

# потокковое аудио/видео вещание с VideoLAN

крис касперски, no-email

прошло то время, когда для открытия собственной радио/теле-станции требовалось приобретать дорогостоящее оборудование, а так же 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 медиаплеера, поддерживающего множество входных форматов с возможностью записи чужого потоккового медиаконтента на жесткий диск

## введение

Какой смысл в потокковом 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! Потокковое 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 файло-обменные 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 онлайнное 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 потокковое аудио/видео намного легче "врезать" рекламу или прочие вставки типа "breaking news", да и квалификация среднестатистического пользователя не позволяет сохранять потокковый контент на диск, что очень нравится держателям авторских прав и прочим медиамагнатам.

Словом, существуют тысячи причин, чтобы воздвигнуть сервер потоккового аудио/видео, вещающий в пределах локальной сети или даже охватывающей весь Интернет. Для потоккового 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).

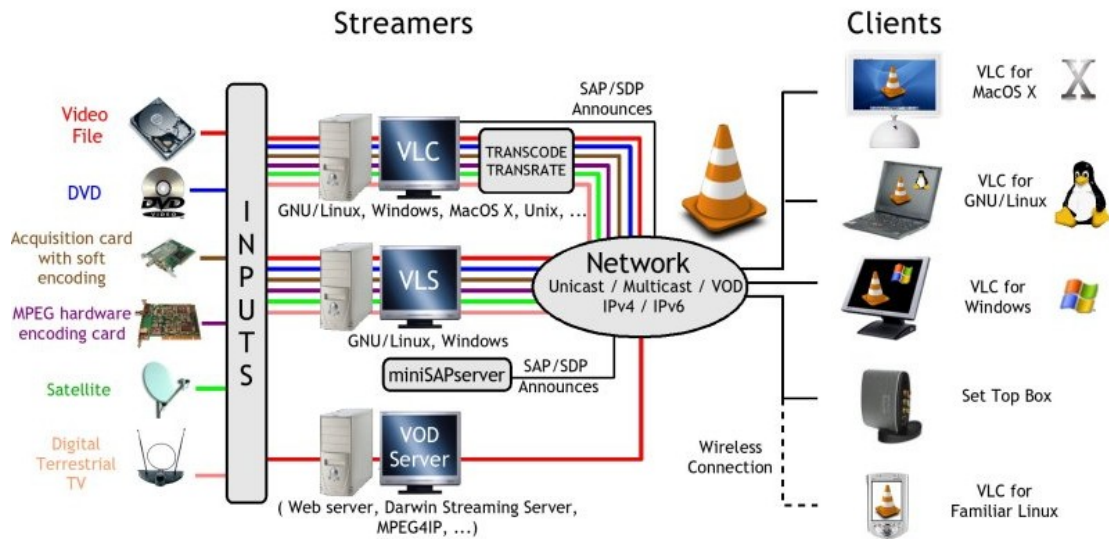


Рисунок 1 потоковое вещание в локальных/глобальных сетях

## 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, Yory/Linury 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	WxWidgets	Cocoa	WxWidgets	Native	WxWidgets	GPE
Qt 4 <a href="#">11</a>	✓	✗	✓	-	?	✗
Skins	✓	✗	✓	✗	✓	✗
Web	✓	✓	✓	✓	✓	✓
Telnet	✓	✓	✓	✓	✓	✓
Command line	✓	✓	✓	✓	✓	✓
Infrared	✗	✗	✓	✗	✗	✗

## Miscellaneous

						
SAP/SDP announces	✓	✓	✓	✗	✓	✗
Bonjour protocol	✗	✓	✓	?	?	?
Mozilla/Firefox plugin	✓	✓	✓	✗	✓	✗
ActiveX plugin	✓	-	-	-	-	-
SVCD Menus	⊖	✗	⊖	✗	⊖	✗
Localization	✓	✓	✓	✓	✓	✓
CD-Text <a href="#">12</a>	✓	✗	✓	✗	⊖	-
CDDDB CD info	✓	✓	✓	✗	⊖	-
IGMPv3 <a href="#">13</a>	✓	✗	✓	✗	✓	✓
IPv6 <a href="#">13</a>	✓	✓	✓	✗	✓	✓
MLDv2 <a href="#">13</a>	✓	✗	✓	✗	✓	✓
CPU acceleration <a href="#">14</a>	✓	✓	✓	✓	✓ <a href="#">15</a>	✗

**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). Так же (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/TELTEL и даже плагины для некоторых популярных браузеров.

