

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Средства он-лайн общения в Linux

*Утверждено Редакционно-издательским советом университета
в качестве методических указаний к лабораторной работе*

САМАРА
Издательство СГАУ

2010

УДК 004.451

Составитель А.М. С у х о в

Рецензент: к.т. н., доц. Попов С.Б.

Механизмы безопасности в Linux: Методические указания к лабораторной работе/ Сост. А.М. Сухов. - Самара: Изд-во Самарского государственного аэрокосмического университета, 2010. 48 с.

В настоящих методических указаниях приведен материал, необходимый для выполнения лабораторных работ по дисциплинам «Операционная система Linux на высокопроизводительных кластерах», «Перспективные информационные технологии»

Целью лабораторной работы является изучение основных возможностей по обеспечению безопасности информационных систем и путей ее достижения в операционной системе Linux.

Предназначено для слушателей ФПК СГАУ и студентов специальностей 010500, 010501

© Самарский государственный
аэрокосмический университет, 2010

1. Цель лабораторной работы

- изучение основных возможностей по обеспечению безопасности информационных систем
- получение опыта работы с основными утилитами, обеспечивающими различные аспекты безопасности информационных систем в операционной системе Linux
- классификация основных угроз безопасности информационных систем

2. Основные сведения

Любой мало-мальски сложный проект одному не поднять. Нет, конечно, можно постараться и через пару лет выдать полусырой продукт, но зачем? Ведь гораздо веселее, быстрее и эффективнее работается в команде. А чтобы делать это умеючи, придется выделить время и освоить полезные инструменты.

Умное слово «collaborate» сейчас популярно как никогда. Поддержка «совместной работы», а именно так оно переводится, означает, что сервис предоставляет возможность одновременной работы сразу для нескольких пользователей, которые трудятся над решением общей задачи. Причем, у каких-то сервисов коллективный труд является лишь одной из опций, а у других – непосредственным предназначением. К сожалению, «collaborate» реальную возможность работать коллективом обозначает далеко не всегда, а среди сервисов, по-настоящему поддерживающих совместную работу, бесплатны лишь некоторые. Можно долго хвалить всемирно известный инструмент *Basecamp*, предназначенный для управления проектами, но весь мед портит ложка дегтя. За пользование им придется выкладывать \$24 в месяц – и это минимум!

1. Работа с документами: Google Docs

Работая в издательстве, имеешь дело с текстами постоянно. Обычный Word, установленный практически везде, отлично справляется со всеми задачами, но сильно обламывается, когда речь идет о коллективной работе. Система рецензирования, хоть и удобна, и позволяет вносить правки разными людьми, но не дает самого главного – работать с документом одновременно нескольким пользователям. К тому же, возникает серьезная проблема, когда нужно предоставить кому-то доступ к документам, находящимся внутри локалки издательства. Так как же быть? На помощь приходит Google Docs, о котором не слышал, пожалуй, только ленивый. Офисный пакет от Google работает через браузер, документы хранит прямо на своих серверах, и при этом (а это самый смак!) он изначально рассчитан на коллективную работу. Каждое внесенное в файл изменение отображается в специальной базе данных, и любую правку в нужный момент можно отменить. С авторами порой вполне удачно получается вместе редактировать текст, уточняя некоторые моменты и по ходу дела рецензируя отдельные моменты с указанием ошибок. А между редакторами расшарен специальный документ, в котором мы делимся друг с другом идеями для статей.

Впрочем, подобные возможности удобно использовать и вне работы. Прямо сейчас Nikitoz планирует новогодние каникулы, оформляя все в виде электронной таблицы с указанием различных расходов, и предоставляет доступ к документу всем участникам поездки.

Чуть меньшими возможностями по коллективной работе обладает аналогичный проект - Zoho Writer. Зато во всем остальном, включая интерфейс, он максимально приближен к Word'у.

4. Проводим встречи онлайн: DabbleBoard и Twiddla

Тратить время на бесконечные встречи в офисе, да еще вытаскивать из дома удаленных сотрудников? Нееее, это не для нас. Уж мы-то знаем, что сейчас вообще можно отказаться от совещаний в редакции. Благодаря Skype'у, легко можно общаться с фрилансером хоть из Зимбабве – при этом не платя ни копейки, а для дополнительных объяснений использовать вспомогательные инструменты. Сразу приходит на ум офисная доска, на которой размашисто рисуешь маркером, объясняя очередную гениальную идею. Реализация подобной офисной доски есть и в Сети.

Лично мне удобнее использовать подобные вещи прямо через браузер. Для Whiteboarding'a (так называется процесс рисования на онлайн-доске несколькими пользователями) есть несколько профильных сервисов. Так, DabbleBoard основан на технологии Flex и полностью предназначен для рисования от руки. Понятно, что нарисовать, скажем, ровный круг не так-то просто, однако сервис распознает очертания основных фигур и автоматически преобразует изображение, чтобы фигура выглядела «красивой и опрятной». Ты можешь создавать и свои собственные тулкиты (наборы объектов рисования). Например, я дважды с ее помощью рисовал эскизы интерфейса.

Twiddla, в отличие от DabbleBoard, использует технологию Ajax, но предоставляет куда большие возможности для проведения совещаний онлайн. На рабочую область можно помещать иллюстрации, текст, различные документы и математические форумы. А по ходу дела – просматривать Web-сайты, оставляя на нужной области заметки, быстро обмениваться файлами, устраивать голосовые беседы. Мечта!

5. Расшариваем рабочий стол: Vuew

Vuew – это тоже сервис для проведения совещаний онлайн. У него есть одна замечательная опция, за которую разработчикам хочется сказать отдельное спасибо. Прямо из браузера с ее помощью можно предоставить доступ к своему рабочему столу! Получаем практически полный эффект присутствия с участием до 20 человек: использовать whiteboarding – это раз, обмениваться файлами – два, показывать презентации – три, совместно править файлы – четыре, наглядно показывать что-то и объяснять, передавая изображения со своего рабочего стола – пять. Для работы необходимы лишь установленный Flash последней версии и Java.

6. Редактирование текста в реальном времени: Etherpad

Сервис Etherpad – это воплощение гениальности и простоты, разработанное группой ex-сотрудников Google. Когда требуется что-то придумать, прикинуть и совместными

усилиями оформить в виде текста, – лучшего инструмента не найти. Все просто: ты создаешь новый документ и получаешь ссылку, которую отправляешь остальным участникам импровизированного совещания – в итоге, каждый получает доступ к одному единственному текстовому редактору. Все строки пронумерованы, можно писать текст... А весь фокус в том, что внесенные изменения в реальном времени отображаются у каждого из участников, – и отображаются разными цветами. Если совместить это с голосовой конференцией, можно быстро набросать нужный текст или даже программный код. Для создания промежуточных версий реализована серьезная система ревизий.

7. Совместная разработка: Assembla.com

Рассказывая о Teamer, я акцентировал внимание на том, что это очень простой сервис, практически не требующий времени для освоения. Но если речь идет о серьезном проекте по разработке ПО, то тут уже сам Бог велел выделить уделить время, чтобы разобраться с замечательным сервисом Assembla.com.

Итак, что это такое? Assembla.com – это полноценный сервис для команды кодеров, занимающихся разработкой ПО. Для проекта ты получаешь любые нужные средства:

- системы контроля версий (на выбор: SVN, Git, Mercurial, а также интеграция с внешним SVN или сервисом Github);
- вики для ведения документации (однако, этот компонент самый слабый и неудобный из всех, если начистоту);
- встроенную систему тикетов или совмещенную с популярным пакетом Trac, дополненным собственным компонентом просмотра кода (Code Browser);
- чат для общения всех участников проекта;
- развитые средства совместной работы над изображениями;
- графики выполнения ключевых шагов (мейлстоунов) и другое.

При этом неважно, придерживаешься ли ты популярных сейчас Agile/SCRUMM манер ведения проекта или просто разрабатываешь что-то для себя в свободное время - всегда можно настроить среду так, как хочется. Все компоненты, включая систему контроля версий, доступны с панели управления и устанавливаются простым кликом. Для продвинутых пользователей в Assembla.com тоже найдется сюрприз. Если ты используешь встроенный модуль тикетов (а не Trac), как я, то рано или поздно тебе может надоесть делать скриншоты багов в ПО или на сайте, потом их обрабатывать и прикреплять к тикетам как файл. Оказывается, у сервиса есть кнопочка «Take screenshot», которая запустит специальную Java-программу для автоматического снятия скриншотов. И таких мелочей, в лучшую сторону отличающих assembla.com от других средств, наберется много. Например, интеграция с микро-блоггингом Twitter.com - теперь все твои друзья будут знать, как продвигается проект, и не станут задавать глупых вопросов, когда релиз (это же касается и менеджеров или заказчиков). А бекап на сервера Amazon S3 спасет, даже если полетят жесткие диски у всех

участников команды. Самым крутым разработчикам припасли подробно документированный API для работы с сервисом программным образом и даже плагин для Eclipse, позволяющий работать с задачами и тикетами прямо в IDE.

Помимо всего прочего, Assembla.com может быть отличным инструментом для фрилансеров и стартапов, так как содержит встроенные средства распределения задач, ведения своего портфолио, поиска и набора сотрудников для определенных работ и учет оплат. За некоторые возможности, кстати, придется платить тебе самому. Так, бесплатные Аккаунты могут быть только открытыми (то есть, код и внутренняя переписка доступны всем участникам), да и места, выделяемого под хранение кода в системе контроля версий и файловое хранилище, не так и много, – всего 200 Мб. Но стоит заплатить (исходя из расчета, 2 USD в месяц на участника проекта), как и места станет больше (до 5 Гб), и расширенные возможности появятся (тот самый бэкап на Amazon S3). Если ты не создаешь нечто секретное (да-да, не пишешь убийцу Google или новую ОС, которая порвет Windows), то все возможности для тебя будут бесплатными!

Skype

Для качественной организации работы офиса, да и для личного общения, сегодня уже нелегко представить использование только телефонной связи и обмена сообщениями по электронной почте. На смену приходят новые технологии общения такие как программы мгновенного обмена сообщениями. Это класс программ, предназначенных для обмена сообщениями через Интернет в реальном режиме времени (служба мгновенных сообщений — Instant Messaging Service, IMS). В зависимости от типа позволяют передавать как текстовые сообщения, так и файлы, звуковые сигналы, видео.

Каждая из программ обмена сообщениями для передачи данных использует те или иные протоколы (протокол — стандарт обмена данными). В частности, для обмена сообщениями в сети ICQ (<http://www.icq.com>) используется закрытый протокол Oscar, разработанный компанией Mirabilis. В основе обмена текстовыми данными в Google Talk лежит открытый протокол Jabber (XMPP), а для передачи звука — Jingle, расширение протокола XMPP для передачи голоса. Подробнее о данных средствах обмена сообщениями и протоколах поговорим в следующей статье. А сегодня хотелось бы остановиться на своего рода универсальном сервисе для обмена как текстовыми сообщениями, так и передачи звука и видео,.

