.x.x
#authorize_query_prof = "SELECT tos, do_with_tos, direction, fixed, fixed_cost, activation_time,
total_time_limit, month_time_limit, week_time_limit, day_time_limit, total_traffic_limit,
month_traffic_limit, week_traffic_limit, day_traffic_limit, total_money_limit, month_money_limit,
week_money_limit, day_money_limit, login_time, huntgroup_name, simultaneous_use, port_limit,
session_timeout, idle_timeout, allowed_prefixes, no_pass, no_acct, allow_callback, other_params,
allowed_servers, disabled_servers FROM ${auth_table} WHERE user = '%{SQL-User-Name}'"
```

```
#####
# Фильтры
#####
```

```
month_time_query = "SELECT sum(time_on) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and start_time >= CONCAT(YEAR(NOW()),'-',MONTH(NOW()),'-01 00:00:00') AND start_time <= ADDDATE(CONCAT(YEAR(NOW()),'-',MONTH(NOW()),'-01 00:00:00'), INTERVAL 1 MONTH)"
```

```
week_time_query = "SELECT sum(time_on) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and YEAR(start_time) = YEAR(NOW()) AND WEEK(start_time,1) = WEEK(NOW(), 1)"
```

```
day_time_query = "SELECT sum(time_on) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and TO_DAYS(start_time) = TO_DAYS(NOW())"
```

```
month_traffic_query = "SELECT sum(in_bytes), sum(out_bytes) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and start_time >= CONCAT(YEAR(NOW()),'-',MONTH(NOW()),'-01 00:00:00') AND start_time <= ADDDATE(CONCAT(YEAR(NOW()),'-',MONTH(NOW()),'-01 00:00:00'), INTERVAL 1 MONTH)"
```

```
week_traffic_query = "SELECT sum(in_bytes), sum(out_bytes) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and YEAR(start_time) = YEAR(NOW()) AND WEEK(start_time,1) = WEEK(NOW(), 1)"
```

```
day_traffic_query = "SELECT sum(in_bytes), sum(out_bytes) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and TO_DAYS(start_time) = TO_DAYS(NOW())"
```

```
month_money_query = "SELECT sum(billing_minus) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and start_time >= CONCAT(YEAR(NOW()),'-',MONTH(NOW()),'-01 00:00:00') AND start_time <= ADDDATE(CONCAT(YEAR(NOW()),'-',MONTH(NOW()),'-01 00:00:00'), INTERVAL 1 MONTH)"
```

```
week_money_query = "SELECT sum(billing_minus) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and YEAR(start_time) = YEAR(NOW()) AND WEEK(start_time,1) = WEEK(NOW(), 1)"
```

```
day_money_query = "SELECT sum(billing_minus) FROM ${acct_table} WHERE user = '%{SQL-User-Name}' and TO_DAYS(start_time) = TO_DAYS(NOW())"
```

```
activation_update_query = "UPDATE ${auth_table} SET add_date = %%s, expired = %%s, activated = %%d, last_connection = from_unixtime(%l) WHERE user = '%{SQL-User-Name}'"
```

```
fixed_select_query = "SELECT to_days('%S'), to_days(last_connection) FROM ${auth_table} WHERE user = '%{SQL-User-Name}'"
```

```
#####
# Запросы для эккаунтинга
#####
```

```
# accounting_onoff_query - Запрос выполняется, когда приходит On или Off-пакет. Не реализовано в версиях 1.x.x и 2.x.x
```

```
# accounting_update_query - Обновление эккаунтинговой информации
```

```
# accounting_start_query - Внесение записи о начале сессии
```

```
# accounting_start_query_alt - Альтернативное внесение записи о начале сессии
```

```
# (в случае, если предыдущий запрос будет неудачным)
```

```
# accounting_stop_query - Внесение записи об окончании сессии
```

```
# accounting_stop_query_alt - Альтернативный запрос об окончании сессии
```

```
# ( в случае, если предыдущий запрос будет неудачным )  
#####
```

```
statistic_select_query = "SELECT time_on, in_bytes, out_bytes, unix_timestamp(start_time),  
billing_minus, hour_in_bytes, hour_out_bytes, hour_traffic_money, last_change FROM  
${acct_table} WHERE id = '%{Acct-Session-Id}' AND unique_id = '%{Acct-Unique-Session-Id}'  
AND server = '%{NAS-IP-Address}' AND port = %{NAS-Port} AND user = '%{SQL-User-Name}'  
AND terminate_cause='Online'"
```