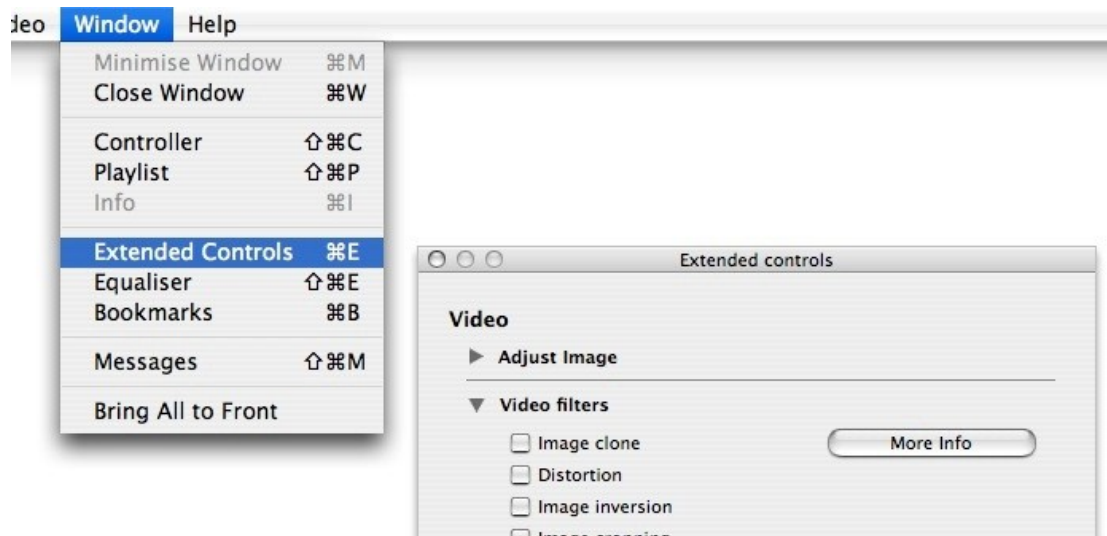


Рисунок 3 графический интерфейс программы VideoLAN



Рисунок 4 текстовой NCURSES интерфейс

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

	UDP	RTP	HTTP	MMSH	File
PS	✗	✓	✓	✗	✓
TS	✓	✓	✓	✗	✓
Ogg	✗	✗	✓	✗	✓
ASF	✗	✗	✓	✓	✓
MP4	✗	✗	✗	✗	✓
QuickTime	✗	✗	✗	✗	✓
Raw	✓	✓	✓	✗	✓
MPJPEG	✗	✗	SVN only	✗	SVN only

**Рисунок 5 допустимые комбинации протоколов трансляции с контейнерами, в которых 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/Linux;
- 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;

