netpd

Welcome to LAC 2013

netpd

What is netpd?

Collaborative
Realtime
Networked
Musik
Making
Environment

written in Pure Data

netpd

What is netpd really?

* a framework written in Pure Data
* shares patches/abstractions among clients
* synchronizes state among clients
* all participants have control over everything
* by transmitting control data
* does not stream audio
* doesn't know anything about sound or music

netpd

Users can:

* chat with each other
* participate in an online jam
* enjoy their ego session
* play with pre-made instruments
* write their own instruments
* share their instruments
* ...

n e t p d

chat.pd

* manages connection to the server

* sends and receives messages over network

* lets the user chat with other users

* launches unpatch

* requests client ID from server

{show chat.pd}

netpd

unpatch

* loads/closes instruments

* keeps list of instruments in sync
  * checks version of instrument
  * resolves dependencies
  * transfers instruments/abstractions
    between clients (p2p for patches)

{show unpatch}

netpd

netpd-abstractions

* help netpd-izing instruments

* synchronize state of instruments

* different types for different data sets

* state initialization

n e t p d

`netpd_head $1 megasynth`   - state management

`netpd_a $1 anything`   - anything message

`netpd_f $1 volume 0.7`   - float (built-in s/r for iemguis)

`netpd_t $1 waveform 256`   - table

`netpd_s $1 something`   - simple send (broadcast)

`netpd_r $1 something`   - simple receive

{show code of bassist}

netpd

netpd metatags

* part of any instr / abs

* specify instrument properties

* parsed by unpatch at loading time

* required for instrument synchronicity

n e t p d

netpd metatags

`pd NETPD 2 0` {   - declares metatags

`version 0 3 1`   - version of instr / abs

`singleton`   - load only one instance of instr

`pd abslist` {   - list of dependencies

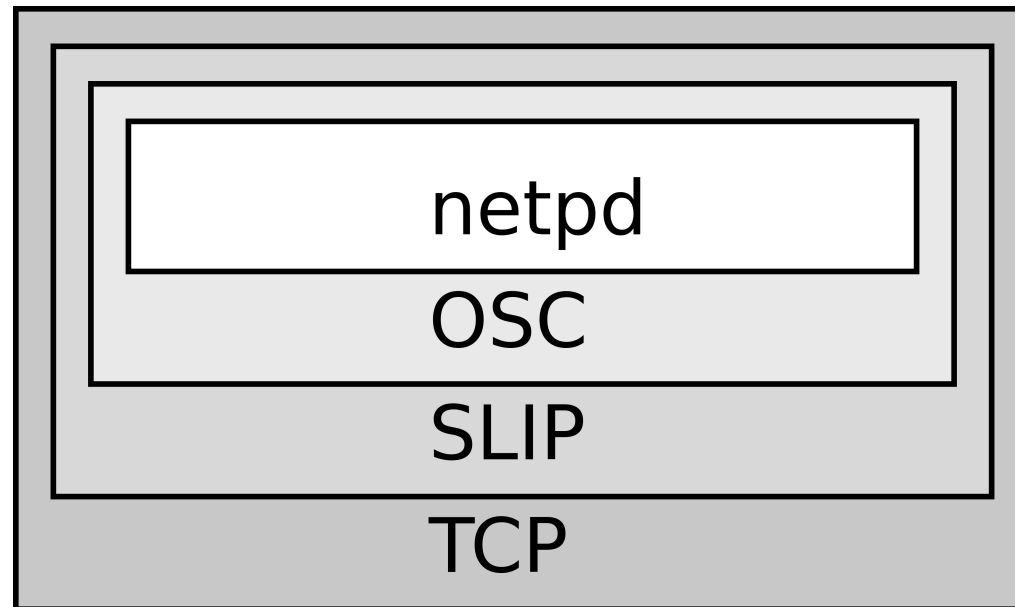`synthvoice`   - dependency (abstraction)