Skype — это система передачи голосовых сообщений, запатентованная Skype Technologies S.A. (<http://www.skype.com>) и обеспечивающее бесплатную голосовую связь с передачей видео через Интернет между компьютерами, а также платные услуги для связи с абонентами стационарной или мобильной телефонной сети (<http://ru.wikipedia.org/wiki/Skype>). В основе Skype лежит технология передачи голоса по IP¹-протоколу (VoIP — Voice over IP). Данная технология позволяет разговаривать абонентам друг с другом, используя наушники и микрофоны, подключенные к



компьютерам. Головой сигнал передается в цифровом виде и перед передачей преобразовывается. Реализована технология VoIP в 1993 г. (<http://ru.wikipedia.org/wiki/VoIP>).

Технология передачи телефонных разговоров абонентов по протоколу IP называется **IP-телефонией** и является частным случаем технологии VoIP. Протоколы IP-телефонии обеспечивают регистрацию устройства для передачи данных из стационарной телефонной сети по IP-сети, в данном случае Интернет, а так же вызов, переадресацию вызова, установление соединения и т.д. Для построения такой технологии передачи данных используются различные протоколы. Наиболее популярными являются протоколы SIP (Session Initiation Protocol — протокол установления сессии, <http://ru.wikipedia.org/wiki/SIP>) и H.323 (<http://ru.wikipedia.org/wiki/H.323>). В частности, решения на основе открытого протокола SIP предлагаются для построения корпоративной системы коммуникации во многих странах. В Беларуси услуги по IP-телефонии предлагает только Белтелком (<http://www.beltelcom.by>).

Возможности Skype

В основе архитектуры Skype лежит технология соединения равноправных узлов P2P (peer-to-peer — одноранговые децентрализованные сети)². Для передачи звонка клиент Skype ищет не центральный сервер, а других клиентов Skype. На основе этих связей строит сеть, которая может быть использована для того, чтобы находить других пользователей и посылать им сообщения.

Программа-клиент Skype (см. рис. 1) распространяется бесплатно (freeware) и свободна от рекламы (adware) и шпионских модулей (spyware). Доход система Skype получает за счет оказания платных услуг.

К бесплатным услугам Skype относятся:

1. Звонки между абонентами Skype
2. Переадресация вызовов от абонентов Skype другим абонентам Skype из записной книжки
3. Видеозвонки через Skype
4. Групповые и индивидуальные чаты
5. Аудио конференции
6. Переадресация звонков другим абонентам Skype

Как пользоваться Skype?

Для того, чтобы зарегистрироваться в Skype, необходимо скачать с сайта <http://www.skype.com> и установить приложение Skype. На сайте предлагаются версии для различных операционных систем, включая Windows, Mac OS X и GNU/Linux (Ubuntu, Debian Etch, Mepis, Xandros, Centos, Fedora 7, Fedora Core 6, OpenSUSE 10+, Mandriva и др.). В зависимости от операционной платформы Skype может отличаться по своему функционалу. Для ОС Windows предлагаются две версии: стандартная и бизнес-версия. Последняя отличается типом инсталлятора, более защищенными инструментами обеспечения безопасности и др.

Создать новую учётную запись в Skype

Полное имя

*Email
Корпоративный email - администраторы смогут восстановить утраченный пароль

Пожалуйста, присылайте мне новости Skype и специальные предложения на email

*Выберите Skype-имя
От 6 до 32 символов

*Пароль
От 8 символов

*Подтвердите пароль
От 8 символов

Да, я прочитал и согласен с [Лицензионным Соглашением Корпоративного Пользователя Skype](#), условиями [использования сервиса Skype](#) и [форматированной политикой конфиденциальности Skype](#)

*Полит. персональные сведения об обработке

Минимальные

требования:

- процессор 400 MHz
- 256 MB RAM.
- 20 MB свободного места на диске (для ОС GNU/Linux)
- Микрофон и наушники или гарнитура
- Подключение к Интернет

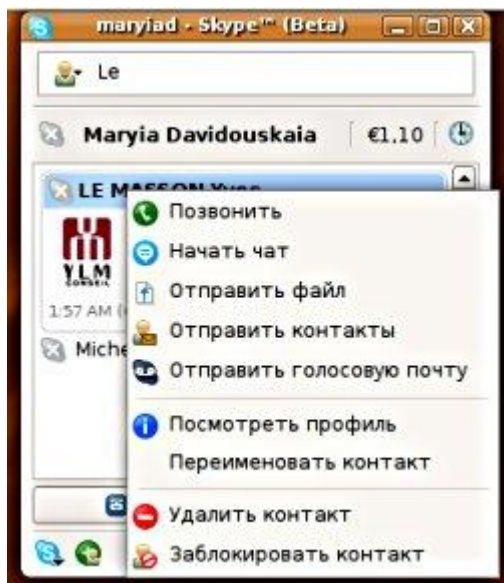
После установки Skype пользователь получает возможность зарегистрировать

учетную запись. Для регистрации необходимо заполнить обязательные поля учетной записи (см. рис. 2). Если Вы используете Skype на публичном компьютере, то не рекомендуется сохранять пароль в ходе создания учетной записи.

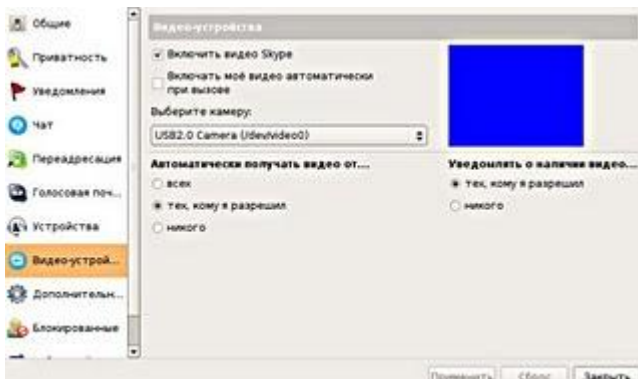
После регистрации Вы можете найти абонентов для общения по имени Skype, персональным данным, стране проживания, полу и т.д. (см. рис. 3). Как и в других программах обмена сообщениями в Skype можно настроить «белый» и «черный» список пользователей. В «белый» список входят пользователи из Вашего списка контактов Skype, а в «черный» - те, кого Вы отнесли в список заблокированных контактов и с кем не хотели бы общаться. Чтобы защитить себя от нежелательных контактов, Вам необходимо настроить параметры конфиденциальности (см. рис. 4), разрешив звонки и общение в чате только с пользователями из списка контактов.

Для того, чтобы позвонить абоненту Skype из списка контактов, достаточно дважды щелкнуть по его имени. Если Вы настроили открытие по двойному щелчку мыши окна чата, то позвонить абоненту Вы можете, выделив его в списке и нажав кнопку

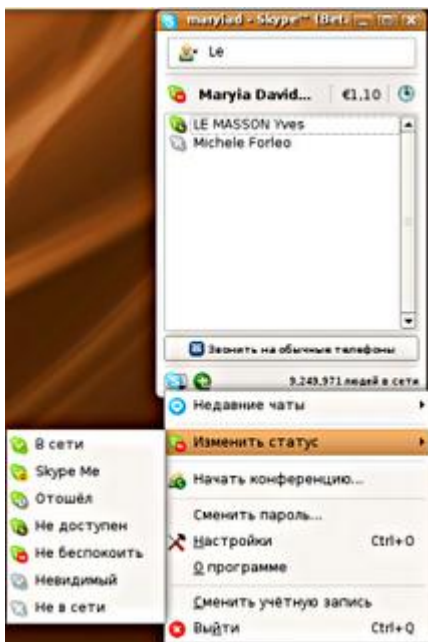
«Позвонить» или выбрав команду «Позвонить» в контекстном меню для данного абонента (см. рис. 5). На период звонка открывается окно, отображающие данные абонента, которому Вы звоните, и данные по длительности звонка. Если абонент на месте, то он ответит на звонок сразу. Если отсутствует и нет переадресации, то Ваш звонок сохранится в истории событий Skype.



При подключенной веб-камере к компьютеру во время звонка Вы можете разрешить просмотр видео Вашему абоненту, даже если у него нет веб-камеры. Правда, в данном случае видео-конференция будет односторонней. Пожалуйста, имейте в виду, что при низкой скорости подключения к Интернет качество одновременной передачи звука и видео будет низким. Для обеспечения достаточного уровня конфиденциальности Вы так же можете настроить кому разрешается просмотр Вашего видео, всем пользователям Skype или тем, кто входит в Ваш список контактов, или не разрешать никому (см. рис. 6). Рекомендуется разрешить только пользователям из списка отправлять уведомление о возможности передачи видео или же полностью отключить передачу видео.



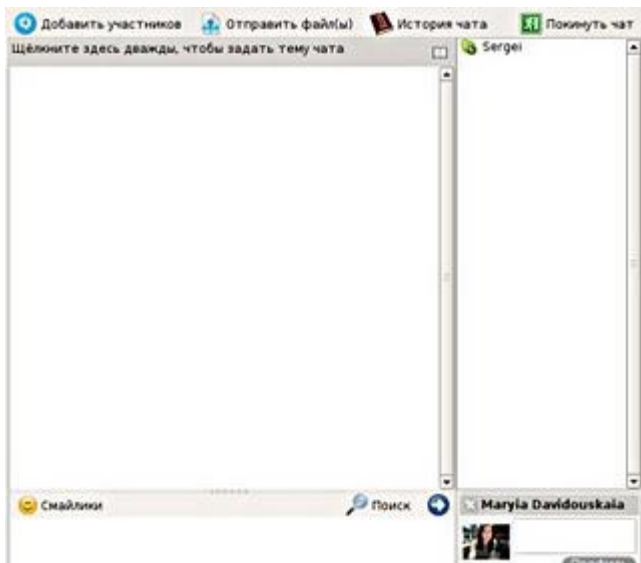
Для того, чтобы начать аудио конференцию с несколькими абонентами, Вы можете воспользоваться командой «Начать конференцию» (см. рис. 7). В результате откроется окно со списком Ваших контактов, из которых Вы можете выбрать участников конференции. В аудио конференции могут принимать участие от 5 до 10 человек. Это ограничение на число участников связано с ограничениями, накладываемыми характеристиками Вашего персонального компьютера. В ходе конференции старайтесь не перебивать собеседников и не говорить одновременно. Так Вы сможете донести свое мнение участникам. :-)



Кроме аудио-конференций с абонентами из списка контактов Вы можете создать скайпкаст (см. рис. 8). Скайпкасты (skyperecasts) – это аудио конференции через Skype, организуемые ведущими для участников со всего мира. Количество участников может достигать от 100 до 200. Скайпкасты позволяют проводить обсуждения в прямом эфире, знакомиться с новыми людьми и просто общаться. Скайпкаст может быть создан любым абонентом, зарегистрированным в сети Skype и пользующимся версией Skype 3.2 и выше. Создать скайпкаст можно или через меню программы Skype или на странице создания скайпкастов.



Обмениваться текстовыми сообщениями в чате Вы можете с одним абонентом или подключить других участников, используя кнопку «Другие участники» в окне чата (см1 9). Так же Вы можете создавать публичные чаты, к которым может подключиться любой пользователь Skype. Для создания публичного чата выполните команду "Создать открытый чат" из меню "Чаты". В процессе создания чата задается его тема и определяются права участников. Публичные (открытые) чаты можно использовать как для общения, так и для консультаций по различным вопросам как сотрудников компании, так и всех пользователей, которые подключаться к чату. Ведущий публичного чата определяет правила поведения участников чата. В частности, определяют все ли пользователи могут писать в чат, кто может писать и читать, а кто только только читать.