```
accounting_start_query = "INSERT INTO ${acct_table} (gid, id, unique_id, user, server, port,  
client_ip, protocol, ip, call_to, call_from, terminate_cause, connect_info, start_time, last_change,  
before_billing) VALUES (%%d, '%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}', '%{SQL-User-  
Name}', '%{NAS-IP-Address}', %{NAS-Port}, '%{Client-IP-Address}', '%{Framed-Protocol}',  
'%{Framed-IP-Address}', '%{Called-Station-Id}', '%{Calling-Station-Id}', 'Online', '%{Connect-Info}',  
'%S', %l, %%16.6f)"
```

```
accounting_start_query_alt = "INSERT INTO ${acct_table} (gid, id, unique_id, user, server, port,  
client_ip, protocol, ip, call_to, call_from, terminate_cause, connect_info, start_time, last_change,  
before_billing) VALUES (%%d, '%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}', '%{SQL-User-  
Name}', '%{NAS-IP-Address}', %{NAS-Port}, '%{Client-IP-Address}', '%{Framed-Protocol}',  
'%{Framed-IP-Address}', '%{Called-Station-Id}', '%{Calling-Station-Id}', 'Online', '%{Connect-Info}',  
from_unixtime(%l - %%d), %l, %%16.6f)"
```

```
accounting_reopen = "UPDATE ${acct_table} SET terminate_cause = 'Online', stop_time = 0, id  
= '%{Acct-Session-Id}', unique_id = '%{Acct-Unique-Session-Id}' WHERE id LIKE '%%%{Acct-  
Session-Id}' AND unique_id LIKE '%%%{Acct-Unique-Session-Id}' AND server = '%{NAS-IP-  
Address}' AND port = %{NAS-Port} AND user = '%{SQL-User-Name}' AND terminate_cause !=  
'Online' AND (-10 <= (unix_timestamp(start_time) - (%l - %{Acct-Session-Time}))) AND  
((unix_timestamp(start_time) - (%l - %{Acct-Session-Time})) <= 10)"
```

```
accounting_alive_auth_query = "UPDATE ${auth_table} SET deposit = deposit - %%16.6f,  
total_traffic = total_traffic + %%ld, total_time = total_time + %%ld, total_money = total_money +  
%%16.6f, last_connection = from_unixtime(%l) WHERE user = '%{SQL-User-Name}'"
```

```
accounting_stop_auth_query = "UPDATE ${auth_table} SET deposit = deposit - %%16.6f,  
total_traffic = total_traffic + %%ld, total_time = total_time + %%ld, total_money = total_money +  
%%16.6f, last_connection = from_unixtime(%l) WHERE user = '%{SQL-User-Name}'"
```

```
accounting_update_query = "UPDATE ${acct_table} SET time_on = if(%{Acct-Session-  
Time},%{Acct-Session-Time},time_on), in_bytes = if(%{Acct-Input-Octets},%{Acct-Input-  
Octets},in_bytes), out_bytes = if(%{Acct-Output-Octets},%{Acct-Output-Octets},out_bytes),  
last_change = %l, billing_minus = billing_minus + %%16.6f, ip = '%{Framed-IP-Address}',  
hour_in_bytes = %%lu, hour_out_bytes = %%lu, hour_traffic_money = %%16.6f WHERE  
id='%{Acct-Session-Id}' AND unique_id = '%{Acct-Unique-Session-Id}' AND server = '%{NAS-IP-  
Address}' AND port = %{NAS-Port} AND user = '%{SQL-User-Name}' AND terminate_cause =  
'Online'"
```

```
accounting_stop_query = "UPDATE ${acct_table} SET time_on = if(%{Acct-Session-  
Time},%{Acct-Session-Time},time_on), stop_time = '%S', in_bytes = if(%{Acct-Input-  
Octets},%{Acct-Input-Octets},in_bytes), out_bytes = if(%{Acct-Output-Octets},%{Acct-Output-  
Octets},out_bytes), billing_minus = billing_minus + %%16.6f, terminate_cause='%{Acct-Terminate-  
Cause}', connect_info = '%{Connect-Info}', id = concat(unix_timestamp(start_time),'-%{Acct-  
Session-Id}'), unique_id = concat(unix_timestamp(start_time),'-%{Acct-Unique-Session-Id}'), ip =  
'%{Framed-IP-Address}' WHERE id='%{Acct-Session-Id}' AND unique_id = '%{Acct-Unique-  
Session-Id}' AND server = '%{NAS-IP-Address}' AND port = %{NAS-Port} AND user = '%{SQL-  
User-Name}' AND terminate_cause = 'Online'"
```

```
#####  
# Запросы проверки одновременных сессий  
#####
```

```

# simul_count_query      - Запрос количества одновременных сессий. Возвращает
информацию о сессиях для дальнейшей проверки.
# simul_verify_query     - Запрос проверки на «забытые» сессии.
#                          - Сделайте пустым или закомментируйте для запрета проверки
#####

