

# Руководство пользователя пакета *Udpху*

Данное руководство составлено с целью разъяснить принципы и основные алгоритмы работы приложений, входящих в состав пакета *udpху* (в настоящее время эти приложения: *udpху*, *udpхrec*). Руководство также ставит своей целью более подробно описать значение параметров командной строки и дать примеры их использования.

## Приложение *udpху*

*Udpху* - серверное приложение, обладающее при определённом образом заданных параметрах атрибутами *UNIX daemon*-а. Основная задача *udpху* заключается в передаче данных, считанных из мультикаст-канала (рассылающего данные подписчикам по протоколу *UDP*), в клиентское соединение, работающее в протоколе *TCP*. Таким образом, клиент, не имея возможности работать с протоколом *UDP*, может послать запрос *udpху*, установить *TCP* соединение и работать с данными, полученными из указанного (в изначальном запросе) мультикаст-канала.

Основным сценарием использования *udpху* является воспроизведение видео или аудио потоков, транслирующихся в локальной сети. Не все видео и аудио плееры (приложения) поддерживают считывание информации непосредственно из мультикаст-канала, но большинство обладает возможностью читать видео- или аудио-поток через *HTTP* запрос (по протоколу *TCP*). Клиент - плеер - посылает *HTTP (GET)* запрос *udpху* в виде *URL* с параметрами, идентифицирующими источник данных (адрес и порт мультикаст-канала); *udpху* устанавливает "подписывается" на соответствующий канал и начинает считывать из него данные, которые затем передаются (по установленному *TCP* соединению) клиенту. Приём и передача данных происходят в цикле до тех пор, пока одна из сторон не прервёт соединение: источник перестанет передавать данные или же клиент со своей стороны прекратит их приём.

## Протоколы передачи и форматы данных

В настоящее время *udpху* поддерживает следующие протоколы передачи аудио- и видео-данных:

1. *TS = MPEG-TS*;
2. *RTP = MPEG-TS* через *RTP*.

Приложение идентифицирует поток по первому полученному сообщению. Тип потока определяет дальнейший способ обработки данных: в случае *TS* никакой дополнительной обработки не производится; если же данные передаются по *RTP*, *udpху* "конвертирует"

поток в *TS* (вычлняя из *RTP* пакетов данные *MPEG-TS*). Таким образом, во всех случаях клиенту выдаётся поток в формате *TS*. В случае, если *udpxu* не может идентифицировать тип потока, выдаётся ошибка и соединение с клиентом закрывается.

## HTTP запросы к *udpxu*

*udpxu* принимает следующие типы запросов от клиентов по *HTTP*:

### 1. Запрос на соединение:

**`http://{address}:{port}/{protocol}/{channel_addr}:{channel_port}`**, где

`{address}`, `{port}` - адрес и порт, на которых *udpxu* отслеживает ("слушает") запросы клиентов (см. параметры командной строки **-a** и **-p**);

`{protocol}` ::= *udp* | *rtp* - формат потока; при указании *RTP* *udpxu* будет сразу считать протоколом передаваемого потока *RTP*, при указании *UDP* *udpxu* самостоятельно попытается определить протокол;

`{channel_addr}` - IP адрес (мультикаст-) канала данных;

`{channel_port}` - порт канала данных;

### 2. Статус приложения

**`http://{address}:{port}/status`** - по данной команде *udpxu* возвращает клиенту (*HTML*) страницу статуса приложения, отображающую в том числе статус клиентов и статистику скорости передачи данных клиентов (см. опцию **-S**).

### 3. Перезапуск

**`http://{address}:{port}/restart`** - по данной команде *udpxu* осуществляет перезапуск: закрывает соединения со всеми клиентами и переходит в состояние ожидания запросов.

## Параметры командной строки *udpxu*

**-v** - включить режим детального вывода в журнал [выключен по умолчанию]

При включённом режиме детального вывода приложение будет выводить в журнал (см. параметр **-l**) отладочную информацию. **Важно:** вывод отладочной информации невозможен для исполняемых файлов *udpxu*, собранных в режиме *lean*. (Режим указан при запуске

приложения без параметров сразу после номера сборки, например: *udpxu 1.0\_Chipmunk (build 8) standard*, где режим сборки - "*standard*".

**-S** - передавать статистику о клиентских соединениях [по умолчанию - не передавать]

Приложение будет накапливать и отображать (по *HTTP*-интерфейсу при запросе *status*) данные о средней скорости чтения и записи данных на потоках клиентов. Например, при наличии трёх клиентов, страница статуса *udpxu* покажет среднюю пропускную способность (в килобайтах в секунду) для каждого из клиентов.