}

}

{show code of bassist}

n e t p d

netpd server

* implements netpd protocol

* acts as OSC relay

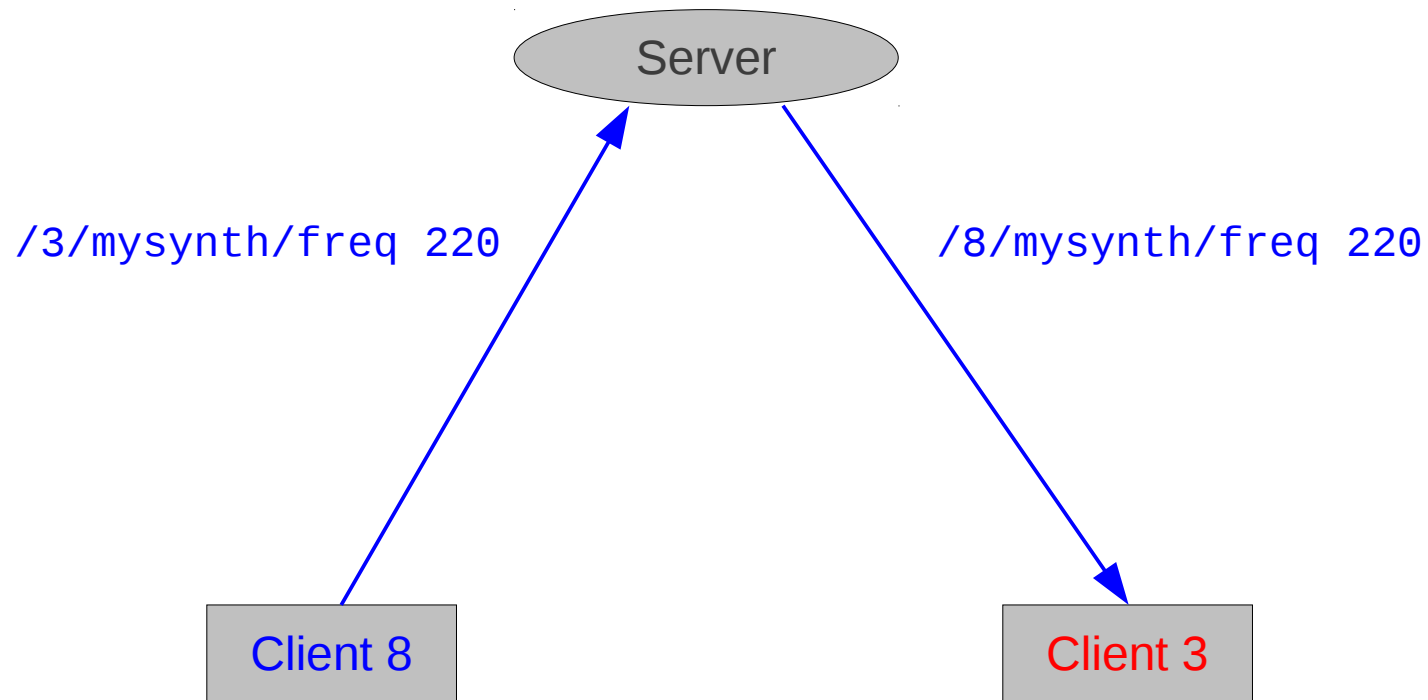* is stateless

* assigns client IDs
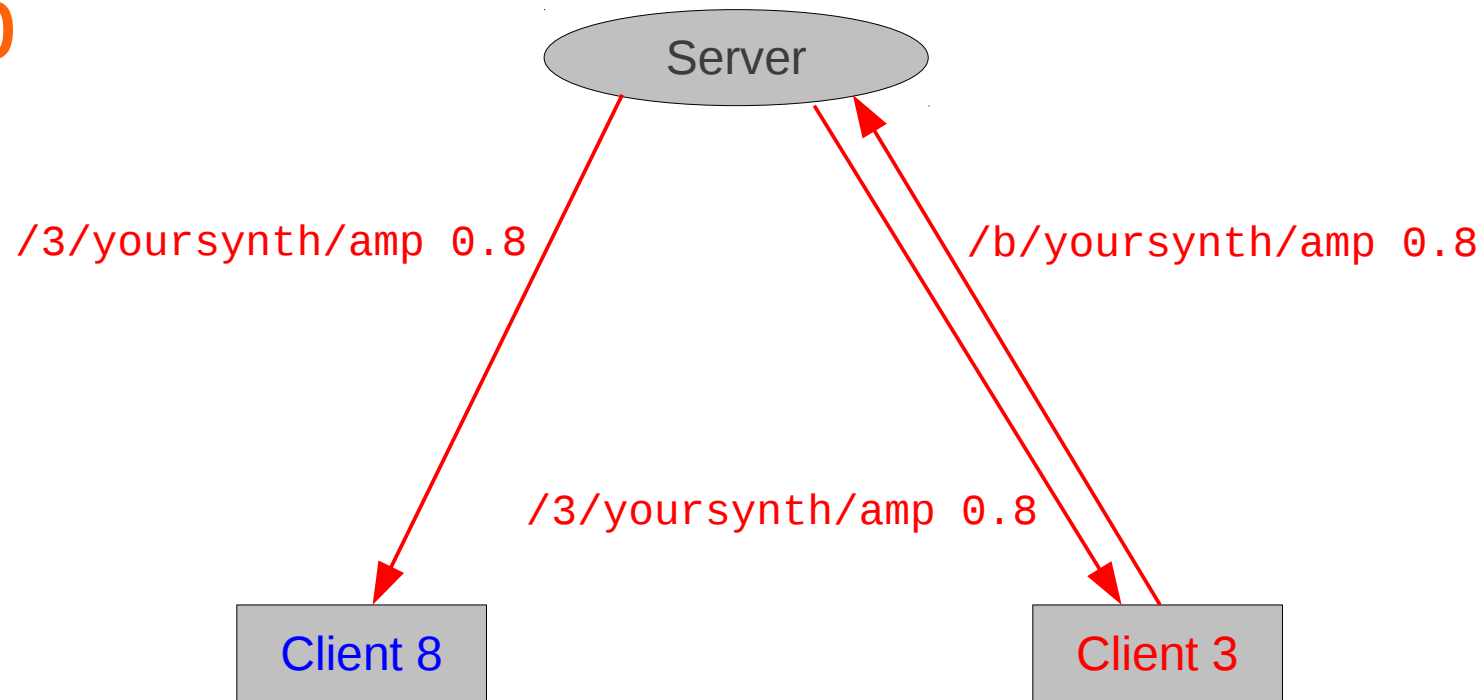
* runs on netpd.org:3025

netpd

protocol stack:

netpd
OSC
SLIP
TCP

netpd

OSC address scheme

```
 receiver ID
 |
 | instrument ID
 | |
 | |       instrument name
 | |       |
 | |       |       parameter name
 | |       |       |
/b/7/megasynth/freq 50
```