# Закомментируйте строку для запрета проверки одновременных сессий
simul_count_query = "SELECT id, user, server, port, ip, call_from FROM ${acct_table} WHERE
user = '%{SQL-User-Name}' AND terminate_cause = 'Online'"
simul_verify_query = "SELECT id, user, server, port, ip, protocol, last_change FROM
${acct_table} WHERE terminate_cause = 'Online' ORDER BY user"

# Удалять «забытые» сессии если checkrad не нашел дополнительных сессий.
deletestalesessions_by_checkrad = yes
deletestalesessions_by_alive = yes
alive_max_interval = 30 # Установить интервал ожидания alive-пакетов. Если в течении
данного промежутка времени не пришел alive-пакет, сессия считается «забытой». Имеет
смысл установить в большее значение.

set_session_timeout = no # Вычислять и передавать серверу доступа информацию о
максимальном времени сессии или нет.
use_black_list = yes # Использовать механизм «белых/черных» списков.
use_ext_price_attr = no # Использовать расширенные атрибуты прайс-листов. Только для
проф-режима. Только для версий 2.x.x
accounting_on_alive = yes # Тарификация по приходу alive-пакетов. Иначе только по start-
stop.
ignore_localhost_acct = yes # Игнорировать пакеты с localhost. В версиях 3.x.x
используется переменная ignore_localhost_pkt
kill_by_alive = yes # Завершать сессии по приходу alive-пакетов (при истечении условий
для предоставления услуги. Работает не со всеми серверами доступа.
kill_program = "${sbindir}/userkill %{User-Name} %{NAS-IP-Address} %{Client-IP-Address}
%{NAS-Port} %{Framed-IP-Address}" # ВрAPPER для завершения сеансов пользователей.
allowed_service = "Framed-User" # Вид сервиса, на пакеты которого реагирует FreeNIBS.
}

```

НАСТРОЙКА ОТДЕЛЬНЫХ ПАРАМЕТРОВ.

Настройка профессионального режима (до версии 3).

Для того, чтобы сделать возможным персональную настройку пользователей, надо выполнить следующие шаги:

Изменить таблицу users, добавив в нее следующие поля:

```

tos
do_with_tos
direction
fixed
fixed_cost
activation_time
total_time_limit
month_time_limit
week_time_limit

```

day_time_limit
total_traffic_limit
month_traffic_limit
week_traffic_limit
day_traffic_limit
total_money_limit
month_money_limit
week_money_limit
day_money_limit
login_time
huntgroup_name
simultaneous_use
port_limit
session_timeout
idle_timeout
allowed_prefixes
no_pass
no_acct
allow_callback
other_params

Начиная с версии 0.2.0 в таблицу пользователей добавлены два поля:

- allowed_servers – Список разрешенных для пользователя серверов доступа;
- disabled_servers – Список запрещенных для пользователя серверов доступа.

Начиная с версии 0.2.0 в таблицу групп добавлены два поля:

- allowed_servers – Список разрешенных для пользователя серверов доступа;
- disabled_servers – Список запрещенных для пользователя серверов доступа.

Начиная с версии 0.2.0 в таблицу прайс-листов добавлены три поля:

- gid – ID пользователя, для которого предназначен прайс-лист;
- type – тип прайс-листа;
- servers – список серверов, для которых применяется данный прайс-лист.

См. rlm_nibs/drivers/rlm_nibs_=/driver=/sql_=/driver=_prof_add.sql

Для MySQL выполнить команду:

```
mysql -uroot -p < sql_mysql_prof_add.sql
```

Раскомментировать в nibs.conf параметр authorize_query_prof

Перезапустить freeradius

Примечание. Начиная с версии 3.0.0 профессиональный режим является единственным режимом работы FreeNIBS.

Зависшие сессии.

В файле конфигурации nibs.conf присутствуют два параметра:

```
deletestalesessions_by_checkrad = yes  
deletestalesessions_by_alive = yes
```

Первый параметр разрешает проверять двойное подключение и определять "зависшие"

сессии методом запуска скрипта checkrad и подходит только для аппаратных

серверов доступа (NAS). Требуется верная настройка clients.conf особенно параметров:

nastype = тип сервера
login = логин для подключения
password = пароль

Ниже приведен список поддерживаемых серверов, взятый из оригинальной документации к FreeRadius

Тип ====	Производ. =====	Метод доступа =====	Требуется наличия =====	Нужен пароль =====
ascend	Lucent	SNMP	SNMP	Нет
bay	Nortel	finger	finger command	Нет
cisco	Cisco	SNMP	SNMP	Опционально [1]
computone	Computone	finger	finger command	Нет
cvx	Nortel	SNMP	SNMP	Нет
digitro	Digitro	rusers	rusers command	Нет
livingston	Livingston	SNMP	SNMP	Нет [2]
max40xx	Lucent	finger	finger command	Нет
netserver	USR/3com	telnet	CPAN Net::Telnet	Да
pathras	Cyclades	telnet	CPAN Net::Telnet	Да
patton	Patton	SNMP	SNMP	Нет
portslave	?	finger	finger command	Нет
pr3000	Cyclades	SNMP	snmpwalk command	Нет
pr4000	Cyclades	SNMP	snmpwalk command	Нет
tc	USR/3com	telnet	CPAN Net::Telnet	Да
usrhyper	USR/3com	SNMP	SNMP	Нет [3]
versanet	VersaNet	SNMP	SNMP	Нет

[1] В файле конфигурации установить login = SNMP, password = требуемая community.

[2] Needs at least ComOS 3.5, SNMP enabled.

[3] Set "Reported Port Density" to 256 (default)

Второй параметр разрешает проверять двойное подключение и определять "зависшие" сессии методом подсчета времени прихода следующего ALIVE пакета. Подходит для любых типов NAS, но требует верно сконфигурированный механизм работы ALIVE.

Работает следующим образом: из таблицы actions берется значение last_change, а из users.other_params - значение Acct-Interim-Interval. Затем по формуле ((now() - last_change) >= (Acct-Interim-Interval + alive_max_interval)) вычисляется "повисшая" сессия. Т.е. если в течении alive_max_interval секунд не пришел ожидаемый ALIVE - сессия считается "зависшей".

Примечание: сессия также считается зависшей, если подключение происходит с порта и сервера доступа, которые уже присутствуют в таблице actions с флагом Online. Такие сессии закрываются автоматически.

Описание функционирования системы черных/белых списков.

Сравнение телефонного номера, пришедшего с оборудования с номером из базы идет слева направо, т.е. номера, полученные от АТС, должны быть одной длины. В маске символ '?' совпадает с любым символом телефона, символ '*' совпадает со всеми оставшимися до конца номера телефона символами.

Например, '8815245*' совпадает со всеми номерами Мурманска, начинающимися на 45 (подразумевается, что АТС отдает номера с междугородним префиксом 8. Это может отличаться в зависимости от настроек Вашей аппаратуры и АТС Вашего телефонного провайдера);

'8815245432?' - совпадает с номерами Мурманска, начинающимися на 45432, и заканчивающимися на любую цифру.

ОБРАТИТЕ ВНИМАНИЕ, что запись '88152*45' фактически эквивалентна записи '88152*', т.к. '*' совпадает со всеми оставшимися до конца номера телефона символами (см. выше).

Проверка привязки к группе/пользователю осуществляется по следующему правилу:

- Если uid==NULL и gid==NULL, то правило относится ко ВСЕМ пользователям; проверяется в первую очередь.
- Если uid==NULL и gid==<число>, то правило относится к пользователям из группы с gid=<число>; проверяется во вторую очередь и является более приоритетным, чем первая проверка, и может изменять результат первой проверки.
- Если uid=<число> и gid==NULL, то правило относится к пользователю с uid=<число>; проверяется в последнюю очередь и является самой приоритетной проверкой, может изменять результат первых двух.
- Если uid=<число> и gid=<число>, то действие НЕ ОПРЕДЕЛЕНО.

Если при одной из последних двух описанных проверок будет найдено соответствие с группой или пользователем, и permit будет равно 'y', то, по-умолчанию, доступ будет ЗАПРЕЩЕН. Т.е. если для группы или пользователя есть хоть один разрешенный номер - пускать будет только с него, если нет других правил.

Проверки делаются до первого совпадения в каждом из описанных трех случаев.

Пример. Имеются записи:

Uid	Gid	Phone	Permit
500	NULL	8152451000	Y
NULL	NULL	8152450000	N
100	NULL	8152541234	Y
NULL	1000	815254*	N

1. позволит пользователю с uid=500 соединиться только с номера 8152451000
2. запретит _всем_ пользователям соединиться с номера 8152450000.

Предположим теперь, что есть пользователь с uid=100 и gid=1000.

3. разрешит ему соединиться с номера 8152541234, хотя!
4. доступ для его группы с этого телефона запрещен.

ОБНОВЛЕНИЕ FreeNIBS.

- Скопировать все файлы из архива новой версии поверх собранного в прошлый раз FreeRadius + NiBS.
- Перейти в каталог src/modules/rlm_nibs.

Внимание! Следующие 6 строк только для версий 0.1.0 и выше.

- Отредактировать файл conf.h. А именно, укажите версию FreeRADIUS-сервера в директиве