Последние версии Skype позволяют исправлять ошибки в отправленных текстовых сообщениях, и Ваши исправления будут видны участникам чата. В настройках Skype Вы можете задать период хранения истории чата. Возможны варианты: 2 недели, месяц, 3 месяца, вечно и отключить историю чата. Если история чата хранится длительный период, то ее можно просмотреть в любой период и провести по ней поиск.

Абонентам из списка контактов Вы можете пересылать файлы, другие контакты из Вашего списка, отправлять голосовую почту при подключенной услуге и т.д. Если список контактов большой, то возможен поиск по фамилии и имени пользователя или его Skype-имени. Только пользователи Skype для Windows могут создавать группы. Перед тем как создавать группы эту функцию необходимо активировать с помощью команды **«Показать группы»** из меню **«Вид»**. Распределив пользователей по группам, Вы можете участникам группы отправить файл, начать группой чат с участниками группы или аудио-конференцию и др. Кроме этого, для той или иной группы Вы можете изменить тип группы на общую группу.

Общие группы позволяют использовать список контактов совместно с другими пользователями. При создании общей группы Вы отправляете приглашение всем абонентам, которых в нее включаете, и все те, кто принимает Ваше приглашение, смогут добавить в свой список контактов Skype данные других абонентов этой группы. Любой из членов общей группы может приглашать в нее других участников. При присоединении нового участника информацию о нем получают все члены группы.

Общаться с абонентами Skype Вы можете, используя и мобильные телефоны. Для мобильных платформ разработано приложение Fring (<http://www.fring.com/>), позволяющее разговаривать или обмениваться текстовыми сообщениями при подключении к Интернет через GPRS или Wi-Fi. С помощью данной программы Вы можете звонить на мобильные телефоны или абонентам Skype, Google Talk, MSN Messenger и т.д. На данный момент доступна версия для платформы Symbian (модели Nokia). Fring может автоматически определять наличие Wi-Fi точки доступа сети и переключать звонки абонента на интернет-звонки, тем самым значительно сократить затраты пользователя на мобильную связь.

Популярность сервиса Skype привела к тому, что производители оборудования стали предлагать различные устройства, облегчающие коммуникацию. Например, вместо наушников и микрофона к компьютеру можно подключить USB Интернет- телефон в виде телефонной трубки или USB флешфон (флеш-диск с подключенной гарнитурой, например, Skypemate USB-M3K), или устройство, обеспечивающее громкоговорящую связь (Speakerphone).

Надеюсь, что изложенный материал поможет Вам понять достоинства Skype. И, если Вы еще не стали зарегистрированным пользователем Skype, то зарегистрироваться и открыть для себя новый способ общения.

1. IP — Internet Protocol — сетевой протокол, используемый для ненадежной доставки данных. В протоколе IP каждому узлу сети ставится в соответствие IP-адрес, уникальный числовой идентификатор (<http://ru.wikipedia.org/wiki/IP>).
2. Компьютерная сеть, в которой все узлы равноправны, т.е. в сети отсутствует выделенный сервер, а каждый узел (peer) является как клиентом, так и сервером.

Потоковое аудио/видео вещание с VideoLAN

Прошло то время, когда для открытия собственной радио/теле-станции требовалось приобретать дорогостоящее оборудование, а также собирать множество лицензий, любая из которых могла быть отозвана в самый неподходящий момент. Теперь достаточно скачать бесплатный программный комплекс VideoLAN и, немного повозившись с его настройками, вещать хоть на локальную сеть, хоть на весь Интернет с максимально доступным для данного канала качеством, или же использовать VideoLAN в качестве обычного медиа-плеера, поддерживающего множество входных форматов с возможностью записи чужого потокового медиаконтента на жесткий диск.

Введение

Какой смысл в потоковом вещании, когда выложенный на HTTP/FTP-сервер медиафайл может быть скачен на локальный диск пользователя и просмотрен/послушен в любое удобное для него время бесчисленное множество раз! Потоковое вещание "привязывает" целевую аудиторию к серверу трансляции, возвращаясь к традициям десятилетней давности, когда приходилось "караулить" интересные передачи или устанавливать таймер на (видео)магнитофоне для

автоматической записи передачи, транслируемой в "неудобное" для зрителя время. Казалось бы, Интернет исповедует гораздо более прогрессивный подход. Или все-таки нет?!

Начнем с того, что каналы не резиновые, их пропускная способность ограничена и все крупные сервера, раздающие медиаконтент обычным способом, обычно очень сильно перегружены. Распределенные файлообменные сети существенно снижают нагрузку, однако реальная скорость передачи данных у них чрезвычайно низка, да еще подолгу стоять в очередях приходится, что совсем не по капиталистически.

Компромиссной технологией раздачи медиаконтента является онлайн-вещание по технологии Multicast, обеспечивающей одновременную доставку идентичного контента всем запросившим его пользователям, что существенно разгружает каналы передачи данных, но... ограничивает свободу пользователей в выборе контента, поскольку если к нам подключились сто тысяч пользователей и каждый из них выбрал свой файл, то никакого выигрыша мы не получим. С другой стороны, никто нам не запрещает иметь несколько независимых Multicast-каналов, передающих различные файлы, к которым может подключаться кто угодно. Разница между обычной "скачкой" файла с сервера в том, что трансляция не позволяет слушателем/зрителем управлять потоком, и они вынуждены слушать/смотреть файл с момента подключения к серверу, который к тому времени мог проиграть половину файла. В некоторых случаях это приемлемо, в некоторых - нет. Как показывает практика, достаточно большой аудитории пользователей совершенно неважно, что именно играет в данный момент - главное, чтобы что-то вообще играло.

К тому же в потоковое аудио/видео намного легче "врезать" рекламу или прочие вставки типа "breaking news", да и квалификация среднестатистического пользователя не позволяет сохранять потоковый контент на диск, что очень нравится держателям авторских прав и прочим медиамагнатам.

Словом, существуют тысячи причин, чтобы воздвигнуть сервер потокового аудио/видео, вещающий в пределах локальной сети или даже охватывающей весь Интернет. Для потокового вещания написано огромное количество программ (и каждый день появляются все новые), но хороших из них немного, а хороших, открытых и бесплатных - еще меньше.

VideoLAN - многофункциональный комплекс, портированный практически под все операционные системы, поддерживающий множество протоколов, форматов и контейнеров, который можно использовать и как локальный аудио/видеоплеер, и как сервер трансляции (см. рис. 1).

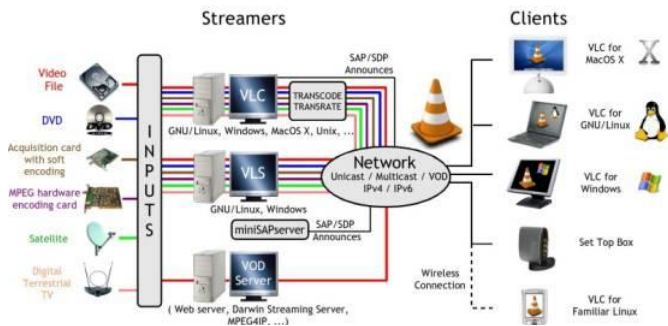


Рисунок 1. Потокное вещание в локальных/глобальных сетях.

Обзор основных возможностей

VideoLAN - это некоммерческий проект, бесплатную версию которого (вместе с исходными текстами и готовыми бинарными сборками) всегда можно скачать с официального сервера <http://www.videolan.org/>.

Клиентская и серверные части исправно работают под Linux, Windows, Mac OS X, BeOS, xBSD, Solaris, Familiar Linux, Yoru/Linux и QNX, однако их функциональность различна и в зависимости от выбранной платформы варьируется в очень широких пределах (подробнее см. рис. 2).

Interfaces and control						
Default	Wx/Widgets	Cocoa	Wx/Widgets	Native	Wx/Widgets	GPE
Qt 4 ^{1.1}	✓	✗	✓	-	?	✗
Skins	✓	✗	✓	✗	✓	✗
Web	✓	✓	✓	✓	✓	✓
Telnet	✓	✓	✓	✓	✓	✓
Command line	✓	✓	✓	✓	✓	✓
Infrared	✗	✗	✓	✗	✗	✗

Miscellaneous						
SAP/SDP announces	✓	✓	✓	✗	✓	✗
Bonjour protocol	✗	✓	✓	?	?	?
Mozilla/Firefox plugin	✓	✓	✓	✗	✗	✗
ActiveX plugin	✓	-	-	-	-	-
SVCD Menus	✓	✗	✓	✗	✓	✗
Localization	✓	✓	✓	✓	✓	✓
CD-Text ^{1.1}	✓	✗	✓	✗	✓	-
CDDB CD info	✓	✓	✓	✗	✓	-
IGMPv3 ^{1.1}	✓	✗	✓	✗	✓	✓
IPv6 ^{1.1}	✓	✓	✓	✗	✓	✓
MLDv2 ^{1.1}	✓	✗	✓	✗	✓	✓
CPU acceleration ^{1.1}	✓	✓	✓	✓	✓ ^{1.1}	✗

Рисунок 2. Возможности программы VideoLAN на каждой из поддерживаемых ею платформ.

Поддерживаются следующие входные форматы данных: MPEG-1, MPEG-2, MPEG-4/DivX (считываемые с локального жесткого диска или CD/DVD); "настоящие" DVD и VCD; спутниковые карты, работающие по стандарту (DVB-S); потоковое видео, "упакованное" в MPEG-1, MPEG-2 и MPEG-4 (то есть, VideoLAN может работать не только как сетевой транслятор, но и как **ретранслятор** чужого контента с возможностью сохранения последнего на жесткий диск).

В настоящий момент реализованы два основных протокола трансляции: **Unicast** ("узконаправленное" вещание с доставкой контента только одному целевому узлу) и **Multicast** (групповая трансляция с доставкой одного и того же контента множеству узлов). Также (формально) имеется возможность широковещательной рассылки контента всем узлам локальной сети (для этого достаточно указать в качестве целевого IP-адреса 255.255.255.255), но с высокой степенью вероятности она будет задавлена брандмауэрами и маршрутизаторами, так что без их радикальной перестройки сеанс вещания не состоится даже в рамках локальной сети.

Еще имеется ограниченная поддержка видео-по-требованию (Video-on-Demand или, сокращенно, VoD) с возможностью выбора контента по HTTP или TELNET интерфейсам, однако эта возможность обычно используется исключительно администраторами для удаленного управления сервером трансляции.

Кстати, об интерфейсах. VideoLAN поддерживает широкий ассортимент, способный удовлетворить даже самых изысканных гурманов: GUI, NCUSERS, командная строка, HTTP и даже плагины для некоторых популярных браузеров.

