

IGMP PROXY

New Version 0.2.1: <https://github.com/pali/igmpproxy>

Old version 0.1.0:

```
wget
http://downloads.sourceforge.net/project/igmpproxy/igmpproxy/0.1/igmpproxy-0
.1.tar.gz
tar xzf igmpproxy-0.1.tar.gz
cd igmpproxy-0.1
```

Компиляция и установка:

```
./configure && make && make install
```

По умолчанию в Ubuntu igmpproxy устанавливается в /usr/local/sbin/

Редактируем конфигурационный файл:

```
vi /usr/local/etc/igmpproxy.conf
```

[Пример конфигурационного файла](#)

```
##-----
## Enable Quickleave mode (Sends Leave instantly)
## Эту строчку раскомментируют только в том случае,
## если внутри всего лишь один клиент IP TV
##-----
#quickleave

##-----
## Configuration for eth0 (Upstream Interface)
## Здесь указаны параметры для публичного интерфейса,
## на который придет поток от провайдера
## и указаны сети, с которых возможно будут подключаться клиенты
##-----
phyint eth0 upstream ratelimit 0 threshold 1
altnet 10.0.0.0/8
altnet 172.20.20.0/24
altnet 192.168.0.0/24

##-----
## Configuration for eth1 (Downstream Interface)
## Типовая конфигурация для внутреннего интерфейса
## смотрящего в домашнюю сеть
##-----
```

```
phyint eth1 downstream ratelimit 0 threshold 1
```

Запуск вручную `/usr/local/sbin/igmpproxy /usr/local/etc/igmpproxy.conf`.
Автостарт - добавить `igmpproxy /etc/igmpproxy.conf` & в `/etc/rc.local`.

Более элегантный вариант запускать как демона через `/etc/init.d/` (Рекомендуется!!!)

```
vi /etc/init.d/igmpproxy
```

Приводим к виду:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: igmp proxy
# Required-Start: $remote_fs
# Required-Stop: $remote_fsx
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Example initscript
# Description: IGMP Proxy init file
### END INIT INFO
# Do NOT "set -e"
# PATH should only include /usr/* if it runs after the mountnfs.sh script
#PATH=/sbin:/usr/sbin:/bin:/usr/bin
DESC="IGMP Proxy"
NAME=igmpproxy
DAEMON=/usr/local/sbin/$NAME
DAEMON_ARGS="/etc/igmpproxy.conf"
PIDFILE=/var/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0
# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME
# Load the VERBOSE setting and other rcS variables
. /lib/init/vars.sh
# Define LSB log_* functions.
# Depend on lsb-base (>= 3.0-6) to ensure that this file is present.
. /lib/lsb/init-functions
#
# Function that starts the daemon/service
#
do_start()
{
# Return
# 0 if daemon has been started
# 1 if daemon was already running
# 2 if daemon could not be started
start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --test
```

```

> /dev/null \
|| return 1
start-stop-daemon --start --quiet --background --make-pidfile --pidfile
$PIDFILE --exec $DAEMON -- \
$DAEMON_ARGS \
|| return 2
# Add code here, if necessary, that waits for the process to be ready
# to handle requests from services started subsequently which depend
# on this one. As a last resort, sleep for some time.
}
#
# Function that stops the daemon/service
#
do_stop()
{
# Return
# 0 if daemon has been stopped
# 1 if daemon was already stopped
# 2 if daemon could not be stopped
# other if a failure occurred
start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE
--name $NAME
RETVAL="$?"
[ "$RETVAL" = 2 ] && return 2
# Wait for children to finish too if this is a daemon that forks
# and if the daemon is only ever run from this initscript.
# If the above conditions are not satisfied then add some other code
# that waits for the process to drop all resources that could be
# needed by services started subsequently. A last resort is to
# sleep for some time.
start-stop-daemon --stop --quiet --oknodo --retry=0/30/KILL/5 --exec
$DAEMON
[ "$?" = 2 ] && return 2
# Many daemons don't delete their pidfiles when they exit.
rm -f $PIDFILE
return "$RETVAL"
}
#
# Function that sends a SIGHUP to the daemon/service
#
do_reload() {
#
# If the daemon can reload its configuration without
# restarting (for example, when it is sent a SIGHUP),
# then implement that here.
#
start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE --name $NAME
return 0
}
case "$1" in
start)

```

```
[ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
do_start
case "$?" in
0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
esac
;;
stop)
[ "$VERBOSE" != no ] && log_daemon_msg "Stopping $DESC" "$NAME"
do_stop
case "$?" in
0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
esac
;;
#reload|force-reload)
#
# If do_reload() is not implemented then leave this commented out
# and leave 'force-reload' as an alias for 'restart'.
#
#log_daemon_msg "Reloading $DESC" "$NAME"
#do_reload
#log_end_msg $?
#;;
restart|force-reload)
#
# If the "reload" option is implemented then remove the
# 'force-reload' alias
#
log_daemon_msg "Restarting $DESC" "$NAME"
do_stop
case "$?" in
0|1)
do_start
case "$?" in
0) log_end_msg 0 ;;
1) log_end_msg 1 ;; # Old process is still running
*) log_end_msg 1 ;; # Failed to start
esac
;;
*)
# Failed to stop
log_end_msg 1
;;
esac
;;
*)
#echo "Usage: $SCRIPTNAME {start|stop|restart|reload|force-reload}" >&2
echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload}" >&2
exit 3
```

```
;;  
esac  
:
```

Устанавливаем права и запуск

```
chown root:root /etc/init.d/igmpproxy # Меняем владельца  
chmod 755 /etc/init.d/igmpproxy # Меняем права и делаем исполняемым  
update-rc.d igmpproxy defaults # Добавляем в загрузку  
  
/etc/init.d/igmpproxy start # Старт/Рестарт/Стоп
```

[Дополнительное инфо](#)

From:
<https://docs.infomir.com.ua/> -

Permanent link:
https://docs.infomir.com.ua/doku.php?id=knowledge_base:igmp_proxy_config

Last update: **2021/12/15 14:36**