```
#ifndef RADIUS_VERSION
#define RADIUS_VERSION 1010
#endif
```

для версии радиус-сервера 1.0.1.

- Дать команды:
make clean
configure
make
make install

проследить, чтобы все прошло без ошибок

- Сохранить ваш старый `/$PATH_TO_INSTALLED_RADIUS$/etc/raddb/nibs.conf`.
- Переписать файл `raddb/nibs.conf` в `/$PATH_TO_INSTALLED_RADIUS$/etc/raddb/` и подправить его под свои нужды.
- Если в CHANGELOG сказано, что были изменения в `radiusd.conf` то перейти в каталог скомпилированного FreeRadius и сделать заново `./configure --prefix=...`

потом скопировать файл `raddb/radiusd.conf` в `/$PATH_TO_INSTALLED_RADIUS$/etc/raddb/`.

- Если в CHANGELOG сказано, что произошли изменения базы, то сверьте свою базу с файлом, идущем в новой версии.
- Перезапустить `radiusd`.

Приложение 1. Сокращения в SQL-запросах.

Значения сокращений, применяемых в SQL запросах в `nibs.conf`

Название	Описание	Эквивалент
%a	Протокол (SLIP/PPP)	{Framed-Protocol}
%c	Callback-Number	{Callback-Number}
%d	День запроса (DD)	
%f	Framed IP address	{Framed-IP-Address}
%i	Calling Station ID	{Calling-Station-Id}
%l	timestamp запроса	
%m	Месяц запроса (MM)	
%n	NAS IP address	{NAS-IP-Address}
%p	Номер порта	{NAS-Port-Id} {NAS-Port}
%s	Speed (PW_CONNECT_INFO)	{Connect-Info}
%t	request in ctime format	
%u	User name	{User-Name}
%A	radacct_dir	{config:radacctdir}
%C	clientname	
%D	request date (YYYYMMDD)	

%H	request hour	
%L	radlog_dir	%{config:logdir}
%M	MTU	%{Framed-MTU}
%R	radius_dir	%{config:raddbdir}
%S	request timestamp in SQL format	
%T	request timestamp in database format	
%U	Stripped User name	%{Stripped-User-Name}
%V	Request-Authenticator (Verified/None)	
%Y	request year (YYYY)	
%Z	All request attributes except password (must have a big buffer)	

Приложение 2. Конфигурация DialUp-сервера на базе CISCO.

Пример конфигурации рутера Cisco в качестве DialUp-сервера, работающего с FreeRADIUS+FreeNIBS.

192.168.101.1 - Default Gateway, DNS, Radius Server

```

!
version 12.3
service config
!
! логи должны быть от настоящего времени, а не из прошлого века
service timestamps debug datetime localtime
service timestamps log datetime localtime
!
! имя роутера, не DNS, а то что Вы увидите в консоли
hostname Router
!
aaa new-model
aaa authentication login default local-case
aaa authentication ppp default group radius
aaa authorization exec default local
aaa authorization network default group radius
aaa accounting delay-start
!aaa accounting update periodic 1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! данную строчку стоит добавить ЕСЛИ Ваш роутер не шлет Вам аливы
! согласно времени указанного Вами в
! Acct-Interim-Interval = 60
aaa accounting network default start-stop group radius
!
! root данной циски
username admin privilege 15 password admin
!
! временная зона Вашего роутера
clock timezone MSK 3
clock summer-time MSD recurring last Sun Mar 2:00 last Sun Oct 2:00
!

```

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.0
 !
interface Ethernet0
 ip address 192.168.101.2 255.255.255.224
 no cdp enable
 !
interface Serial0
 no ip address
 shutdown
 !
interface Serial1
 no ip address
 shutdown
 !
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool default
 no keepalive
 no fair-queue
 ppp authentication pap callin
 group-range 1 16
 !
! пул адресов для юзеров
 ip local pool default 10.0.0.2 10.0.0.18
 !
! дефолтный роут
 ip route 0.0.0.0 0.0.0.0 192.168.101.1
 !
! тут мы указываем от имени какого интерфейса и соответственно его адреса
! будет ходить наша циска к RADIUS серверу за инфой
 ip radius source-interface Ethernet0
 !
! собсно сам RADIUS сервер
! "RADIUS_secret" заменить на свой пароль!
 radius-server host 192.168.101.1 auth-port 1812 acct-port 1813 key RADIUS_secret
 !
line con 0
 exec-timeout 0 0
line 1 16
 modem Dialin
 autoselect ppp
line aux 0
 !
end
```

Приложение 3. Установка MPD+FreeNIBS под FreeBSD.