reserved namespaces:
=====================

/*/chat/
/*/unpatch/
/s/server/

{show DEBUG_PROTO}

netpd

/3

Server

/3/mysynth/freq 220   /8/mysynth/freq 220

Client 8   Client 3

netpd

/b

Server

/3/yoursynth/amp 0.8

/b/yoursynth/amp 0.8

/3/yoursynth/amp 0.8

Client 8

Client 3

netpd

/l

Server

* used for state initialisation

/l/mysynth/att 10
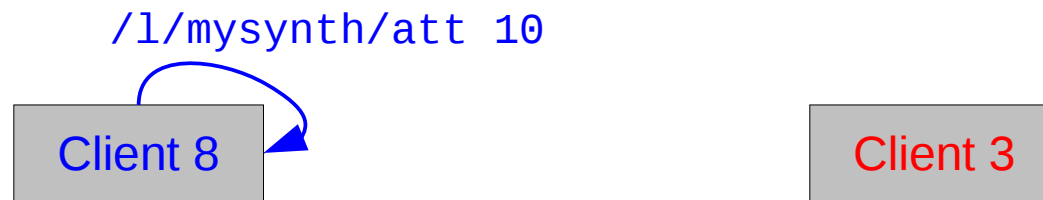
Client 8

Client 3

netpd

/s

server methods:
===============

/s/server/socket
/s/server/ip
/s/server/num_of_clients
/s/server/protocol_version

Server

/s/server/socket

/s/server/socket 3

Client 8

Client 3
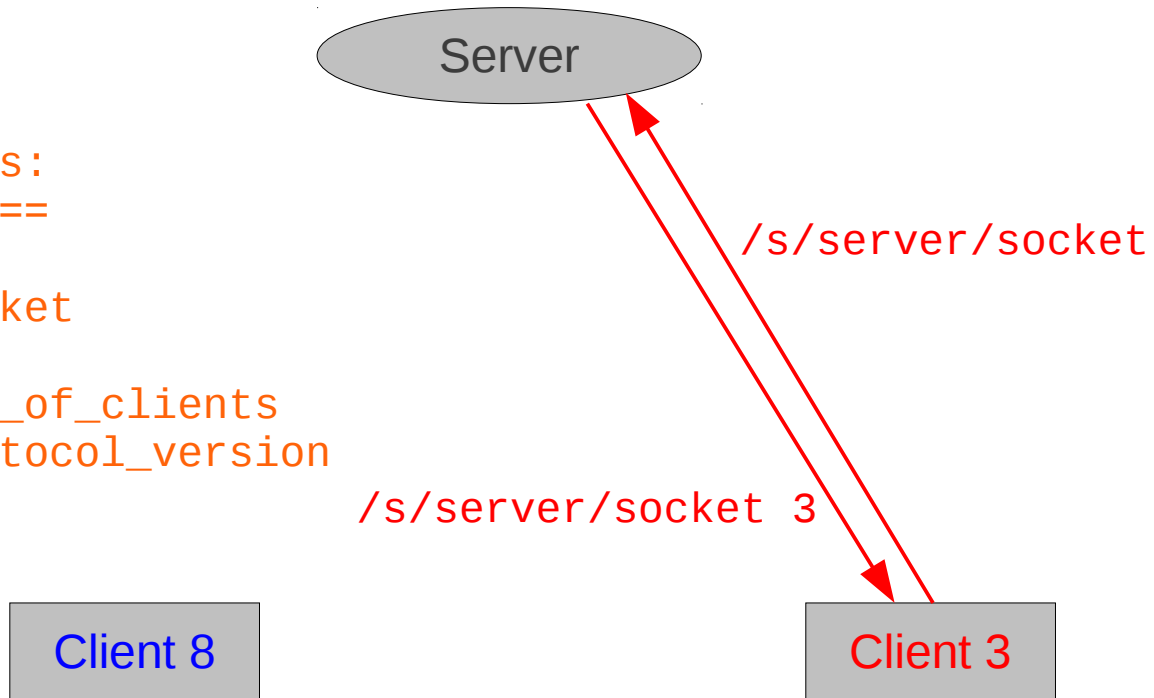
netpd

showcase

{show neptd in action}

netpd

issues

* security (remote code execution by design)

* audio drop outs

* no sync verification

* limited GUI possibilites (in Pd)

netpd

alternative usage scenarios:

* OpenGL visuals jam (with Gem)

* networked Pd based games

* remote teaching

* ...

n e t p d

resources

* web: http://www.netpd.org

* code: https://github.com/reduzent/netpd2.git

* sound: http://www.netpd.org/sessions/

netpd

questions?