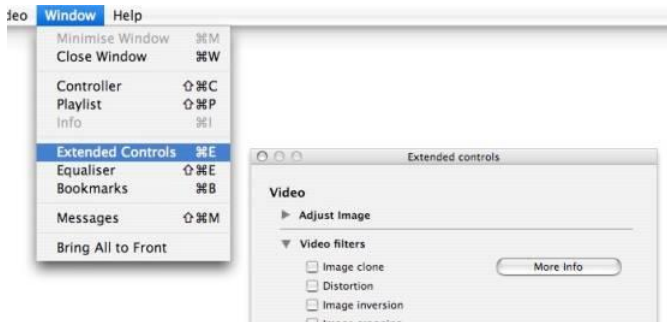


Рисунок 3. Графический интерфейс программы VideoLAN.

```

VLC media player (ncurses interface) [ h for help ]

Source  : BG/05 - Cup Of Coffee.mp3
State   : Playing
Position : 0:01:42/0:04:31 (37.89%)
Volume  : 62%

-----
                                Playlist
-----
0 - 'BG/01 - Shut Your Mouth.mp3' (01 - Shut Your Mouth.mp3)
1 - 'BG/02 - Androgyny.mp3' (02 - Androgyny.mp3)
2 - 'BG/03 - Can't Cry These Tears.mp3' (03 - Can't Cry These Tears.mp3)
3 - 'BG/04 - Til The Day I Die.mp3' (04 - Til The Day I Die.mp3)
> 4 - 'BG/05 - Cup Of Coffee.mp3' (05 - Cup Of Coffee.mp3)
5 - 'BG/06 - Silence Is Golden.mp3' (06 - Silence Is Golden.mp3)
6 - 'BG/07 - Cherry Lips (Go Baby Go... (07 - Cherry Lips (Go Baby Go!),mp3)
7 - 'BG/08 - Breaking Up The Girl.mp3' (08 - Breaking Up The Girl.mp3)
8 - 'BG/09 - Drive You Home.mp3' (09 - Drive You Home.mp3)
9 - 'BG/10 - Parade.mp3' (10 - Parade.mp3)
10 - 'BG/11 - Nobody Loves You.mp3' (11 - Nobody Loves You.mp3)
11 - 'BG/12 - Untouchable.mp3' (12 - Untouchable.mp3)
12 - 'BG/13 - So Like A Rose.mp3' (13 - So Like A Rose.mp3)

```

Рисунок 4. Текстовый NCURSES интерфейс.

Контейнеры, в которые помещается транслируемый поток, зависят от типа трансляции, допустимые комбинации которых перечислены в таблице на рис. 5. Естественно, все это хозяйство работает как с IPv4, так и с IPv6.

	UDP	RTP	HTTP	MMSH	File
ES	✗	✓	✓	✗	✓
TS	✓	✓	✓	✗	✓
CSG	✗	✗	✓	✗	✓
ASF	✗	✗	✓	✓	✓
MMS	✗	✗	✗	✗	✓
QuickTime	✗	✗	✗	✗	✓
Raw	✓	✓	✗	✗	✓
MJPEG	✗	✗	SVN only	✗	SVN only

Рисунок 5. Допустимые комбинации протоколов трансляции с контейнерами, в которые упаковывается транслируемый медиа-поток.

Устанавливаем VideoLAN

На странице <http://www.videolan.org/vlc/> выложены готовые бинарные сборки для следующих операционных систем, установка которых проходит без проблем (автор тестировал Windows, Debian, Mandriva, Fedora Core и SuSE):

- Windows;
- Mac OS X;
- BeOS;
- Debian GNU/Linux;
- Ubuntu Linux;

- Mandriva Linux;
- Fedora Core;
- Familiar Linux;
- SUSE Linux;
- Red Hat Linux;
- Slackware Linux;
- ALT Linux;
- YOPY/Linupy;
- Zaurus;
- Arch Linux;

Операционные системы, перечисленные ниже, формально поддерживаются, но заниматься компиляцией под них приходится самостоятельно.

- NetBSD;
- OpenBSD;
- FreeBSD;
- Solaris;
- QNX;
- Gentoo Linux;
- Crux Linux;

Поскольку VideoLAN входит в набор портов для FreeBSD (любимой xBSD системы автора!), то сборка под нее проходит без единого слова нареканий. Достаточно набрать команду: `"# cd /usr/ports/multimedia/vlc && make install clean"` и немного подождать. Остальные системы в этом плане несколько менее предсказуемы и способны преподнести букет неприятных сюрпризов, но всестороннее тестирование VideoLAN'a выходит за рамки данной статьи.

Начинаем трансляцию

Разработчики рекомендуют начинающим "телемеханикам" использовать "Мастер Трансляции" (Файл -> Мастер, соответствующий горячей клавише <CTRL-W>), однако он содержит несколько подводных камней, кроме того ограничивает наши возможности по кручению тонких настроек, а потому мы пойдем другим путем.

В меню "Файл" выбираем пункт "Открыть файл" <CTRL-F>, если хотим транслировать один или несколько mp3/avi/mpeg-файлов; "Открыть каталог" <CTRL-E>, чтобы одним махом выделить все содержащиеся в нем файлы; "Открыть диск" <CTRL-D> для трансляции контента непосредственно с DVD/VCD/Audio CD (правда, несмотря на все ухищрения, автору так и не удалось заставить VideoLAN транслировать музыку прямо с Audio CD без ее предварительной перекодировки); еще можно открыть URL (для ретрансляции чужого контента) или выбрать устройство типа спутниковой карты.

Начнем с простого. С обычного AVI/MPEG файла, транслируемого по локальной сети на соседний компьютер. Жмем <CTRL-F> (см. рис. 6) и через "Обзор" выбираем один или несколько файлов (не обязательно одного и того же типа). Для подключения субтитров (если мы хотим их подключать) взводим одноименную галочку и указываем путь к файлу с субтитрами, положение и цвет которых определяется кнопкой "Расширенные настройки". VideoLAN поддерживает множество субтитров различных типов (включая .srt и .sub), что позволяет нам, в частности, накладывать рекламу на видеопоток или различные сведения чисто информационного характера.

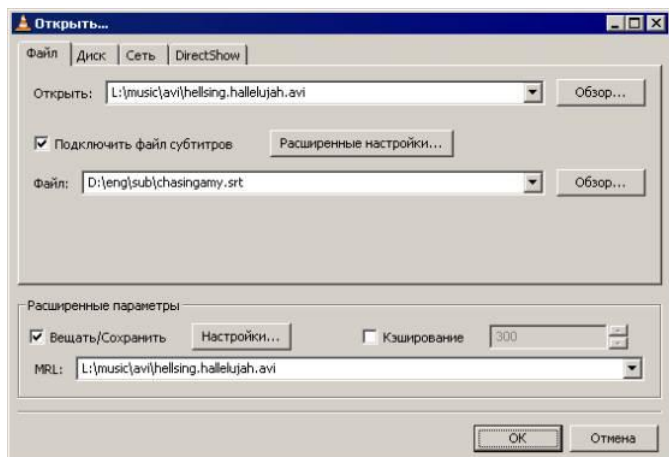


Рисунок 6. Выбор файла для трансляции в локальную/глобальную сеть.

Взводим галочку "Вещать/Сохранить" и давим "Настройки", открывающие огромный диалог (см. рис. 7) с кучей галочек, строк редактирования и прочих элементов управления, в которых на первых порах не так-то просто разобраться, но ведь мы же не из пугивых!!!

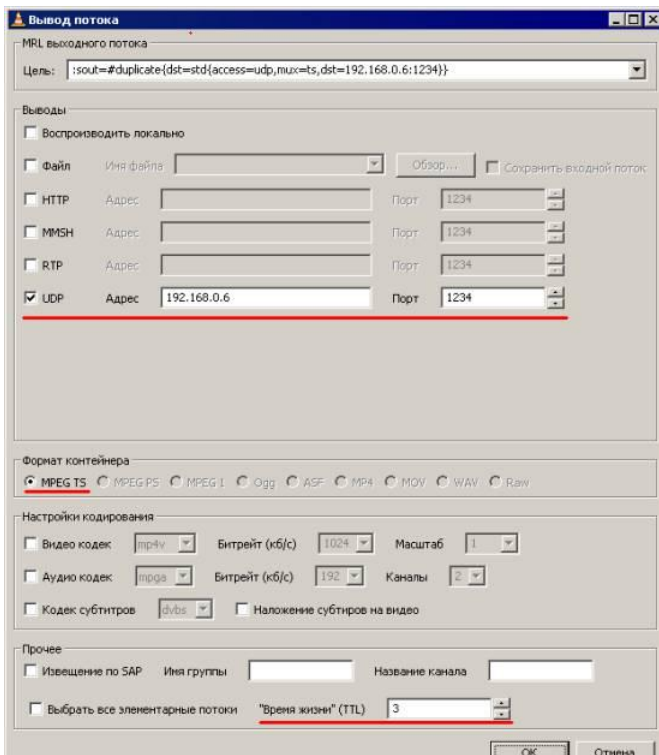


Рисунок 7. Настройки транслятора для вещания по протоколу UDP.

Взводим галочку "UDP" и указываем IP-адрес машины, на которую мы собираемся вещать (например, "192.168.0.6"), а также порт (по умолчанию "1234"), формат контейнера не предоставляет нам свободы выбора, т.к. UDP unicast работает только с MPEG TS (вообще-то, еще поддерживается и RAW, но в данном диалоговом окне он заблокирован и в этом есть свой резон, поскольку RAW - не лучший выбор для трансляции в реальном времени). Аудио/видео-кодеки в графе "Настройки кодирования" можно выбрать на свой вкус, а можно оставить их по умолчанию (сравнение качества кодеков - тема отдельного большого разговора, традиционно сопровождаемого яростными священными войнами, которые нам ни к чему).

Время жизни пакетов (TTL) зависит от количества узлов, через которые проходит транслируемый контент, и чтобы он не ушел чересчур далеко, это значение можно установить равным трем или даже одному. О строке "MRL выходного потока" можно не заботиться, программа сформирует ее за нас.

Остальные значения лучше пока не трогать, оставив их такими, какие они есть. Это слегка ухудшит функциональность транслятора, но на первых порах нам главное вообще разобраться, как его запускать!

Ок, нажимаем <ENTER> и возвращаемся в предыдущий диалог, в котором из всех немногочисленных опций обращает на себя параметр времени кэширования. В локальной сети на быстрых машинах особой разницы нет, но если возникнут проблемы и транслируемое видео станет двигаться судорожными рывками, имеет смысл взвести эту галочку.

Нажимаем "ОК" еще раз и VideoLAN начинает трансляцию на заданный узел (о чем говорит бегущая полоска, отображающая текущее положение видеофайла). Кстати говоря, UDP unicast обходит некоторые типы персональных брандмауэров типа SyGate Personal Firewall/Windows Firewall, которые его не видят, а потому и не блокируют (даже если мы этого хотим).

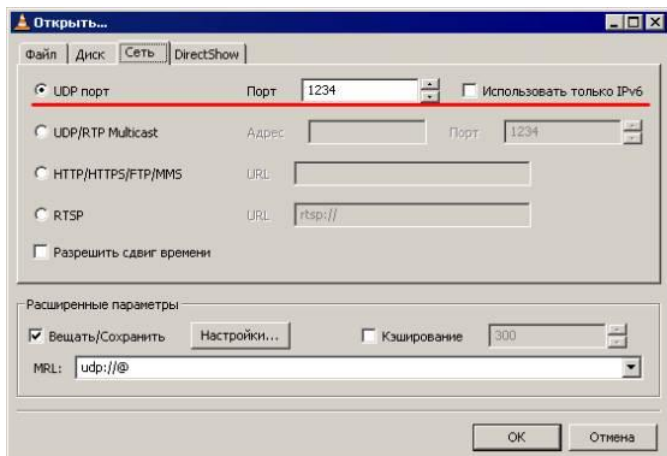


Рисунок 8. Настройки клиентского узла для приема медиа-потока, транслируемого по протоколу UDP.