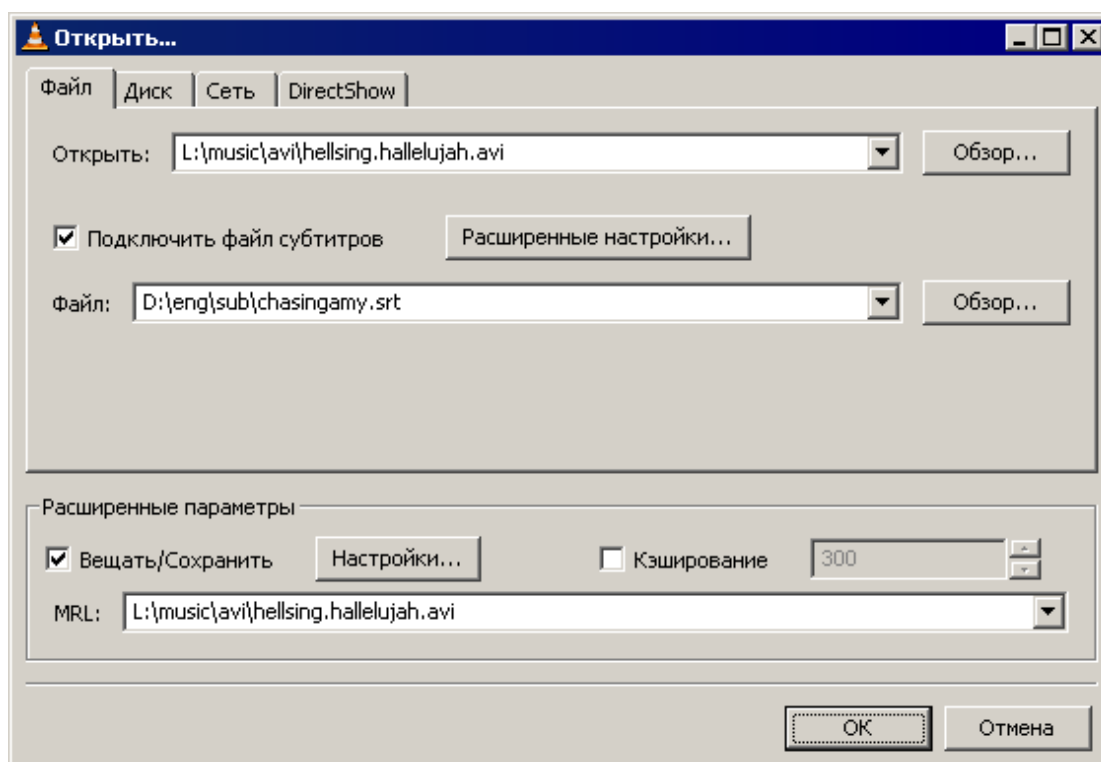
Поскольку, VideoLAN входит в набор портов для FreeBSD (любимой xBSD системы автора!), то сборка под нее проходит без единого слова нареканий. Достаточно набрать команду: `"# cd /usr/ports/multimedia/vlc && make install clean"` и немного подождать. Остальные системы в этом плане несколько менее предсказуемы 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, соответствующей горячей клавише <CTRL-W>), однако, он содержит несколько подводных камней, кроме того, ограничивает наши возможности по кручению тонких настроек, а потому мы пойдём другим путем.

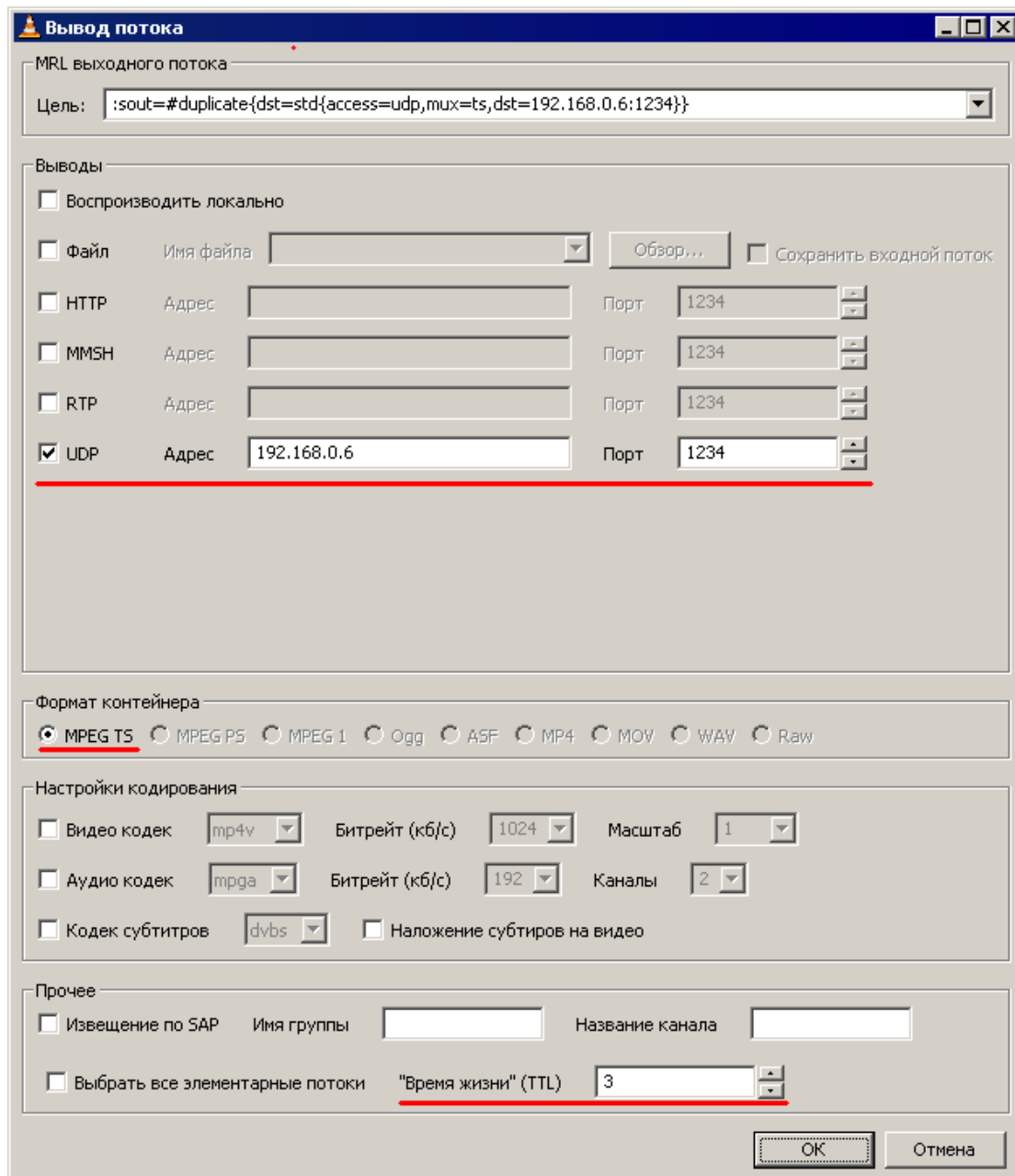
В меню "Файл" выбираем пункт "Открыть файл" <CTRL-F>, если хотим транслировать один или несколько mp3/avi/mpeg-файлов; "Открыть каталог" <CTRL-E>, чтобы одним махом выделить все содержащиеся в нем файлы; "Открыть диск" <CTRL-D> для трансляции контента непосредственно с DVD/VCD/Audio CD (правда, несмотря 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 пугливых!!!