**-T** - в случае запуска администратором (*root*) не пытаться работать в режиме сервера [по умолчанию - работать в режиме сервера (*daemon*), если процесс запущен пользователем с административными правами]

При запуске непривилегированным пользователем *udpxu* будет работать в окне терминала и реагировать на интерактивные команды терминальной сессии, такие как выход в фоновый режим, прерывание по *Ctrl-C* и т.д. При запуске администратором (*root*) или пользователем с привилегиями администратора ( $0 == $EUID$ ) приложение по умолчанию попытается работать в режиме сервера (*daemon*). Задействовав переключатель **-T**, пользователь может заставить *udpxu* при запуске администратором (*root*) работать с текущим терминалом.

**-a** - указать адрес (*IPv4*) или имя интерфейса для (*HTTP*) запросов к приложению [0.0.0.0 - по умолчанию]

**-p** - указать *TCP* порт для (*HTTP*) запросов к приложению (обязательный параметр)

**-m** - указать адрес (*IPv4*) или имя интерфейса мультикаст-каналов [0.0.0.0 - по умолчанию]

**-c** - максимальное количество клиентов, обслуживаемых одновременно [по умолчанию - 3 клиента, потолок - 16]

*udpxu* способно обслуживать не более обозначенного количества клиентов одновременно; приложение не даст установить соединение в превышение указанной квоты. Переопределив константу *MAX\_CLIENT\_COUNT* (в файле *uopt.h*), *udpxu* может быть скомпилировано с иным потолком соединений.

**-l** - записывать журнал приложения в указанный файл [по умолчанию - *stderr*]

*udpxu* будет вести запись журнала в указанный файл; записи будут добавляться или же файл будет создан заново. В случае, если приложение будет запущено пользователем с правами администратора в режиме детального вывода (параметр **-v**), но файл не будет указан, *udpxu* выдаст ошибку.

**-B** - указать размер буфера для входящих данных [65536 байт по умолчанию]

Параметр заставляет *udpxu* накапливать входящие данные в буфере указанного размера прежде, чем передать их в соединение клиента. Данные читаются порциями, обозначаемыми как "сообщения" (*messages*). Таким образом, до тех пор, пока буфер указанного размера не будет заполнен должным количеством сообщений (определяемым параметром **-R**) или же (с момента получения первого пакета) не пройдет время, обозначенное параметром **-H**, данные не будут переданы клиенту.

**-R** - указать максимальное количество сообщений, накапливаемых в буфере [по умолчанию -1, т.е. максимальное количество, вмещающееся в буфер (см. параметр **-B**) ]

Параметр определяет, сколько сообщений должно быть в буфере прежде, чем он будет отправлен клиенту. Значение "1" (единица) ограничивают буфер одним сообщением, что означает, что все приходящие данные сразу же отправляются клиенту, и накапливания не происходит. Значение "-1" (минус единица) снимают ограничение с буфера: сообщения накапливаются до полного насыщения буфера. (В этом случае *udpxu* прерывает накопление, если в буфере осталось меньше места, чем то, что занимает последняя считанная запись.)

**Важно:** в сборках *udpxu* версии 0.1 (т.е. до "*Chipmunk*") данный параметр выставлялся по умолчанию в "1", хотя и обозначался (ошибочно) в подсказке как "-1" (минус единица); таким образом данные в буфере по умолчанию не накапливались. Таким образом, переход на новую версию может означать и изменение поведения *udpxu*. **Для того, чтобы *udpxu* вело себя "по-старому", следует выставлять значение параметра **-R** в -1.**

**Важно:** при достаточно низкой скорости чтения входного потока и значительном размере буфера, клиент может посчитать время простоя (когда ему не поступают данные) чрезмерным и закрыть соединение или же отказывать в приёме сообщения. (В этом случае в журнале отладки можно видеть периодически появляющиеся записи: "*write\_buf: write: Resource temporarily unavailable*").

**-H** - указать максимальное время (в секундах) накопления сообщений в буфере (см. параметры **-B**, **-R**) [по умолчанию - 4 секунды]

Приложение засекает время "прибытия" первого пакета (из серии), записываемого в буфер. После чтения каждого пакета, разница во времени сравнивается с указанной (в секундах) величиной: если пройденное время превышает заданный промежуток, накопленные в буфере данные передаются клиенту.

**-n** - смещение приоритета приложения ("*nice value*") [величина по умолчанию - 0]

Величина смещения приоритета, устанавливаемая POSIX командой *nice*.

**-M** - периодически возобновлять подписку на мультикаст-канал [по умолчанию - 0 (секунд), т.е. не возобновлять]

Установка данного параметра в значение *N*, отличное от нуля, заставляет приложение возобновлять подписку на мультикаст-канал(ы) клиента каждые *N* секунд.