Настройка mpd+freeradius+freenibs+mysql

Необходимость назрела как всегда, домашняя сеть и трафик считать нечем. Набрёл в интернете на сайт www.nibs.net.ua, решение понравилось, взялся за его реализацию.

```
cd /usr/ports/net/mpd
make
make install
```

версия mpd у меня вот такая:

```
su-2.05# mpd -v
Version 3.18 (root@solar.oln.ru 07:41 28-Mar-2005)
```

Версия системы:

```
su-2.05# uname -a
FreeBSD solar.oln.ru 4.11-RELEASE FreeBSD 4.11-RELEASE #0: Sun Mar 27
07:42:36 NOVST 2005 root@:/usr/src/sys/compile/PROXY_SAT i386
```

Ставим freeradius и freenibs следующим образом.

Разархивируем radius к примеру в /tmp/billing,

```
tar -zxvf freeradius-0.9.3.tar.gz /tmp/billing
```

а потом туда же с перезаписью разархивируем freenibs

```
tar -zxvf freenibs-1.1.1.tar.gz /tmp/billing
cd /tmp/billing
./configure
make
make install
```

Файл конфигурации MPD

```
bash-2.05# more mpd.conf
```

```
default:
  load ptp0
  load ptp1
  load ptp2
  load ptp3
  load ptp4
  load ptp5
```

load pptp6

pptp0:

new -i ng0 pptp0 pptp0
set ipcp ranges 10.10.10.1/30 10.10.10.47/30
load pptp_standart

pptp1:

new -i ng1 pptp1 pptp1
set ipcp ranges 10.10.10.1/30 10.10.10.20/30
load pptp_standart

pptp2:

new -i ng2 pptp2 pptp2
set ipcp ranges 10.10.10.1/30 10.10.10.74/30
load pptp_standart

pptp3:

new -i ng3 pptp3 pptp3
set ipcp ranges 10.10.10.1/30 10.10.10.223/30
load pptp_standart

pptp4:

new -i ng4 pptp4 pptp4
set ipcp ranges 10.10.10.1/30 10.10.10.10/30
load pptp_standart

pptp5:

new -i ng5 pptp5 pptp5
set ipcp ranges 10.10.10.1/30 10.10.10.175/30
load pptp_standart

pptp6:

new -i ng6 pptp6 pptp6
set ipcp ranges 10.10.10.1/30 10.10.10.50/30
load pptp_standart

pptp_standart:

set iface disable on-demand
set iface enable tcpmssfix
set bundle enable multilink
set link yes acfcomp protocomp
set link no pap chap
set link enable chap
set link keep-alive 60 180
set ipcp yes vjcomp
set ipcp dns 192.168.2.1
set ipcp nbns 192.168.2.1
set iface enable proxy-arp
set bundle enable compression
set ccp yes mppc
set ccp yes mpp-e40
set ccp yes mpp-e56
set ccp yes mpp-e128
set ccp yes mpp-stateless
set bundle yes crypt-reqd
set pptp self 0.0.0.0
set pptp enable incoming
set pptp disable originate

```
set radius server 192.168.2.1 <пароль к радиусу> 1812 1813
set radius timeout 10
set radius config /usr/local/etc/raddb/radius.conf
set radius retries 3
set bundle enable radius-acct
set bundle enable radius-auth
set ipcp yes radius-ip
open
```

Необходимо заметить, что mpd не любит пробелов в конфиге, поэтому следует использовать табуляцию (Мнение автора примера).

```
bash-2.05# more mpd.links
```

```
pptp0:
  set link type pptp
```

```
pptp1:
  set link type pptp
```

```
pptp2:
  set link type pptp
```

```
pptp3:
  set link type pptp
```

```
pptp4:
  set link type pptp
```

```
pptp5:
  set link type pptp
```

```
pptp6:
  set link type pptp
```