**Рисунок 7** настройки транслятора для вещания по протоколу UDP

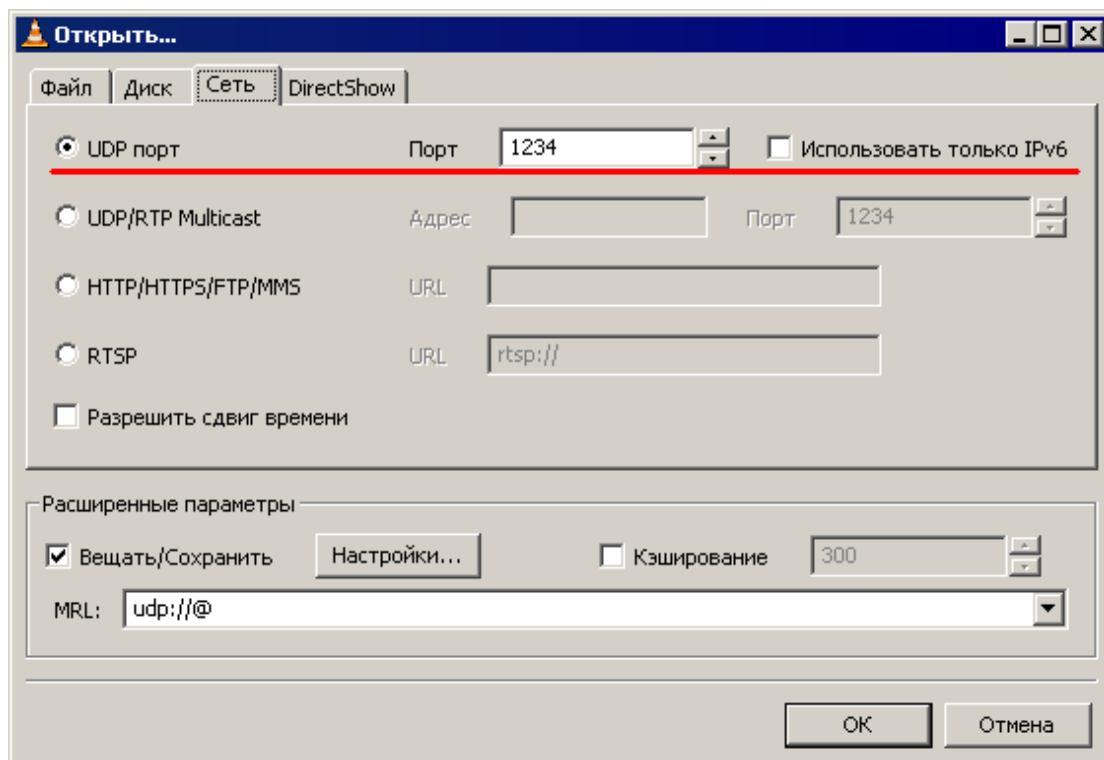
Взводим галочку "UDP" и указываем IP-адрес машины на которую мы собираемся вещать (например, "192.168.0.6"), а так же порт (по умолчанию "1234"), формат контейнера не предоставляет нам свободы выбора, т. к. UDP unicast работает только с MPEG TS (вообще-то, еще поддерживается и RAW, но в данном диалоговом 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 через чур далеко это значение можно установить равному 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 себе параметр времени кэширования. В локальной сети на быстрых машинах особой разницы нет, то 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 видео-файла). Кстати говоря, UDP unicast обходит некоторые типы персональных брандмауэров типа например SyGate Personal Firewall/Windows Firewall, которые его не видят, а потому и не блокируют (даже если мы этого хотим).