Идем на машину с IP-адресом 192.168.0.6, запускаем на ней VideoLAN, заходим в меню "Файл", находим там пункт "Открыть URL" и в появившемся диалоговом окне (см. рис. 8) переводим радиокнопку в положение "UDP порт" с указанием порта-приемника, если он отличен от "1234", выбранного сервером по умолчанию. Вот и все! Нажимаем "ОК", поудобнее устраиваемся в кресле и наслаждаемся транслируемым видеоконтентом (см. рис. 9).



Рисунок 9. Просмотр потокового видео на клиентской стороне.

При желании (на клиентской стороне) можно нажать <CTRL-G> (Настройки -> Расширенный интерфейс) и поиграться яркостью, контрастностью, насыщенностью, эквалайзером и прочими "вкустностями" (см. рис. 10). На серверную сторону они не оказывают никакого влияния и потому крутить их на узле-трансляторе никакого смысла нет.

Трансляция через Web

Главным недостатком unicast-трансляции является невозможность вещания на произвольные узлы локальной/глобальной сети. Сервер должен иметь список IP-узлов для рассылки пакетов. Получателям знать же IP-адрес транслятора ни к чему. Им достаточно "помнить" назначенный UDP-порт, чтобы ловить трафик. Странная какая-то трансляция у нас получается... В обычной жизни все наоборот. Передатчик ничего не знает о приемнике (приемниках), а каждый из приемников в любой момент времени может настроиться на волну любого из многочисленных передатчиков и отключиться, если передача ему неинтересна.

Специально для реализации подобного способа общения, VideoLAN поддерживает трансляцию через Web по TCP/IP-протоколу. Возвращаясь к серверной стороне, говорим Файл -> Открыть файл -> Настройки и в уже знакомом нам диалоговом окне (см. рис. 11) сбрасываем (при желании) галочку "UDP" и взводим "HTTP". В поле "адрес" ничего вводить не нужно!!! Порт можно оставить в значении по умолчанию ("1234") или выбрать любой другой (например, "8080", чтобы поменьше привлекать к себе внимание). Также рекомендуется увеличить и значение TTL, особенно если мы

собираемся вещать в Интернет на далекие расстояния. Строку "MRL выходного потока", как и в прошлый раз, сформирует сама программа.

Обратите внимание, сколько доступных контейнеров теперь появилось: MPEG TSMPEG PS, MPEG 1, Ogg, ASF, MP4, MOV, WAV и RAW. Какой из них выбрать? Если все клиенты используют в качестве приемника программу VideoLAN, то особой разницы нет и лучше оставить контейнер по умолчанию (MPEG TS), если же планируется транслировать аудио/видеопоток на компьютеры, где кроме Windows и штатного медиаплеера ничего нет, лучше выбрать ASF, однако в таком случае следует позаботиться о совместимости с кодеками, поставляемыми вместе с Windows и в графе "видеокодек" выбрать что-то очень хорошо известное и проверенное временем (например, DIV3, WM1, WM2), аналогичным путем поступить и со звуком, в противном случае слушателям придется рыскать в поисках нужных кодеков перед началом воспроизведения контента. Впрочем, мы не будем вдаваться в дебри (не)совместимости различных кодеков, а поскорее нажмем на "ОК".

Бегущий ползунок линейки прогресса подтверждает, что вещание началось, даже если к нам еще никто не подключен (настройки персонального брандмауэра не имеют никакого значения, поскольку VideoLAN обходит все известные автору брандмауэры).

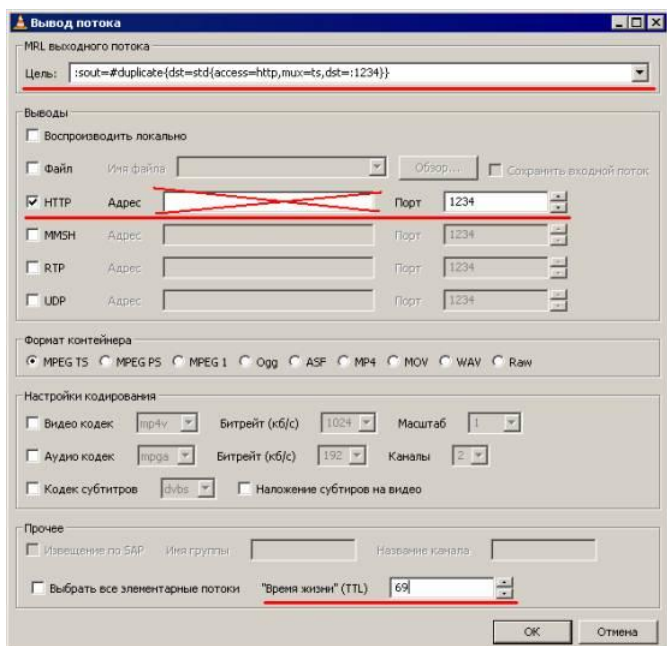


Рисунок 11. Настройки транслятора для вещания по протоколу HTTP.

На клиентском узле (который может находиться где угодно) запускаем VideoLAN, говорим Файл -> Открыть URL и в появившемся диалоговом окне перемещаем

радиокнопку к строке HTTP/HTTPS/FTP/MMS, в графе URL указываем адрес сервера вместе с портом (например, "http://192.168.0.1:1024") и нажмем "ОК".

И все это отлично работает с той лишь разницей, что теперь к серверу клиенты подключаются отовсюду и ему совершенно необязательно знать их IP-адреса!



Рисунок 12. Настройки клиентского узла для приема медиапотока, транслируемого по протоколу HTTP.

Интеграция с браузерами

Скачивать/устанавливать VideoLAN или запускать штатный media-player, чтобы ввести туда адрес сервера (вместе с портом!) - на это откажется далеко не каждый пользователь, особенно если media-player выдаст ошибку, потребовав установить некоторый кодек (и ведь, подлец, не скажет какой!).

Намного удобнее внедрить видеоплеер непосредственно в HTML-страничку. И VideoLAN позволяет сделать это! Для Mozilla, FireFox и Safari-браузеров выпущены специальные плагины, исходные тексты которых можно скачать непосредственно с сервера проекта VideoLAN: <http://developers.videolan.org/> и откомпилировать их.

Каркас страницы, принимающей UDP-потоки, выглядит следующим образом (полужирным шрифтом выделен адрес целевого узла, на который предполагается вести трансляцию):

```
<html>
<head><title>Demo of VLC mozilla plugin</title></head>

<body>
```

```
<h1>Demo of VLC mozilla plugin - Example 2</h1>
```

```
<embed type="application/x-vc-plugin"
```

```
  name="video2"
```

```
  autoplay="no" loop="no" hidden="yes"
```

```
  target="udp:@239.255.12.42" />
```

```
<br />
```

```
  <a href="javascript:;" onclick='document.video2.play()>Play video2</a>
```

```
  <a href="javascript:;" onclick='document.video2.stop()>Stop video2</a>
```

```
  <a href="javascript:;" onclick='document.video2.fullscreen()>Fullscreen</a>
```

```
</body>
```

```
</html>
```

Листинг 1. Каркас Web-страницы для приема потокового видео по протоколу UDP.

Прием HTTP-потоков осуществляется слегка иначе (полужирным шрифтом выделен адрес сервера трансляции и имя транслируемого файла):

```
<html>
```

```
<head><title>Demo of VLC mozilla plugin</title></head>
```

```
<body>
```

```
<h1>Demo of VLC mozilla plugin - Example 1</h1>
```

```
<embed type="application/x-vc-plugin"
```

```
  name="video1"
```

```
  autoplay="no" loop="yes" width="400" height="300"
```

```
  target="http://server.example.org/video1.vob" />
```

```
<br />
```

```
<a href="javascript:;" onclick='document.video1.play()>Play video1</a>
<a href="javascript:;" onclick='document.video1.pause()>Pause video1</a>
<a href="javascript:;" onclick='document.video1.stop()>Stop video1</a>
<a href="javascript:;" onclick='document.video1.fullscreen()>Fullscreen</a>

</body>
</html>
```

Листинг 2. Каркас Web-страницы для приема потокового видео по протоколу HTTP.

Как нетрудно видеть, в обоих случаях мы имеем дело с "видео по требованию", то есть позволяем пользователям самостоятельно выбирать, какой файл вещать и куда. Если подобная степень демократичности не устраивает администратора сервера, он может составить обыкновенную HTML-страничку, используя штатный ActiveX-компонент медиаплеера, но гарантий, что ее удастся воспроизвести на произвольной машине - никаких. Увы! Штатные Microsoft-кодеки не обеспечивает надлежащего уровня качества, а нестандартные - требуют установки (с правами администратора, разумеется).

Поэтому из двух зол приходится выбирать меньшее.

Сохранение потокового контента в файл

Большинству пользователей (особенно тех, кому безлимитный Интернет всего лишь снится) не слишком нравится перспектива просмотра аудио/видео контента без возможности его сохранения на локальный диск для последующего просмотра, особенно если сервер/канал тормозит или изображение идет рывками и насладиться качеством никак не получается в силу отсутствия такового.

VideoLAN позволяет сохранять потоковый контент на диск, автоматически конвертируя его в наиболее предпочтительный формат. Удобнее всего это делать через "Мастера" (Файл -> Мастер или <CTRL-W>). В появившемся диалоговом окне переводим радиокнопку в положение "Кодировать/Сохранить в файл" и говорим "Next". Радиокнопку "Выберите поток" оставляем в значении по умолчанию и нажимаем "Выберите", где выбираем вкладку "Сеть", а в ней протокол (и порт!), на который осуществляется вещание (транслятором может выступать как сам VideoLAN, запущенный на соседней машине, так и посторонний сервер).

Следующее диалоговое окно (см. рис. 13) предлагает нам выбрать формат сжатия аудио и видео. MPEG1, выставленный по умолчанию - наилучший выбор с точки зрения качества, но если мы не хотим жертвовать размером (а у MPEG1 размер ого-го), лучше остановиться на MPEG4 или других более продвинутых форматах, не забывая, впрочем, о совместимости с популярными кодеками.

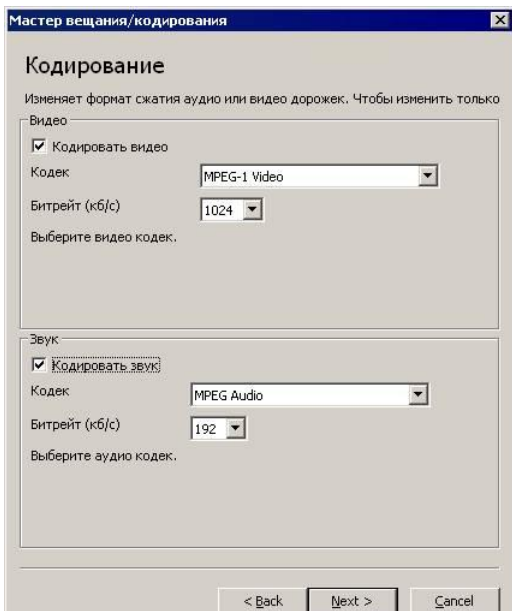


Рисунок 13. Выбор формата сохраняемого видео и аудио.

