

LSCube

- Core
 - NetEmbryo : Funzionalità di Rete
 - Bufferpool : Gestione coda pacchetti RTP
- Services
 - Feng : Server RTSP
 - Felix : Mixer e "regia" RTP
 - libNemesi : Client RTSP
 - Theater : Accesso Web alla mediateca di Feng

NetEmbryo

Layer astrazione di rete

- Astrazioni di rete
- Supporto SCTP
- **Supporto SSL**
- Report errori RTSP
- Parsing URI
- Compatibilità WIN32

Bufferpool Memory Manager

- Memorizzazione specifica dei pacchetti RTP
- Uso di un buffer privato o Shared Memory
- Sincronizzazione su shared semaphores
- Supporta un producer e N consumer
- Tiene traccia dello stato dei consumer per l'accesso sequenziale

Feng

- **Video**

- MPEG Video (MPEG-1/2) (rfc2250)
- MPEG 4 Visual (MPEG-4 Part 2) (rfc3016)
- H.264 (MPEG-4 Part 10) (rfc3984)
- H.263 / H.263+ (rfc4629)
- Theora

- **Audio**

- MPEG (MPEG-1/2 Layer I/II/III) (rfc2250)
- Vorbis
- AAC (MPEG-4 Part 3) (rfc3640)
- SPEEX

- **Metademuxers**

- SD
- DS

Supporto di rete

- Mezzi trasmissivi
 - **TCP**
 - **UDP**
 - **SCTP** (ove almeno le due estremità lo supportino)
- Traffic Smoothing
- Multicast tramite SD con client pilota

Preferenze di Feng

- **root** = directory con i filmati
- **tcp_port** = porta ascolto richieste
- **sctp_port** = porta ascolto richieste SCTP
- **max_session** = numero massimo di client (Limite hardcoded di 100)
- **log_file** = file di logging
- **first_udp_port** = prima porta UDP da cui iniziare le trasmissioni
- **buffered_frames** = frame da prebufferizzare
- **user** = utente con cui gira il servizio
- **group** = gruppo con cui gira il servizio

Lemon Feng

```
## new lighttpd like configuration
var.basedir = "/var/feng"
var.logdir  = "/var/log"
var.statedir = "/var/state"

server.modules = (
    "mod_access",
    "mod_accesslog"
)
server.port      = 8554
server.username  = "feng"
server.groupname = "feng"
server.max-connections = 100
server.document-root = var.basedir + "/avroot"
server.errorlog    = var.logdir + "/error.log"
accesslog.filename = var.logdir + "/access.log"
```

Metademuxer DS

- Permette la concatenazione di parti di più stream in uno.
- Gli stream devono essere tutti dello stesso formato (codec, risoluzione, bitrate)
- Il risultato è seekabile se gli input sono seekabili

Definito da un file .ds posto nella avroot di feng.

E` una lista di tuple del formato: *nome_risorsa TempoInizio TempoFine*

ESEMPIO

resource.mov 0.0 1.0

resource.ds 1.0 3.0

resource.nut 3.0 4.0

Metademuxer SD

- Permette di definire stream che sono la fusione di altri stream
- Ereditato dal vecchio fenice
- Non garantisce la sincronia tra gli stream
- E` attualmente instabile per i payload non definiti nello standard RTP (payload type \geq 96)

Configurazione SD

- Definito da un file .sd posto nella avroot di feng.

- E` una lista di blocchi del formato:

stream

[opzioni]

stream_end

ESEMPIO:

```
stream
  file_name audio.mp3
  payload_type 14
  clock_rate 90000
  encoding_name MPA
stream_end
stream
  file_name video.264
  payload_type 96
  clock_rate 90000
  encoding_name H264
  frame_rate 25
stream_end
```

Opzioni stream SD

- **file_name** nome_del_file
- **encoding_name** nome_del_codec (Supportati: MPV, MPA, H264, aac, H263P, mp4v-es, MP2T, theora, vorbis)
- **payload_type** id (I codici < 96 sono preassegnati secondo rfc)
- **clock_rate** rate_in_hz
- **audio_channels** numero_canali (obbligatorio solo per i payload > 96)
- **media_type** "AUDIO"/"VIDEO" (obbligatorio solo per payload > 96)
- **frame_rate** rate_float (solo per VIDEO)
- **bit_rate** numero_bit_sec (opzionale)
- **sample_rate** rate_float (usato solo in MP3, opzionale)
- **media_source** "STORED"/"LIVE" (definisce se la sorgente è live o salvata (le live non sono pausabili), opzionale)
- **fntp** extradati_del_codec (opzionale, solo per i payload con extradati)
- **multicast** ip (opzionale, specifica che lo stream risultante è multicast)
- **port** porta_int (opzionale)

