

Streaming Audio from one Raspberry Pi to another.

How to stream audio from one Pi to another Pi (or an Ubuntu Linux PC). We use the Netcat command to pipe the audio over a network connection from one Pi to the other.

This method uses no more than 30% of processor cycles on an old model B or A+ Pi and works flawlessly. Our preferred method of setting this up is to simultaneously SSH into each Pi remotely from our desktop PC.

On both the sending and receiving Raspberry Pi you need to install the Opus codec:

```
sudo apt-get install opus-tools
```

Discover and make a note of the IP address (here 192.168.1.39) of each Pi with the command :

```
ifconfig
```

```
pi@raspberrypi:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:81:0a:33
          inet addr:192.168.1.39  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::7b52:55f5:7d60:3e81/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19164 errors:0 dropped:6 overruns:0 frame:0
          TX packets:3804 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1213749 (1.1 MiB)  TX bytes:2240409 (2.1 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi:~$
```

Tip: setup each Pi in your router so they always get given the same static IP address. Discover and note useful information about the USB sound cards with commands :

```
lsusb
cat /proc/asound/pcm
```

```
pi@raspberrypi:~$ cat /proc/asound/pcm
00-00: bcm2835 ALSA : bcm2835 ALSA : playback 8
00-01: bcm2835 ALSA : bcm2835 IEC958/HDMI : playback 1
01-00: USB Audio : USB Audio : playback 1 : capture 1
pi@raspberrypi:~$
```

In the example above our USB audio device is on port plughw:1,0

On the receiving Pi

Open up port 8086 in your firewall with:

```
sudo /sbin/iptables -I INPUT 1 -p tcp --dport 8086 -j ACCEPT
```

and check it has been accepted with

```
sudo iptables-save
```

```
pi@raspberrypi ~ $ sudo iptables-save
# Generated by iptables-save v1.4.21 on Thu Apr 28 19:39:04 2016
*filter
:INPUT ACCEPT [3977:258407]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [11006:634460]
-A INPUT -p tcp -m tcp --dport 8086 -j ACCEPT
COMMIT
# Completed on Thu Apr 28 19:39:04 2016
pi@raspberrypi ~ $
```

then set the Pi listening with

```
nc -l -p 8086 | opusdec - - | aplay -D plughw:0,0 -f dat
```

if you don't set the receiving Pi going first, the sender will fail.

```
pi@raspberrypi ~ $ nc -l -p 8086 | opusdec - - | aplay -D plughw:0,0 -f dat
Decoding to 48000 Hz (2 channels)
Encoded with libopus 1.1
ENCODER=opusenc from opus-tools 0.1.9
ENCODER_OPTIONS=--bitrate 64 --max-delay 0 --comp 0 --framesize 2.5 --hard-cbr
Playing raw data 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
[\] 00:01:00
```

You can force the receiving Pi to use the 3.5mm headphone output, instead of HDMI audio out, with:

```
amixer cset numid=3 1
```

On the sending Pi:

Plug in your USB Audio Mic device and use commands:

```
lsusb
```

and

```
cat /proc/asound/pcm
```

```
pi@raspberrypi:~$ cat /proc/asound/pcm
00-00: bcm2835 ALSA : bcm2835 ALSA : playback 8
00-01: bcm2835 ALSA : bcm2835 IEC958/HDMI : playback 1
01-00: USB Audio : USB Audio : playback 1 : capture 1
pi@raspberrypi:~$
```

The correct audio device here is plughw:1,0 - see below. The correct IP address in the example below should be the Pi you are transmitting to:

```
arecord -D plughw:1,0 -f dat | opusenc --bitrate 64 --max-delay 0
--comp 0 --framesize 2.5 --hard-cbr - - | nc 192.168.1.18 8086
```

(The command above all goes on one line)

```
pi@raspberrypi:~$ arecord -D plughw:1,0 -f dat | opusenc --bitrate 64 --max-delay 0 --comp
0 --framesize 2.5 --hard-cbr - - | nc 192.168.1.30 8086
Recording WAVE 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
Encoding using libopus 1.1 (low-delay)
-----
Input: 48kHz 2 channels
Output: 2 channels (2 coupled)
        2.5ms packets, 64kbit/sec CBR
Preskip: 120

[/] 00:05:37.89 0.997x realtime, 64kbit/s
```

If the command above exits immediately, then the port 8086 probably isn't open on the receiving Pi (or you haven't set the receive command running on the other Pi). Check the correct port is open on the receiving Pi with:

```
sudo iptables-save
```


Different audio sampling rates

At the moment we're capturing audio in stereo at 48000Hz, which is higher quality than a CD. If capturing CCTV audio and streaming it over the internet, then we can manage with a lower sample rate and single channel mono.

