

How are we doing? Please help us improve Stack Overflow. [Take our short survey](#)

## C# - Live Streaming Audio over Socket Connection

Asked 1 year, 8 months ago   Active 1 year, 8 months ago   Viewed 1k times



2



I am trying to stream audio from one computer to another over a tcp socket connection, one-way.

Computer 1 ---> Switch ---> Computer 2

Computer 1 is the sender/client, Computer 2 is the receiver/server that plays the audio.

I managed to get it to somewhat work, but there's a lot of noise being received.

### CSCore:

This is the code I'm using at the moment to send the data over:

```
CaptureInstance = new WasapiLoopbackCapture();
CaptureInstance.Initialize();
CaptureInstance.DataAvailable += (s, a) => onStart(s, a);
CaptureInstance.Start();
```

where onStart simply writes the bytes to the socket's stream using a BinaryWriter:

```
writer.Write(a.Data, a.Offset, a.ByteCount);
```

This is the code I'm using at the moment on the receiving computer to play the audio:

```
int size = 3840;
WriteableBufferingSource src = new WriteableBufferingSource(new
CSCore.WaveFormat(48000, 32, 2)) { FillWithZeros = true };

WasapiOut soundOut = new WasapiOut();
soundOut.Initialize(src);
soundOut.Play();

while (true)
{
    byte[] bytes = reader.ReadBytes(size); // BinaryReader
    src.Write(bytes, 0, size);
}
```

### NAudio:

Upon further search, trial and error, I managed to piece together a new solution using NAudio, which also suffers from the same issue. A lot of noise, and very loud.

**Join Stack Overflow** to learn, share knowledge, and build your career.

[Sign up](#)



```

var capture = new WasapiLoopbackCapture();

capture.DataAvailable += (obj, args) =>
{
    stream.Write(args.Buffer, 0, args.BytesRecorded); // NetworkStream(socket)
    stream.Flush();
};

capture.StartRecording();

```

The server (receiver) is using this:

```

// ----- Init -----
provider = new BufferedWaveProvider(new WaveFormat(48000, 32, 2));

WaveOut o = new WaveOut();
o.Init(provider);
o.Play();

// ----- Callback -----
socket = (Socket)ar.AsyncState;

int received = socket.EndReceive(ar);
byte[] dataBuffer = new byte[received];
Array.Copy(buffer, dataBuffer, received);

provider.AddSamples(dataBuffer, 0, received);

socket.BeginReceive(buffer, 0, buffer.Length, SocketFlags.None, new
AsyncCallback(ReceiveCallback), socket);

```

I tried allowing the provider to buffer 5-10 sec before playing, just in case it's a buffering issue, but no luck.

I checked the WaveFormat, it's 48k, 32 bit 2 channel on both PCs. I played with the values anyway, failed.

I tried using UdpClient, but that went worse. I can show the code for that if necessary. The playback was slower, and if not as much, more noisy. I used this [link](#) for setting that up.

I ran a loop on both client/server showing the received/sent bytes for each block, printing out the first and last byte and they seemed to be fine and in-order, but while debugging a setup I had earlier with MemoryStream, there was a 12 byte difference in the length of the memory streams (one on the client, writing same bytes that were being sent to the server, and the other on the server, writing the received bytes), and some bytes were off.

Recording the audio on the client for 15 sec and playing it back right there works. When sent to the other computer, it gets noisy as hell.

I am thinking the problem is with my socket setup, but I am not too sure and I am running out of options

tl;dr; question is: What could be causing this heavy noise? Both NAudio/CSCore setups are suffering from the same issue it seems like. Same outcome.

Or how can I proceed to troubleshoot the problem?

c# sockets audio naudio cscore

Share Improve this question Follow

asked Jun 4, 2020 at 21:03



## 1 Answer

Active	Oldest	Score
--------	--------	-------



I figured it out.

3

`new WaveFormat(48000, 32, 2)`



that creates a PCM encoded format, while the Capture Instance's format is leeefloat encoded.

In case anyone needs this, replacing

`new WaveFormat(48000, 32, 2)`

with

`WaveFormat.CreateIeeeFloatWaveFormat(48000, 2)`

fixed the issue.

More to read on WaveFormat [here](#)

Share Improve this answer Follow

edited Jun 5, 2020 at 22:09

answered Jun 5, 2020 at 21:58