Наконец, нас спрашивают о формате контейнера. Контейнер - это то, куда ложится видеопоток, перемеженный с аудиопотоком. Наилучший выбор с точки зрения совместимости - это либо ASF, либо MPEG1. Все остальные контейнеры хоть и обладают определенными преимуществами, требуют обязательной установки дополнительного программного обеспечения (впрочем, к VideoLAN это не относится, так как он поддерживает их все).

Все! Последний "Next" выносит нас к строке ввода, запрашивающей имя файла, в который следует толкать весь принимаемый контент и после нажатия на "FINISH" начинается процесс грабежа (сохранение потокового контента может быть признано нелегальным, так что поосторожнее с награбленным!).

Заключение

На этом возможности программного комплекса VideoLAN не заканчиваются, а только-только начинают раскрываться. Для работы с ним необходимо не только помнить расположение пунктов меню, но и нехило разбираться в тонкостях многочисленных алгоритмов сжатия, знать - чем отличается один формат контейнера от другого, словом - быть настоящим профессионалом своего дела.

Как ни крути, а методом тыка такие вещи не осваиваются и воздвигнутые сервера трансляции оказываются либо недостаточно производительными, либо страдают хронической несовместимостью с клиентскими машинами, однако поскольку

VideoLAN включает в себя как серверную, так и клиентскую части, да и к тому же портирован под все платформы, наилучшим решением будет использование этой программы как на стороне транслятора, так и на приемной стороне.

Установка VLC

Бинарные файлы установки VLC имеются для многих, но не для всех поддерживаемых ОС. Если для нужной вам ОС нет бинарных файлов или если необходимо изменить установки по умолчанию, можно откомпилировать VLC из исходных кодов.

Debian GNU/Linux

Debian stable (woody)

Добавьте в /etc/apt/sources.list следующие строки:

```
deb http://www.videolan.org/pub/videolan/debian $(ARCH)/
deb-src http://www.videolan.org/pub/videolan/debian sources/
```

Затем для обычной установки наберите:

```
# apt-get update
# apt-get install gnome-vlc libdvdcss2
```

Debian unstable (sid)

Добавьте в /etc/apt/sources.list следующие строки:

```
deb http://www.videolan.org/pub/videolan/debian $(ARCH)/
deb-src http://www.videolan.org/pub/videolan/debian sources/
```

Затем для обычной установки наберите:

```
# apt-get update
# apt-get install wxvlc libdvdcss2
```

Debian testing (sarge)

Не следует использовать Debian testing, если вы не вполне определенно понимаете, что делаете. Практически невозможно поддерживать стабильную версию VLC для Debian testing и таких планов на будущее нет. Дополнительная информация о Debian testing представлена на [соответствующей странице](#).

Linux Mandrake

Существуют пакеты VLC для Mandrake 9.1 и Cooker.

Чтобы установить их, добавьте исходный код либо к Mandrake 9.1, либо к Cooker (для этого можно использовать [Easy urpmi](#)): contrib из основного дистрибутива и pfl (Penguin Liberation Front) из внешних дополнений.

Затем установите необходимые пакеты с помощью urpmi:

```
# urpmi libdvdcss2 libdvdisplay0 wxvlc vlc-plugin-a52 vlc-plugin-ogg vlc-plugin-mad
```

Linux Redhat

Загрузите RPM-пакет vlc и пакеты, перечисленные в разделе необходимые библиотеки и кодеки (остальные пакеты необязательны), со [страницы загрузки для Red Hat](#) и поместите их в один каталог.

Затем установите загруженные RPM пакеты:

```
# rpm -U *.rpm
```

При частичной установке пакетов, включенных в дистрибутив, может появиться запрос на дополнительную установку некоторых из них.

Самостоятельная компиляция из исходных кодов (для других ОС)

Описанный далее метод годится для любой поддерживаемой VLC Unix-системы, для которой нет готовых бинарных пакетов. Здесь будет рассказано как скомпилировать из исходных кодов и установить VLC вместе с необходимыми библиотеками.

Также можно скомпилировать VLC под Linux, если нужно изменить установки по умолчанию для поддерживаемых модулей.

Установка библиотек

Необходимы несколько библиотек:

- libdvbpsi (обязательно),
- mpeg2dec (обязательно),
- libdvdcss, если нужно читать закодированные DVD,

- libdvdplay, если нужна поддержка меню навигации DVD,
- a52dec, если нужно декодировать часто используемый в DVD звуковой формат AC3 (т.е. A52),
- ffmpeg, libmad, faad2, если нужно читать файлы MPEG-4/DivX,
- libogg и libvorbis, если нужно читать файлы Ogg Vorbis.

Загрузите библиотеки со [страницы исходных кодов VLC](#).

Для каждой библиотеки:

- распакуйте архив:

```
% tar xvzf library.tar.gz
```

- или

```
% tar xvjf library.tar.bz2
```

- настройте:

```
% cd library
% ./configure
```

- скомпилируйте и установите:

```
% make
# make install
```

Убедитесь, что файл конфигурации `/etc/ld.so.conf` содержит следующую строку: `/usr/local/lib`

Если ее нет, добавьте и запустите от имени пользователя root:

```
# ldconfig
```

Установка VLC

Загрузите исходный код последнего релиза - файл `vlc-version.tar.gz` со [страницы загрузки исходных кодов VLC](#). Распакуйте архив:

```
% tar xvzf vlc-version.tar.gz
% cd vlc-version
```

Чтобы получить список опций настройки наберите

```
% ./configure --help
```

Обратите внимание: все модули описаны в разделе *Модули* [Руководства пользователя VLC](#).

Примеры простых конфигураций:

- для базового VLC:

```
% ./configure
```

-

- если вместо интерфейса GTK необходим интерфейс Gnome (потребуется пакеты разработки Gnome):

```
% ./configure --enable-gnome
```

-

Затем нужно скомпилировать и установить:

```
% make
% su
Password: [пароль Root]
# make install
```

Обратите внимание, что установка (команда `make install`) необязательна. Можно запустить VLC из того каталога, в котором он был скомпилирован.

Перекомпилированные пакеты *VLC* доступны для большинства дистрибутивов. В системах, базирующихся на *APT*, все, что касается *VLC*, можно найти одной командой:

```
$sudo apt-cache search vlc
```

Список будет большим, но, к счастью, все устанавливать не нужно. Например, чтобы установить *VLC* в минимальной конфигурации в *KUbuntu*, достаточно ввести:

```
$ sudo apt-get update
$ sudo apt-get install vlc vlc-plugin-alsa
#или vlc-plugin-esd для Ubuntu
```

При этом будут установлены все необходимые пакеты, включая зависимости. По умолчанию используется графический интерфейс *wxvlc* (библиотека *wxWidgets/wxGTK*), но при желании можно добавить к списку *kvlc*, *qvlc* или *gnome-vlc* и установить интерфейсы для *KDE*, *Qt* или *GNOME* [имейте в виду, что некоторые из них уже официально не поддерживаются, - прим.ред.]. Кроме того, *VLC* доступен в виде подключаемого модуля *Mozilla mozilla-plugin-vlc*. Для *ALTLinux* команда установки выглядит так.

```
$ apt-get install vlc-normal
```

На странице <http://www.videolan.org/vlc/> приведены ссылки с информацией по установке *VLC* в основных дистрибутивах *GNU/Linux*.

Если вы будете устанавливать *VLC* вручную, вам понадобится еще несколько дополнительных библиотек:

- *libdvdcss* – если вы планируете читать зашифрованные DVD-диски (кстати, модуль не обращает внимание на зоны);
- *libdvdpplay* – если требуется поддержка DVD-меню;
- *libdvbps* – если вы планируете читать видеопотоки TS/DVB со спутников или цифрового телевидения;
- *a52dec* – при необходимости декодировать звук в формате AC3 (A52), принятом в мире DVD;
- *ffmpeg*, *libmad* и *faad2* – для чтения файлов MPEG 4/DivX;
- *libogg* и *libvorbis* – для поддержки звукового формата Ogg Vorbis;

- *lirc* – для управления с помощи пульта ДУ

Ключ на старт!

Запустить *VLC* можно как из основного меню рабочего стола, в которое встраивается ярлык программы, так и с консоли, командой *vlc*. В результате перед вами появится главное окно *VLC* (рис.1). Теперь можно просмотреть локальный видеофайл, создать поток или подключиться к уже имеющемуся. Разберем со всем этим по порядку.



Чтобы просмотреть файл, введите в командной строке:

```
$ vlc -vvv my_videofile.mpg
```

Программа сама подберет необходимый декодер. В случае неудачи его можно задать вручную параметром **--codec**.

```
$ vlc -vvv --codec ffmpeg my_videofile.mpg
```

Просмотр VCD или DVD начинается командой

```
$ vlc -vvv vcd:/dev/cdrom:@1:1
```

То же самое можно сделать и через меню. Просто откройте пункт **Файл** и выберите источник сигнала, файл, папку или диск.

Создание и просмотр потока

VLC (как и *VLS*) может создавать два типа видеопотоков: *unicast* и *multicast* (возможно применение обеих вариантов одновременно). В первом случае видеофайл разбивается на пакеты и отправляются по индивидуальному IP-адресу, указанному программе при запуске. Принимающая сторона просто считывает данные, поступающие на открытый для приема UDP-порт (по умолчанию используется 1234). Во втором случае сервер организует передачу на специальный групповой IP-адрес, с которого и считывают информацию многочисленные клиенты. Для создания *unicast*-потока следует ввести команду вроде:

```
# vlc -vvv videofile.avi --sout udp://192.168.0.42 --ttl 1
```

где **videofile.avi** – транслируемый видеофайл, **udp://192.168.0.42** – *unicast* IP-адрес (имя *unicast* IP-адрес (имя DNS) отдельного компьютера, **ttl** означает Time To Live т.е. время жизни пакета. Каждый маршрутизатор уменьшает значение TTL на единицу и, когда оно становится равным нулю, пакет уничтожается. Таким образом, установив значение TTL в 1, мы не допустим передачи трансляции в глобальную сеть.

Можно отправить в сеть и содержимое DVD диска.

```
# vlc -vvv dvd:/dev/dvd --sout udp://example.org
```

Для трансляции DVD или VCD под Unix потребуется право на запись в **/dev/dvd (/dev/cdrom)**. Обычно для этого необходимо включить пользователя, от имени которого запускается сервер, в группу **cdrom**. Более грубый, но и более простой подход состоит в передаче этого права всем и каждому:

```
# chmod 666 /dev/dvd
```

Аналогичным образом можно транслировать сигнал и с других источников.

Подключиться к *unicast*-потоку можно следующим образом.

```
# vlc -vvv udp:
```

В случае использования порта, отличного от 1234, необходимо явно указать его номер:

```
# vlc -vvv udp:@:9876
```

Забегая немного вперед, скажем, что при подключении к multicast- потоку необходимо добавить еще и IP-адрес.

```
# vlc -vvv udp:@multicast_address[:server_port]
```

Для ширококвещательных передач зарезервирован специальный диапазон IP-адресов: от 224.0.0.0 до 239.255.255.255. Для организации multicast-потока можно выбрать любой понравившийся адрес из этого диапазона и настроить его использование как на сервере, так и на клиенте. Для трансляции можно воспользоваться и стандартным для любой сети ширококвещательным адресом, как правило, заканчивающимся на 255, но в большой сети это может помешать нормальной работе некоторых служб.