Файл mpd.secret пустой, потому как пароли проверяет радиус.

```
bash-2.05# more acct_users
```

```
DEFAULT
  Service-Type == Framed-User,
  Service-Type == Login-User,
  Login-Service == Telnet,
  Login-Service == Rlogin,
  Login-Service == TCP-Clear,
  Login-TCP-Port <= 65536,
  Framed-IP-Address == 255.255.255.254,
  Framed-IP-Netmask == 255.255.255.255,
  Framed-Protocol == PPP,
  Framed-Protocol == SLIP,
  Framed-Compression == Van-Jacobson-TCP-IP,
  Framed-MTU >= 576,
  Framed-Filter-ID =* ANY,
  Reply-Message =* ANY,
```

```
Proxy-State =* ANY,  
Session-Timeout <= 28800,  
Idle-Timeout <= 600,  
Port-Limit <= 2
```

bash-2.05# more clients.conf

```
client 127.0.0.1 {  
    secret      = <пароль к радиусу>  
    shortname   = localhost  
    nastype    = other    # localhost isn't usually a NAS...  
}  
  
client 192.168.2.1 {  
    secret      = <пароль к радиусу>  
    shortname   = localhost  
    nastype    = other    # localhost isn't usually a NAS...  
}
```

bash-2.05# more naslist

# NAS Name	Short Name	Type
#-----	-----	----
#portmaster1.isp.com	pm1.NY	livingston
#portmaster2.isp.com	pm1.LA	livingston
localhost	local	portslave
192.168.2.1	local	portslave

bash-2.05# more radius.conf

```
acct 192.168.2.1 <пароль к радиусу>  
auth 192.168.2.1 <пароль к радиусу>
```

Как правило, часто возникает проблема, что nibs трафик считает, а с линии юзера не скидывает, в логах вот что пишет:

```
Sat Apr 9 15:58:48 2005 : Info: rlm_nibs (rlm_nibs_accounting):  
User `drive' must be killed (no money on deposit) [192.168.2.1:0] |192.168.2.47]
```

в файлике nibs.conf есть вот такая строчка:

```
kill_program = "/usr/local/etc/raddb/kill.pl %{User-Name} %{NAS-IP-Address} %{Client-IP-Address} %{NAS-Port}"
```

Здесь автор примера написал собственный скрипт для сброса пользователей с линии.

Кроме скрипта nibs.conf есть смысл править в начале:

```
# SQL Server configuration  
server = "localhost"  
port = "3306"
```

```
login = "freenibs"  
password = "ваш пароль к базе"
```

Как настроить nibs и mysql написано в документации по nibs, на русском языке. ;)

Ниже приводится написанный автором примера скрипт на Perl.
Содержимое kill.pl

```
bash-2.05# more kill.pl
```

```
#!/usr/bin/perl -w
```

```
if( $ARGV[3] eq " " ) { die 'Usage: kill.pl user nasip userip nasport'; }
```

```
$user=$ARGV[0];  
$nasip=$ARGV[1];  
$userip=$ARGV[2];  
$nasport=$ARGV[3];  
$nastelnetport=5005;
```

```
use IO;
```

```
$sock = IO::Socket::INET->new(  
PeerAddr => $nasip,  
PeerPort => $nastelnetport,  
Proto => 'tcp') or die "Can not connect to mpd!\n$!";  
$sock->autoflush(1);
```

```
print $sock "link pptp", $nasport, "\n";  
print $sock "show radius\n";  
print $sock "close\n";  
close $sock;  
exit 0;
```

После подобных телодвижений, у меня всё заработало.

FAQ.

В: Что такое FreeNibs

О: Neon internet Billing System for FreeRadius. А по-русски, это биллинговая система для Интернет-провайдера ;)

В: Что может FreeNibs

О: ой, много – смотри главу «Функциональные особенности».

В: Возможно ли использовать систему NiBS для подсчета трафика в локальной сети

О: Да, но только при использовании VPN. Хорошо это или плохо - спорят до сих пор. Лично я считаю, что это очень гибко и безопасно.

В: Как можно узнать значения полей таблиц.

В: Какие поля используются в таблицах а какие нет.

О: Смотри главу «Структура базы данных FreeNIBS».

В: Можно ли менять поля в таблицах местами

В: Можно ли добавлять свои поля в таблицы

О: Менять и добавлять поля можно, выборка производится по именам полей. См. nibs.conf.

В: Можно ли изменять названия полей в таблицах.

О: Можно, но тогда надо подкорректировать файл nibs.conf.

В: Обязательно ли заполнять таблицы тарифов (времени и трафика) для пакета в котором это НЕ УЧИТЫВАЕТСЯ?

О: Нет!

Для этих пакетов можно не создавать записи. Также можно НЕ создавать записи и для пакетов, которые используют тарифы. Только тогда ничего учитываться не будет ;)

В: Умеет ли NiBS выдавать выделенные IP? То есть в зависимости от юзера давать различные адреса (например, для разделения сеток инет/не инет).

О: Можно задать IP и маску из SQL таблицы packets или users поля framed_ip и framed_mask.

В: login_time - каков формат этой строки? если я задаю интервал для ночного тарифа (т.е. 23:00-7:00), то при калькуляции session-timeout, login_time как-нибудь учитывается?