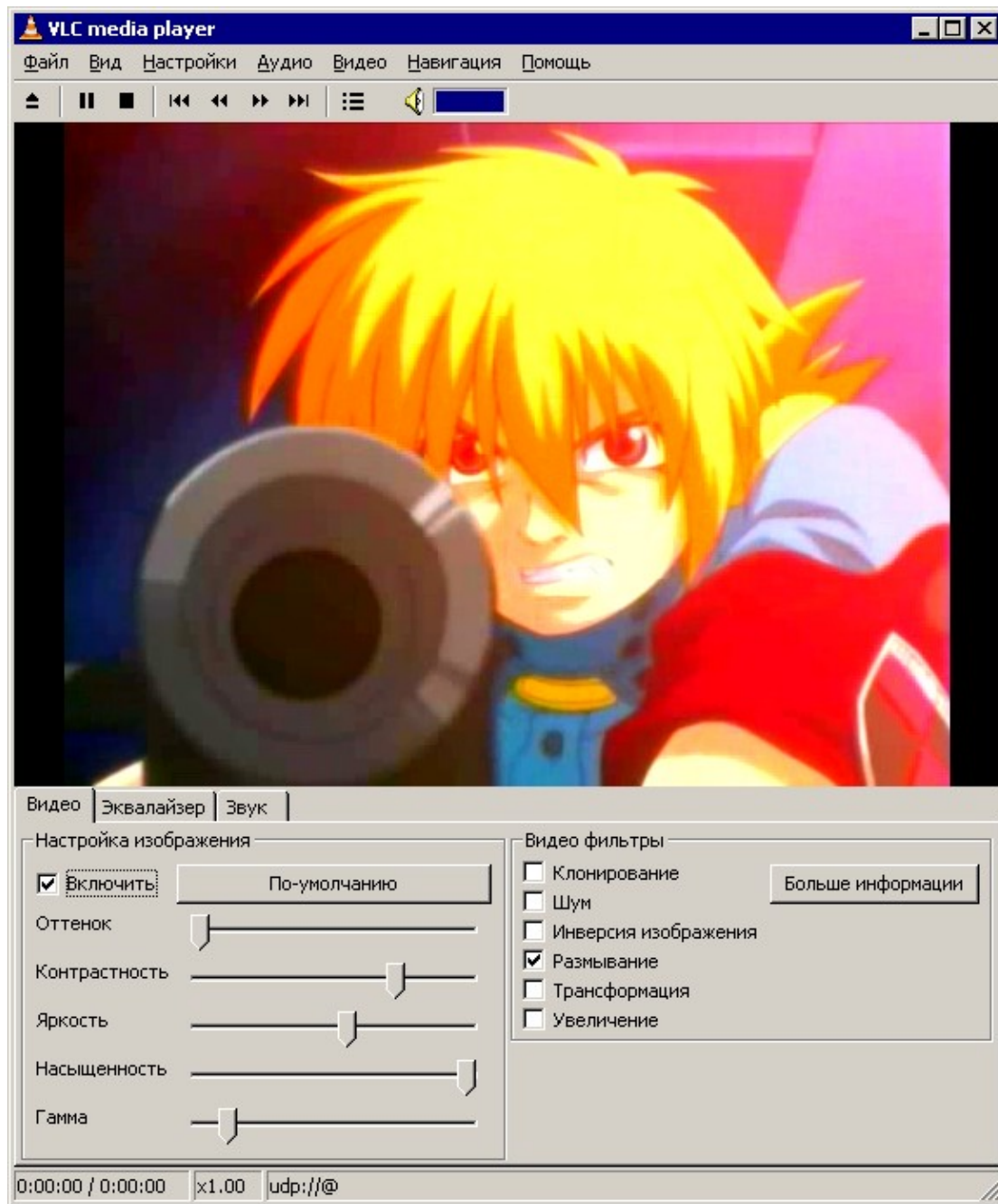
**Рисунок 8** настройки клиентского узла для приема медиа-потока, транслируемого по протоколу UDP