Felix

- Dato un input in ingresso permette di:
 - Inserirlo nel bufferpool di feng su Shared Memory (Funziona solo per payload statici)
 - Rispedirlo ad un altro host

```
felix -c in=1233,out=rtp-shm://video.rtp,out=udp://host:port -c  
in=1234,out=rtp-shm://audio.rtp
```

libNemesi

- Video

- H263
- H264
- MPEG 4 - Video
- MPEG 1/2
- Theora

- Audio

- MPEG 1/2
- AAC
- Vorbis
- SPEEX

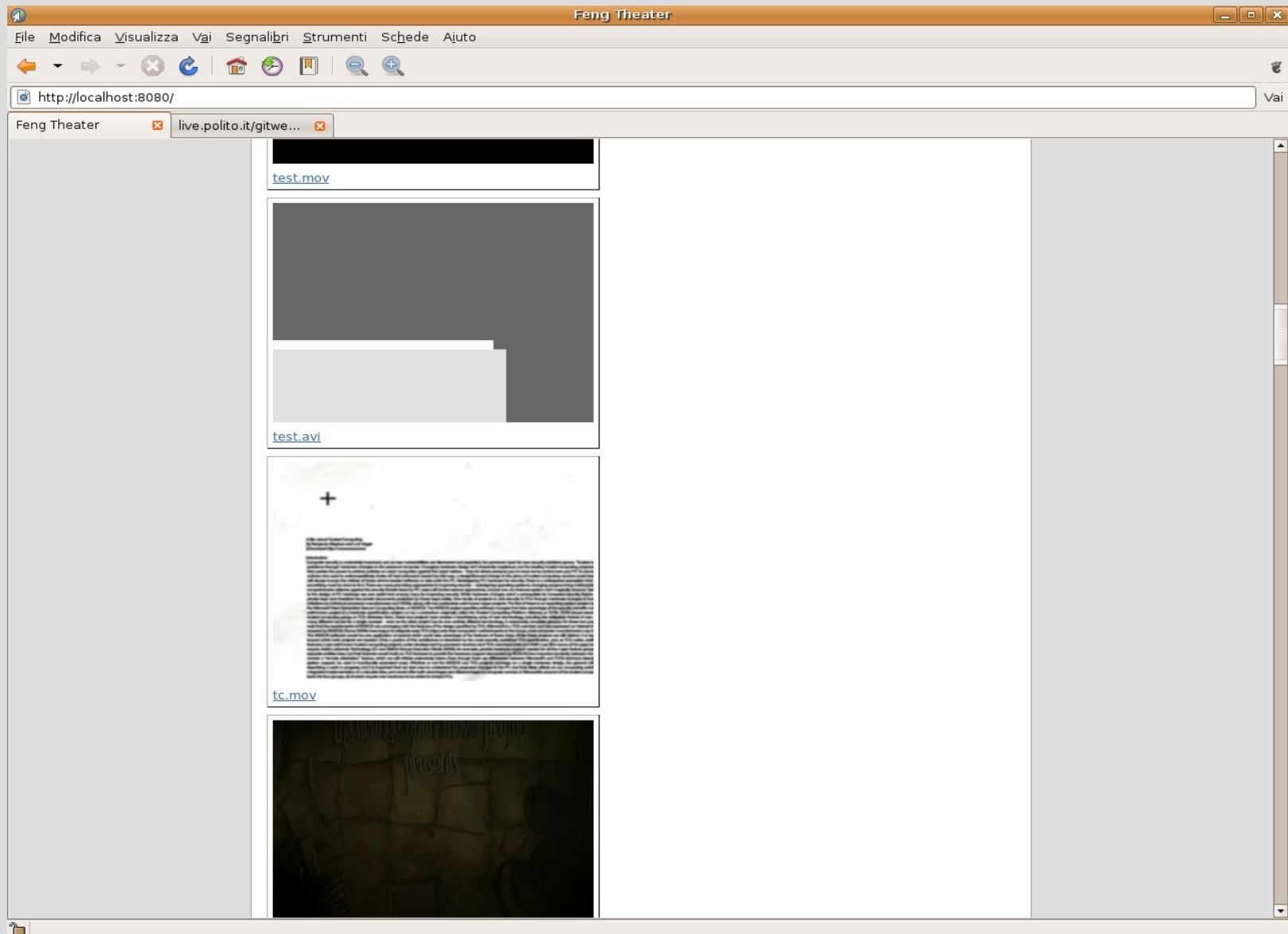
libNemesi Status

- Compatibile con formato buffer atteso da ffmpeg
- Attualmente in uso dentro MPlayer
- **Funzionamento di base in ambiente WIN32**
- Documentazione completa su http://live.polito.it/files/nemesi_doc/index.html

Theater

- Listing automatico dei contenuti serviti da Feng
- Generazione anteprime per i filmati
- Generazione automatica pagina di visione dei filmati via web

Theater Listing



Theater Content View

Feng Theater - /tc.mov

File Modifica Visualizza Vai Segnalibri Strumenti Schede Aiuto

http://localhost:8080/view?what=/tc.mov

Feng Theater - /tc... live.polito.it/gitwe...

FENG AVRoot

[Back to Directory](#)




FIGURE: SOME COMMENTS ON MC

Computer security is undeniably important, and as new vulnerabilities are discovered and exploited, the perceived need for new security solutions grows. "Trusted computing" initiatives propose to solve some of today's security problems through hardware changes to the personal computer. Changing hardware design isn't inherently expensive, but the leading

Playing (00:09/03:30)

TURBOGEARS
under the hood

TurboGears is a open source front-to-back web development framework written in Python

Copyright © 2007 Kevin Dangoor

LSCube in the future

- 1) Configurazione unificata Feng (15 giorni)
- 2) Port del layer di rete di libNemesi e Felix su NetEmbryo (10 giorni)
- 3) Implementazione transport TCP+SSL su Felix (15 giorni)
- 4) Implementazione Funzionalità di regia su Felix (1 mese)
- 5) Supporto per plugins Feng (1 mese)
- 6) Supporto autenticazione Feng, libNemesi
- 7) Secure RTP Feng, libNemesi
- 8) Funzionamento completo in ambiente WIN32 di libNemesi
- 9) Migliore implementazione layer RTCP e gestione pacchetti misordered/duplicati in libNemesi