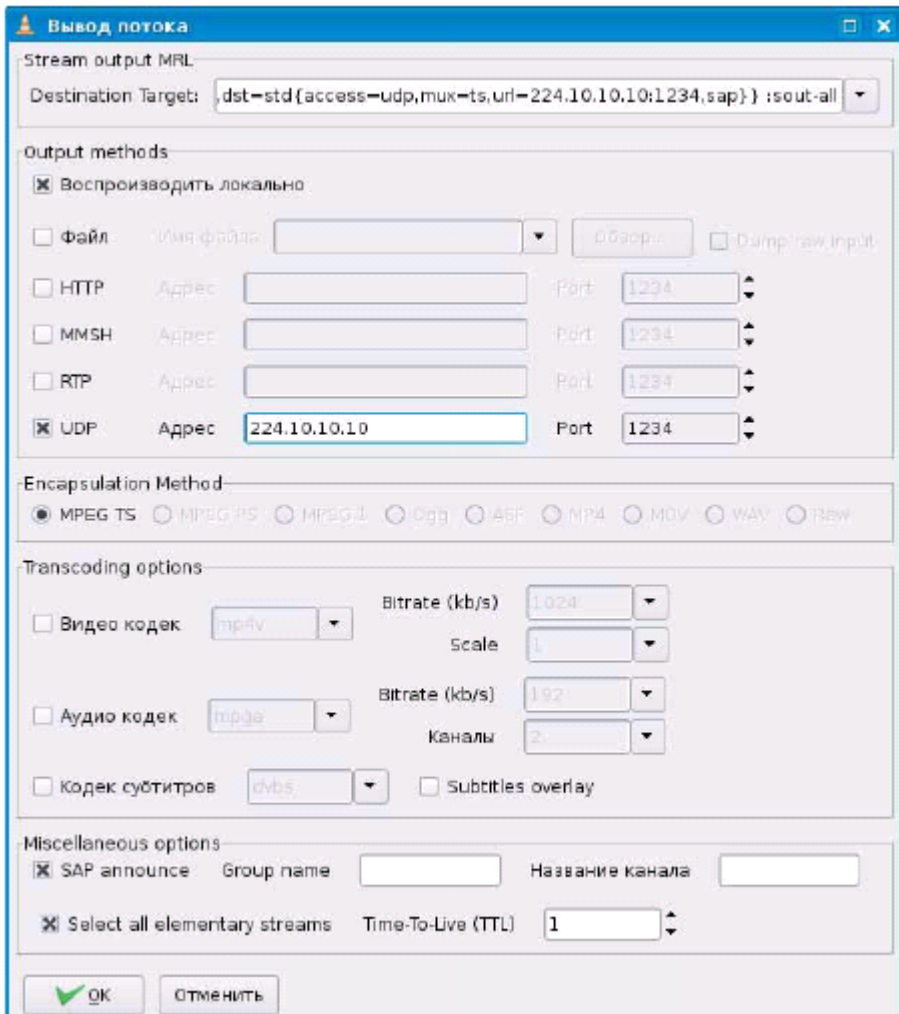
VideoLAN поддерживает и так называемое HTTP-вещание, когда клиенты подключаются к почти обычному Web-серверу, только в ответ на запрос получают не текстовый файл, а видеоданные. Сервер в таком случае запускается так.

```
# vlc -vvv input_stream --sout  
'#standard{access=http,mux=ogg,url=server.example.org:1234}'
```

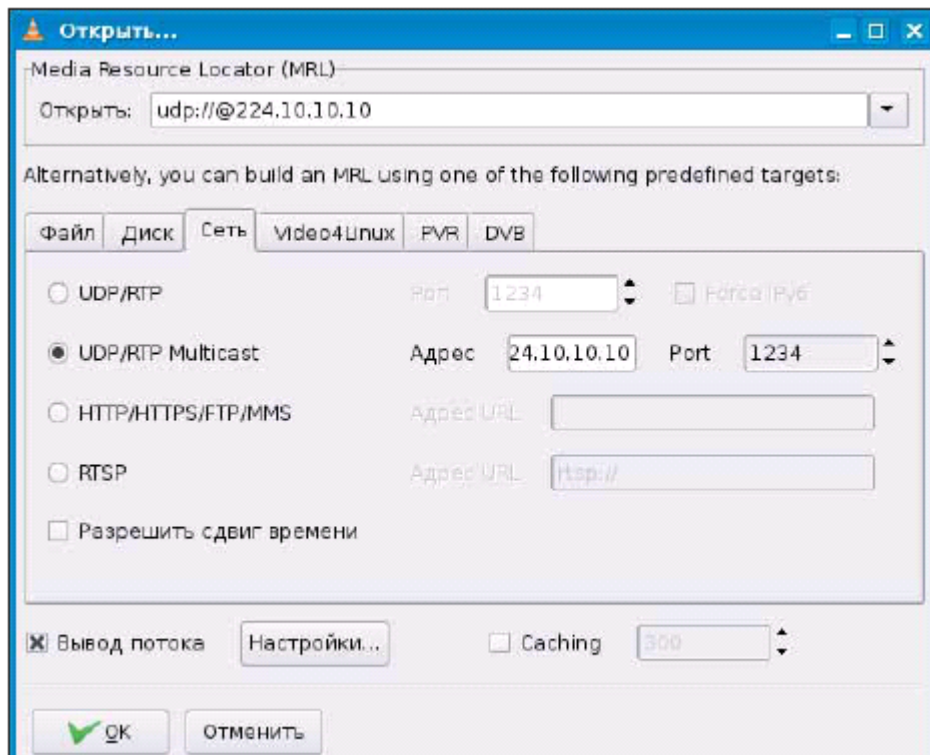
Клиенту (*VLC* или другой программе, например, *Xine*) достаточно указать стандартный URL, начинающийся с **http://**:

```
# vlc http://server.example.org:1234
```

Поток можно создать и с помощью меню. Для этого выберите источник (**Файл – Открыть файл**). В появившемся диалоговом окне отметьте нужный файл и поставьте флажок напротив **Вывод потока (Stream Output)**, затем нажмите **Настройка**. В открывшемся окне (рис.2) можно либо перечислить параметры командной строки в поле **Destination Target**, либо (что более удобно), воспользоваться группой элементов **Output Methods**.

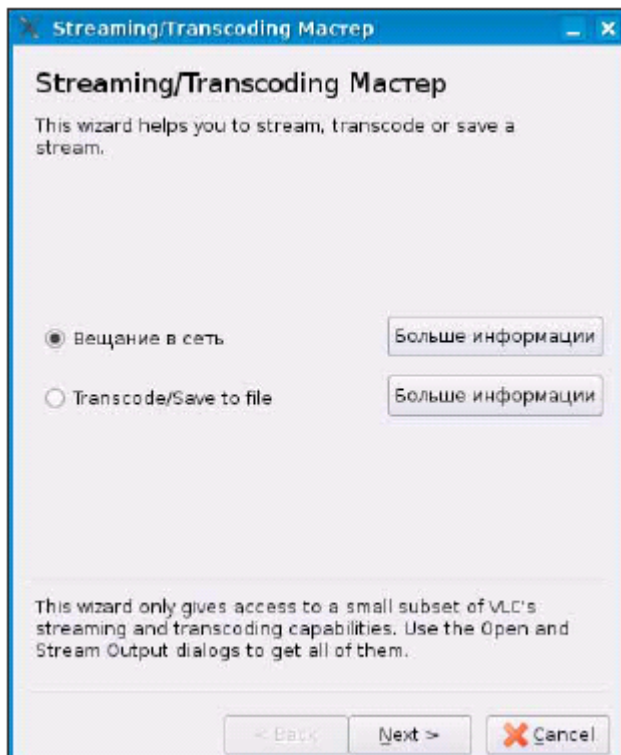


Например, для организации потока UDP multicast необходимо поставить флажок напротив пункта **UDP**. Параметры **Адрес** (224.10.10.10) и **порт** (1234) можно оставить без изменения. Активировав пункт **Воспроизводить локально**, вы сможете контролировать, что уходит в сеть, на компьютере-сервере. Для приема потока на другом компьютере используйте меню **Файл – Open Network Stream**, в появившемся окне укажите реквизиты multicast-сервера. Полученный поток можно тут же ретранслировать на другой адрес, как показано на рис.3.

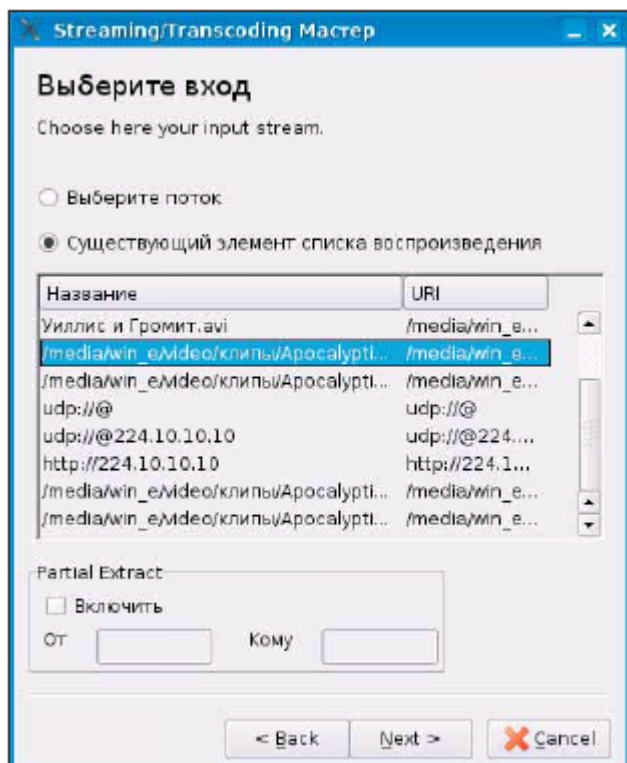


Существует и третий вариант – использование мастер-трансляции и перекодирования, которое вызывается из одноименного пункта меню **Файл**. В этом случае вам необходимо будет пройти всего пять шагов:

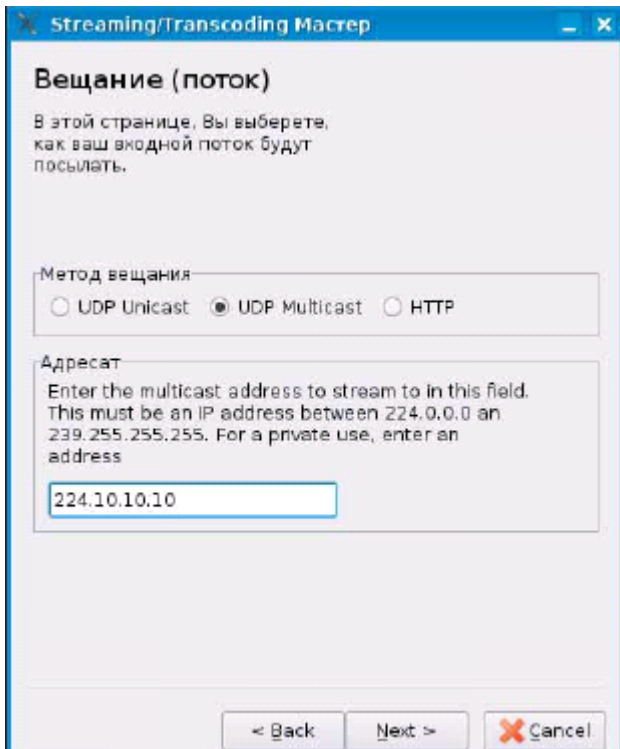
- выбор режима работы – **Вещание в сеть**