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 радио-кнопку 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 проиграться яркостью, константностью, 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.



**Рисунок 10 на клиентской стороне можно свободно менять основные параметры изображения (яркость, контрастность, насыщенность...) а так же накладывать многочисленные фильтры**

## ***трансляция через WEB***

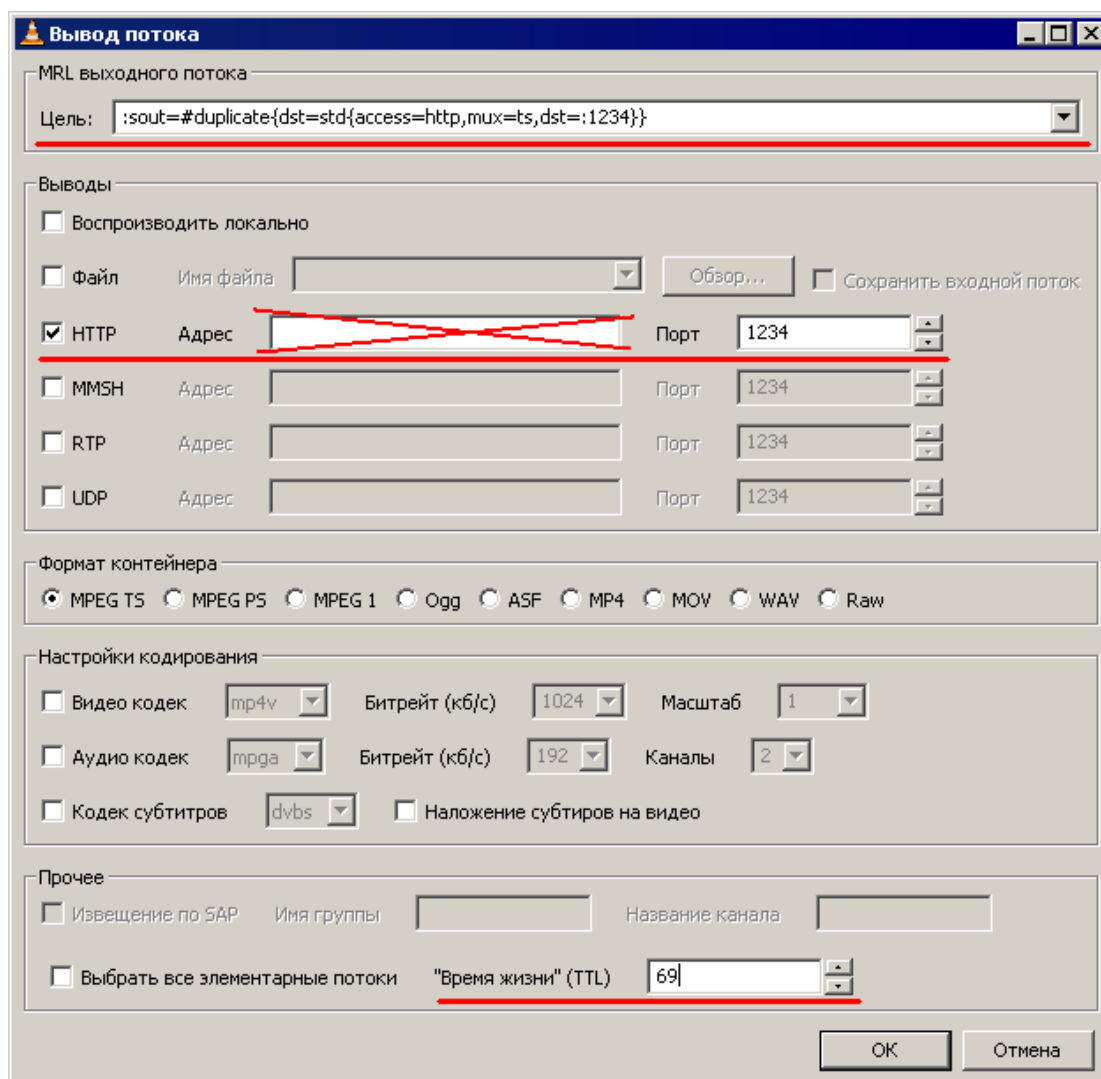
Главным недостатком unicast-трансляции является невозможность вещания на произвольные узлы локальной/глобальной сети. Сервер должен иметь список IP-узлов для рассылки пакетов. Получателям знать же IP-адрес транслятора ни к чему. Им достаточно "помнить" назначенный UDP-порт, чтобы ловить трафик. Странная 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). Так же 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 TS/MPEG 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/видео-поток 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 "видео-кодек" 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 "OK".