О: Читать в общем README от freeradius (аналогичен одноименному параметру для файла users, только добавлен еще день HI - праздник).

О: Формат прост - разделенные запятыми интервалы, состоящие из дня и времени (DDHHMM-HHMM) или просто дня (DD).

Вот поддерживаемые дни:

Su - воскресенье

Mo - понедельник

Tu - вторник

We - среда

Th - четверг

Fr - пятница
Sa - суббота
Wk - рабочие дни (= Mo,Tu,We,Th,Fr)
Al - все дни
HI - праздники

вот пример:

Wk0000-0800,Sa,Su,HI2200-1000

Впускать с понедельника по пятницу только ночью с 0 и до 8 утра, суббота и воскресенье круглосуточно, в праздники с 10 вечера и до 10 утра.

В: Есть ли планы на поддержку других баз данных (Oracle, Sybase, Postgre)

О: Поддержка встроена, только надо корректно настроить nibs.conf и откомпилировать соответствующие модули

В: Что такое RADIUS?

Как работает RADIUS?

Что такое ALIVE?

Почему нельзя отключить пользователя по исчерпанию лимита по трафику?

Почему нельзя считать трафик в реальном времени?

Почему нельзя сбросить пользователя в любой момент?

О: RADIUS это протокол разработанный не понятно кем, когда, и зачем. В нем очень много всего можно и еще больше нельзя. Протокол тупой для попыток делать на нем биллинг (Здесь и ниже - личное мнение автора предыдущих версий FreeNIBS).

Особенно он тупой для очень жадных провайдеров и/или их клиентов которые твердят, что пересидеть это есть нехорошо! Клиенты со своей колокольни вечно трубят – “нам пофиг - сами нас сбрасывайте, почему мы еще должны следить? чтобы не влезть в минуса!”

Провайдеры же твердят, что это не возможно (в лучшем случае) и все минуса придется оплатить.

Весь протокол, естественно, построен на пакетах.

Пакеты естественно ходят на назначенные порты.

Протокол состоит из 2 частей: AUTH часть и ACCT часть. AUTH часть принимает запросы на авторизацию (проверку пароля) и отвечает разрешением или запретом. ACCT часть принимает данные о начале сессии и конце сессии.

Как и говорилось выше, пакеты ходят на порты. Каждая часть на свой порт:

1812 udp (1645 udp) - Radius auth

1813 udp (1646 udp) - Radius acct

Обе части работают независимо. Рассмотрим каждую из них в деталях:

AUTH часть

Тут все просто! Получили логин и пароль, проверили наличие пользователя в базе, сравнили пароль, проверили счета, даты истечений пароля и решили – «пустить/не пустить».

Все начинается с прихода пакета Access-Request от клиента (pppd, Cisco...)

В текстовом виде пакет имеет примерно такой вид

```
Service-Type = Framed-User
Framed-Protocol = PPP
User-Name = "test"
User-Password = "pass"
NAS-IP-Address = 192.168.0.1
NAS-Port = 5
```

Что это значит? Это технические трудности, мы их не решаем ;)

Для биллинга тут есть интерес только в логине и пароле. Может немного в порте и IP сервера доступа.

Логин и пароль проверяются, если он неверный, то пользователя не впускает Access-Reject.

Если верный - выставляются нужные атрибуты и отправляются как Access-Accept.

```
Service-Type = Framed-User
Framed-Protocol = PPP
Framed-IP-Address = 192.168.1.5
Framed-Compression = Van-Jacobson-TCP-IP
Idle-Timeout = 900
Session-Timeout = 3600
```

Биллинг может в ответ подставить следующие атрибуты:

- выделенный IP (Framed-IP-Address);
- номер для callback (Callback-Number);
- время сессии (Session-Timeout).

Зная текущее время, цены на время и количество денег на счету можно рассчитать время, на которое хватит денег, и выставить Session-Timeout в это время.

Если пользователь засидится на линии, то его все равно сбросит по истечению этого времени.

Как видно из этого примера - ничего точного с трафиком при такой схеме придумать нельзя.

Можно конечно анализировать трафик за предыдущее подключение, рассчитывать скорость для этого пользователя...

Можно еще брать какую-то среднюю скорость, например 21000 и рассчитывать время для этой скорости...

Эти обе попытки не точные и могут скинуть как намного раньше, так и намного позже

Были еще опытные варианты параметра Session-Octets-Limit для указания лимита трафика на сессию. Но это не все серверы доступа поддерживают...

ACCT часть

У acct части есть 3 вида пакетов (точнее их больше, но нас пока интересуют только 3)

```
START
STOP
ALIVE
```