Either of these commands will make a local recording at a lower sample rate:

```
arecord -D plughw:1,0 -f S16_LE -c1 -r8000 -t wav test8000.wav
```

```
arecord -D plughw:1,0 -f S16_LE -c1 -r22050 -t wav test22050.wav
```

You could for example use a PIR sensor, connected to the Pi's GPIO pins, to only record audio for a set duration (-d) when motion is detected and have the filename be the date and time.

You can compare the file sizes with

```
ls -al
```

The 8000Hz sample rate generates the smallest files and is okay for speech. The 22050Hz sample rate is equivalent to FM radio. 44100Hz is CD quality and 48000Hz is DAT or DVD audio.

Try this one on the receiving Pi:

```
nc -l -p 8086 | opusdec - - | aplay -D plughw:0,0 -f S16_LE -c1 -r8000 -t wav
```

and this on the transmitting Pi:

```
arecord -D plughw:1,0 -f S16_LE -c1 -r8000 -t wav | opusenc --bitrate 64 --max-delay 0 --comp 0 --framesize 2.5 --hard-cbr - - | nc 192.168.1.8 8086
```

(commands all go on one line, type it in. Don't copy & paste from the PDF)

If you want to capture the audio into a file at the receiving end, you can do this instead:

```
nc -l -p 8086 | opusdec - - > test.wav
```

and play the recording back later with

```
aplay -f S16_LE -c1 -r8000 -t wav test.wav
```

Dropping the sample rate from 48000Hz down to 8000Hz, and mono rather than stereo, means that on an original Raspberry Pi model B or A+ the %CPU cycles used with Opus (measured with command `top`) drop from around 27% to 17%. But what advantage does using Opus give us?

We used the command `iftop` to measure network traffic on each Pi while using 48000Hz and Opus compression (151Kb a second) and without compression (10x more traffic = 1.52Mb a second). At 8000Hz Mono though it turns out it's not worth using Opus (151Kb/s v 132Kb/s none).

Hardware Choices.

The Raspberry Pi doesn't have a microphone input, so we need to use some kind of USB interface and a microphone.

There are essentially two types of microphones we're interested in: Mics that are meant to capture just the audio a couple feet around them (commonly used for Skype etc), and powered microphones with a pre-amp which capture all the audio in the environment (used for CCTV and surveillance).

If you want to add audio to your bird box project, then you could use a small USB microphone like this one (which typically cost just a few pounds from Chinese eBay sellers).



Portable Mini USB Desktop Studio Speech Recording Microphone F
Skype Msn Video

£0.99
Buy It Now
or Best Offer
Free Postage

From China

Item: 381621072230

 eBay Premium Service

But this Mic is no good for picking up audio from more than 2 feet away – of course you can also use a USB extension lead of up to 6 metres to move this Mic nearer to your source.

It's also possible to buy USB adapters with a standard 3.5mm mic input, and you can then attach a better quality Mic to it, but again these are only for audio which is near the mic.

A CCTV microphone.



The CCTV mic is powered by a 12 volt DC supply. It contains a microphone and an amplifier. The 12V supply attaches to the red 2.1mm plug. The white phono socket attaches to a USB soundcard with a Line Level input.

CCTV mics are sometimes referred to as “room mics”, because they pick-up every noise in a room very clearly.

Power supply for the CCTV Mic



Universal DC 12V Volt 5A Amp 4 Way Splitter CCTV Power Supply Adapter UK Plug (111829710042)



ITEM PRICE:
£11.99

i Estimated delivery Fri, 27 May

+ Add note

We recommend that you use a 12 volt regulated & earthed power supply like this one from eBay for £11.99. It can also power Infra Red security lights. We tried several cheaper unregulated unearthed 12 volt supplies, but we always got an annoying buzzing background noise that disappeared once we used a decent power supply. It's also possible to power the Mic from a 9 or 12 volt battery pack.

USB Audio capture device.

Because the CCTV mic is at line-level, you can't attach it to the Mic-In port on a laptop, PC or USB adapter, you have to use Line-In or a USB adapter with Line-In, like this one:



The Ezcap USB audio adapter works really well with the Raspberry Pi. If you wanted to, you could attach two of the CCTV microphones for recording proper stereo sound. Don't bother trying to use the cheap £3 USB mic & headphone adapters, as they're not line-level & don't work correctly.

The USB lead plugs into your Raspberry Pi and you use a male-2-male phono adapter to join the two white phono sockets together.

As we've seen in the examples, it's possible to use this equipment & two Raspberry Pi boards to stream audio from one location to another over the internet using netcat.

There are some useful tips for voice activated recording here: <http://mocha.freeshell.org/audio.html>

You can download our CCTV Mic sample recording here: <http://www.tpr2.co.uk/test8000.wav>