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 радио-кнопку 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!

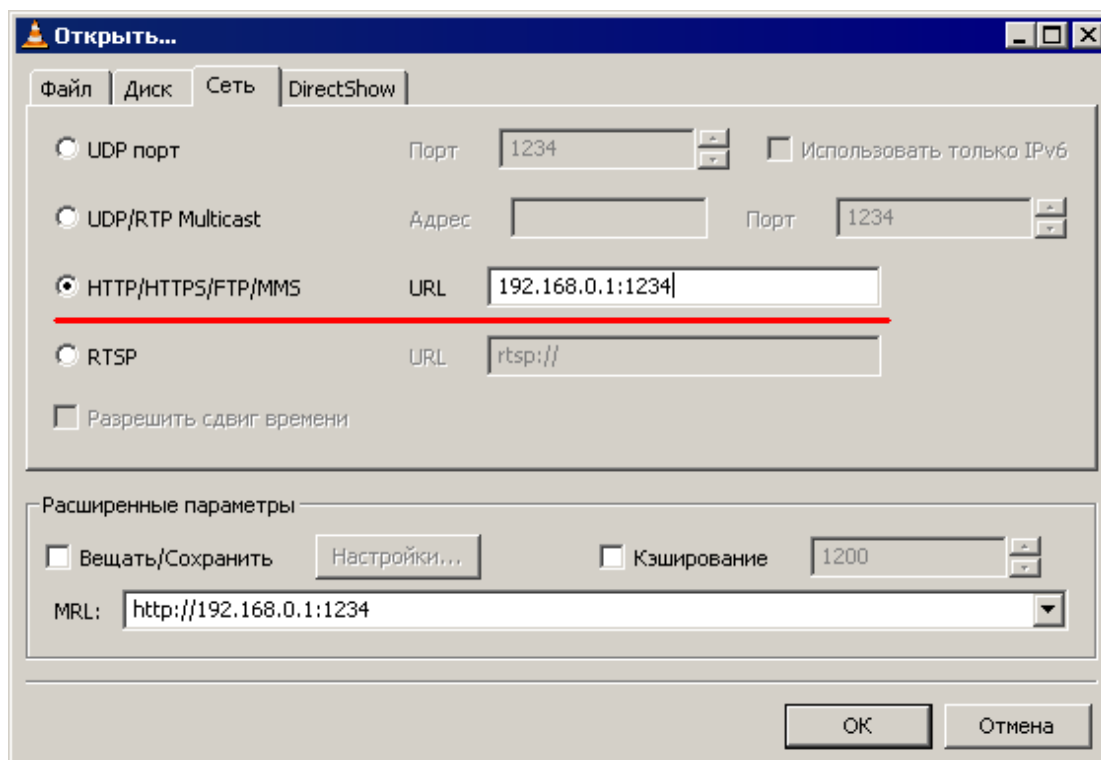


Рисунок 12 настройки клиентского узла для приема медиа-потока, транслируемого по протоколу HTTP

