Realtime Audio Compression in QVD

Vadim Troshchinskiy Shmelev

vtroshchinskiy@qindel.com

qindel.com | theqvd.com

Twitter | LinkedIn | Facebook | Google + | YouTube

About Us

- Qindel Group is an international consulting company
- Specialized in Linux
- We have our own product: QVD

How QVD Works

- We provide virtual desktops
- Desktops are hosted in KVM VMs or LXC containers
- We use NX to provide the desktop environment
- Our aim is large deployments, with hundreds of desktops per server
- Data may be stored, or discarded on shutdown

Audio in NX

- nxagent listens on media port
- The connection is forwarded as-is to the remote side.
- VM has $PULSE_SERVER pointing to the port

Advantages

- It’s simple

Disadvantages

- Bandwidth usage is enormous
- One connection per application
- Applications lose the audio connection when the user disconnects
- Applications have access to the host’s PulseAudio daemon
- Audio traffic competes with other NX data

The Solution

- A local PulseAudio instance in the VM/container
  - This solves the audio server disappearing
  - And provides mixing
- Opus compression for PulseAudio
  - Dramatically decreases bandwidth usage
  - Improves desktop performance
  - Can be tuned for the desired quality/bandwidth

Why Do It This Way?

- We need to use and package PulseAudio anyway
- Minimizes the number of components
- Fits quite well into PulseAudio
- PulseAudio provides a control protocol that can be used for adjustments and monitoring
- We hope to get this accepted upstream

The Patch

- Originally developed by gavin_darkglider: https://bugs.freedesktop.org/show_bug.cgi?id=56993
- Ported to the latest version
- Improved
  - Slightly refactored
  - Made Opus parameters configurable
  - Made Opus parameters possible to change at runtime

Using the Patch

Sender

```
load-module module-tunnel-sink-new sink_name=QVD_Audio
server=tcp:127.0.0.1:$pa_port sink="@DEFAULT_SINK@”
compression=opus
```

Receiver

```
load-module module-native-protocol-tcp auth-ip-
acl=$client_ip
```

Modifying Parameters

```
update-sink-proplist QVD_Audio
compression.opus.max_bandwidth=64000
```

Implementation in QVD

- Windows
  - We build and package our own PulseAudio
- OSX
  - We build and package our own PulseAudio
- Linux
  - We now must build and package our own PulseAudio
  - But...

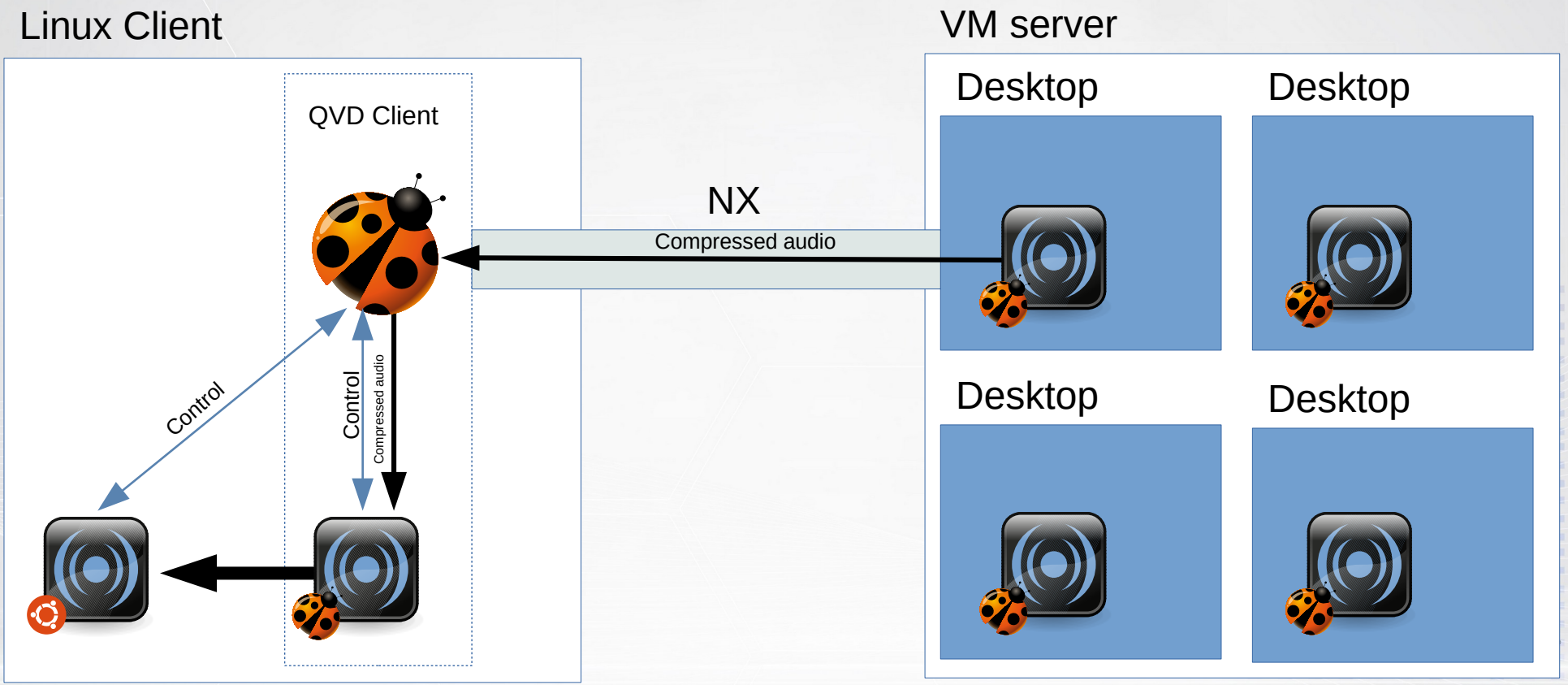
Linux Implementation

- Three possibilities
  - The system supports Opus natively
  - The user has our PulseAudio package
  - The user doesn’t have our package

Running a Custom PulseAudio on Linux

- Replacing the system PulseAudio is a dangerous idea
  - Needs to be done for every distribution
  - Easy to get wrong and break the user's system
  - It will be there even when not running QVD
  - It will complicate upgrades
- So we don't
  - One instance in the VM
  - Two instances on the client side:
    - Our patched version
    - The system's own version

Diagram


Linux Client
QVD Client
Control
Control
Compressed audio
NX
Compressed audio
VM server
Desktop
Desktop
Desktop
Desktop

Linux Client Implementation

- Talks to the system PA using the UNIX socket
- Tests whether it supports compression
- Launches a patched version of needed
- Sets up the network connection

Windows and OSX Implementation

- The client launches our custom PulseAudio build
- It has a custom-made configuration file
- There’s no system PA, so it’s simple

Challenges

- Currently PulseAudio doesn’t provide a very good way of checking whether compression is supported.
- We need to do a custom build without DBus support to avoid system interference

Future Improvements

- Better compressed audio support detection
- Take advantage of the possibilities of runtime control

Questions?