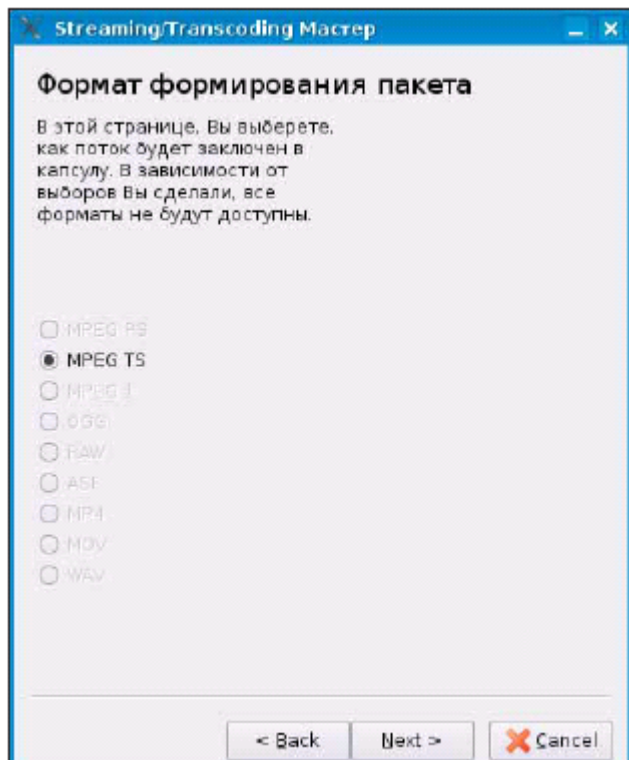
- выбор источника сигнала – **поток** или **список**



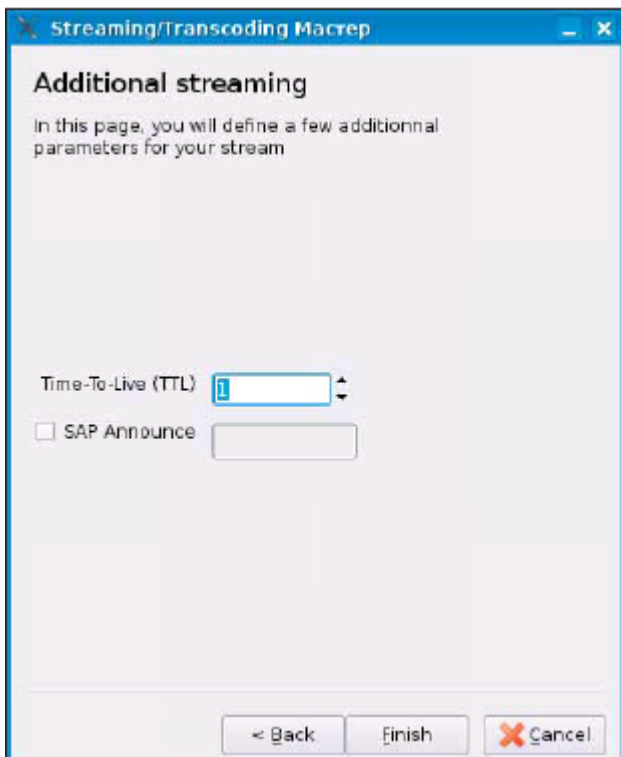
- выбор варианта выходного потока – **UDP unicast, multicast** или **HTTP**



- выбор формата форматирования пакета



- установки **TTL** и анонсов **SAP**



После нажатия на кнопку **Finish** можно подключаться к потоку и проверять результат. Как можно видеть, при использовании мастера некоторые опции оказываются недоступными.

Продвинутые возможности

На закуску поговорим о некоторых возможностях *VLC*, выходящих за пределы базовой функциональности «потокового вещателя». Например, очень полезная в хозяйстве вещь – transcoding. При этом исходный файл перекодируется в любой из поддерживаемых *VLC* форматов «на лету» и затем уже отправляется в странствие или сохраняется на жестком диске (опция **File** во вкладке **Stream Output**). Активировав пункты **Audio** и **Video codec** в поле **Transcoding Options**, можно установить кодек, битрейт и количество аудио-каналов. В командной строке все эти функции можно подключить через опцию **transcode**, если, конечно, вам нравятся конструкции вида:

```
# vlc -vvv dvd:/dev/dvd --sout '#transcode{vcodec= DIV3, acodec=vorb,v  
b=800,ab=128,channels=2,deinterlace}:standard{ access=udp,mux=ts,url=2
```

```
39.255.12.42,sap=TestStream}'
```

Подобным образом можно легко собрать очень неплохую и к тому же универсальную программу «ограбления» видеодиска и захвата видео. С помощью модуля **duplicate** есть возможность разделить входящий поток на два и затем каждый из них обработать по своему усмотрению.

```
# vlc -vvv videofile.avi --sout '#duplicate{dst=display,  
dst=`transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:  
duplicate{dst=standard{access=udp,mux=ts,url=192.168.1.2},  
dst=standard{access=udp,mux=ts,url=192.168.1.12}}}'
```

Все параметры, чтобы не вводить их каждый раз, можно прописать в конфигурационном файле. По умолчанию используется **\$HOME/.vlc/vlcrc**.

В итоге, после небольших манипуляций, мы получили свой сервер, транслирующий видеопоток. Ответы на все вопросы о возможностях программ проекта *VideoLAN* можно найти в документации, которой более чем достаточно на официальном сайте. **LXF**

Список контрольных вопросов

1. При помощи каких основных утилит обеспечивается безопасность в операционной системе Linux?
2. При помощи какой утилиты производится выполнение программ от своего и/или чужого имени?
3. Какой файл отвечает за настройки прав для утилиты *sudo*?
4. Каковы основные отличия удаленного доступа *telnet* от сетевого протокола *ssh*?
5. Назовите основные программные пакеты, реализующие *ssh* в различных операционных системах?
6. Какая утилита реализует механизм генерации открытого ключа *ssh*?
7. Каковы основные принципы работы файрвол?
8. Приведите примеры фильтрации пакетов при помощи *iptables*

Задания по лабораторной работе

1. Проверьте, установлена ли на Вашем рабочем месте утилита *sudo* и, при необходимости, установите ее.
2. Отредактируйте файл с конфигурацией *sudo*, дав своему пользователю, и запустите команду *ifconfig* от своего имени.
3. Проверьте, какими правами наделен Ваш пользователь?
4. Установите серверный пакет Open SSH и настройте его
5. Попробуйте зайти по протоколу *ssh* на рабочее место соседнего компьютера и откройте свое место для соседа по лабораторной работе.
6. Сгенерируйте и установите открытый ключ для доступа, попробуйте беспарольный вход.
7. Установите правило, блокирующее пакеты от соседнего рабочего места и проверьте работоспособность фильтра. По окончании теста отмените это правило.

Список литературы

1. *Уэли. М. и-др.*, Руководство по установке и использованию системы **Linux**. М.: ИЛКиРЛ, 1999
2. Александр Боковой, Александр Колотов, Александр Прокудин, Алексей Новодворский, Алексей Смирнов, Анатолий Якушин, Антон Бояршинов, Антон Ионов, Вадим Виниченко, Виталий Липатов, Георгий Курячий, Даниил Смирнов, Дмитрий Аленичев, Дмитрий Левин, Илья Трунин, Кирилл Маслинский, Максим Отставнов, Мэтт Уэлш, Олег Власенко, Сергей Турчин, Станислав Иевлев, Юрий Коновалов и другие; ALT Linux снаружи. ALT Linux внутри, ISBN 5-9706-0029-6, Издатель: ДМК пресс, 2006 г. Москва
3. Марк Г. Собелл, Практическое руководство по Red Hat Linux: Fedora Core и Red Hat Enterprise Linux, 2-е издание (Practical Guide to Red Hat Linux: Fedora Core and Red Hat Enterprise Linux), 1072 стр., с ил.; ISBN 5-8459-0841-8, 0-13-147024-8; формат 70x100/16; твердый переплет DVD-ROM; 2005, 2 кв.; Вильямс.
4. Разработка приложений в среде Linux. Программирование для linux, 2-е издание, Майкл К. Джонсон, Эрик В. Троан
5. Руководство администратора Linux. Установка и настройка. 2-е издание, Эви Немет, Гарт Снайдер, Трент Хейн
6. Linux. Библия пользователя, Кристофер Негус
7. Linux для чайников, 6-е издание, Ди-Анн Лебланк
8. Разработка ядра Linux, 2-е издание, Роберт Лав
9. Библиотека Qt 4. Программирование прикладных приложений в среде Linux., Чеботарев Арсений Викторович
10. Red Hat Linux Fedora 4. Полное руководство, Пол Хадсон, Эндрю Хадсон, Билл Болл, Хойт Дафф
11. Искусство программирования для Unix, Эрик С. Реймонд
12. Linux для "чайников", 5-е издание, Ди-Анн Лебланк
13. Red Hat Linux. Секреты профессионала, Наба Баркакати
14. Использование Linux, Apache, MySQL и PHP для разработки Web-приложений, Джеймс Ли, Brent Уэр
15. Секреты хакеров. Безопасность сетей - готовые решения, 4-е издание, Стюарт Мак-Клар, Джоэл Скембрей, Джордж Курц
16. FreeBSD: полный справочник., Родерик Смит

17. Секреты хакеров. Безопасность Linux — готовые решения, 2-е издание, Брайан Хатч, Джеймс Ли, Джордж Курц
18. Red Hat Linux 8. Библия пользователя, Кристофер Негус
19. Серверы Linux. Самоучитель, Птицын Константин Александрович
20. Безопасность Linux, 2-е издание, Скотт Манн, Эллен Л. Митчелл, Митчелл Крелл
21. Сетевые средства Linux, Родерик Смит
22. Руководство администратора Linux, Эви Немет, Гарт Снайдер, Трент Хейн
23. Сети TCP/IP, том 3. Разработка приложений типа клиент/сервер для Linux/POSIX, Дуглас Камер, Дэвид Л. Стивенс
24. Секреты хакеров. Безопасность Linux — готовые решения, Брайан Хатч, Джеймс Ли, Джордж Курц
25. Программирование для Linux. Профессиональный подход, Марк Митчелл, Джеффри Оулдем, Алекс Самьюэл
26. Использование Linux, 6-е издание. Специальное издание, Дэвид Бендел, Роберт Нейпир
27. Создание сетевых приложений в среде Linux, Шон Уолтон
28. Освой самостоятельно Linux за 24 часа, 3-е издание,
29. Система электронной почты на основе Linux. Руководство администратора, Ричард Блам
30. Системное администрирование Linux, М. Карлинг, Стефан Деглер, Джеймс Деннис

Учебное издание

Средства он-лайн общения в Linux

Методические указания

Составитель: Сухов Андрей Михайлович

**Изд-во Самарского государственного
аэрокосмического университета.
443086 Самара, Московское шоссе, 34.**