## интеграция с браузерами

Скачивать/устанавливать VideoLAN или запускать штатный media-player, чтобы ввести туда адрес сервера (вместе с портом!) на это отважится 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, Fire-Fox 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, принимающий 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-vlc-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-vlc-plugin"
name="videol"
autoplay="no" loop="yes" width="400" height="300"
target="http://server.example.org/videol.vob"/>
<br />
<a href="javascript:;" onclick='document.videol.play()'>Play videol</a>
<a href="javascript:;" onclick='document.videol.pause()'>Pause videol</a>
<a href="javascript:;" onclick='document.videol.stop()'>Stop videol</a>
<a href="javascript:;" onclick='document.videol.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 устает администратора сервера, он может составить обыкновенную HTML-страничку, используя штатный ActiveX-компонент медиа-плеера, но гарантий, что ее удастся воспроизвести на произвольной машине — никаких. Увы! Штатные Microsoft-кодеки не обеспечивают надлежащего уровня качества, а нештатные — требуют установки (с правами администратора, разумеется).

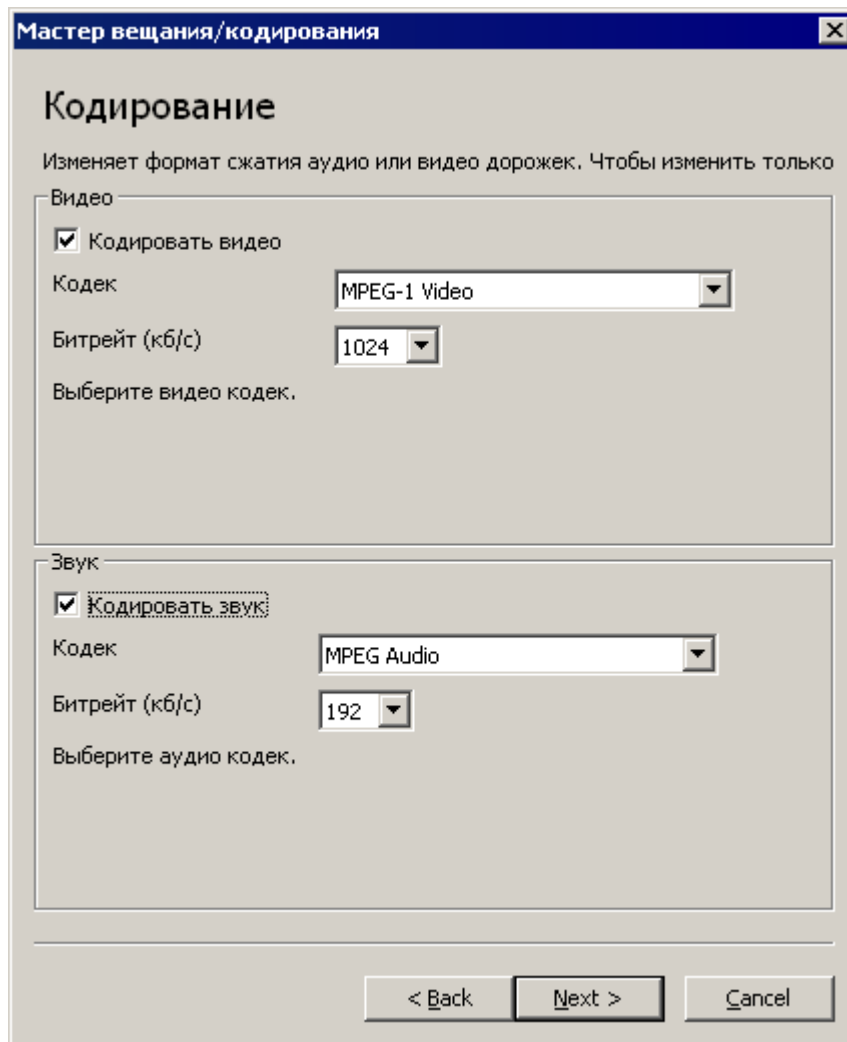
Поэтому, из двух зол приходится выбирать меньшее.

## 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 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 радио-кнопку в положение "Кодировать/Сохранить в файл" и говорим "Next". Радио-кнопку "Выберете поток" оставляем в значении по умолчанию и нажимаем "Выберете", 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-го), лучше остановится 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 видео-поток, перемеженный с аудио-поток. 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 всех).

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.