



Visualizzatore audio MjSIP

Progettazione e realizzazione di un visualizzatore grafico per l'audio inviato e ricevuto tramite MjSIP.

Autori

Mirko Boggian: mirko.boggian@gmail.com

Michele Longari: longari@ce.unipr.it

Lorenzo Mignani: lorenzo.mignani@gmail.com

BoTuLo Team

Descrizione Generale

Il fine del progetto è la realizzazione di un *plugin* che permetta agli utenti MjSIP di visualizzare in *Real-Time* il flusso audio in invio ed in ricezione durante la chiamata SIP.

La *release* di partenza è:

- MjSIP: http://www.mjsip.org/download/mjsip_1.7_beta071105.zip.

Dopo una prima analisi dell'architettura si è pensato di prelevare gli *stream* audio direttamente dalle classi RTPStreamSender e RTPStreamReceiver, in particolar modo andando ad agire nei metodi *read()* e *write()* rispettivamente.

L'idea è quella di creare un Java *Listener* che ad ogni chiamata dei metodi precedentemente descritti permetta il *refresh* dei grafici.

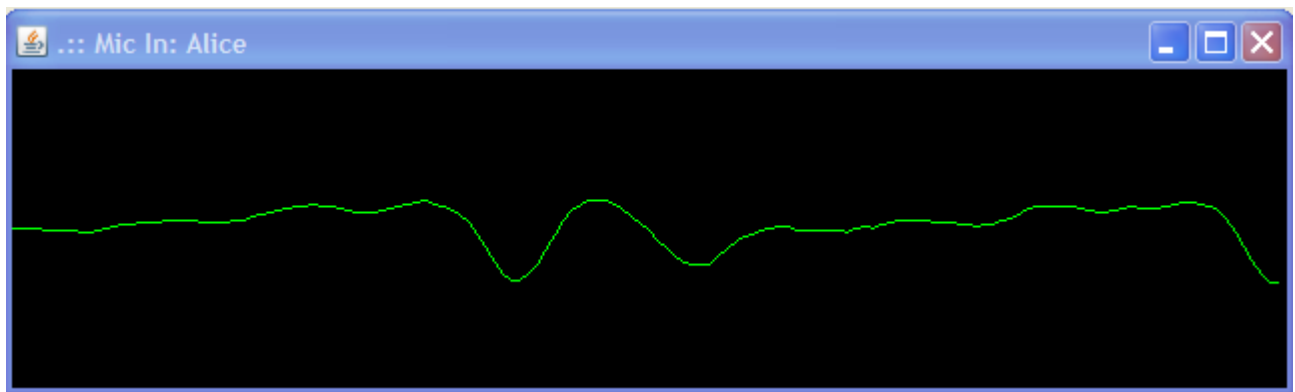
Tali grafici vengono creati in nuovi *Thread* tramite l'estensione del *JPanel* di Java2D e sono accessibili tramite i pulsanti Mic e Speaker presenti nella GUI originaria di MjSIP (GraphicalUA).



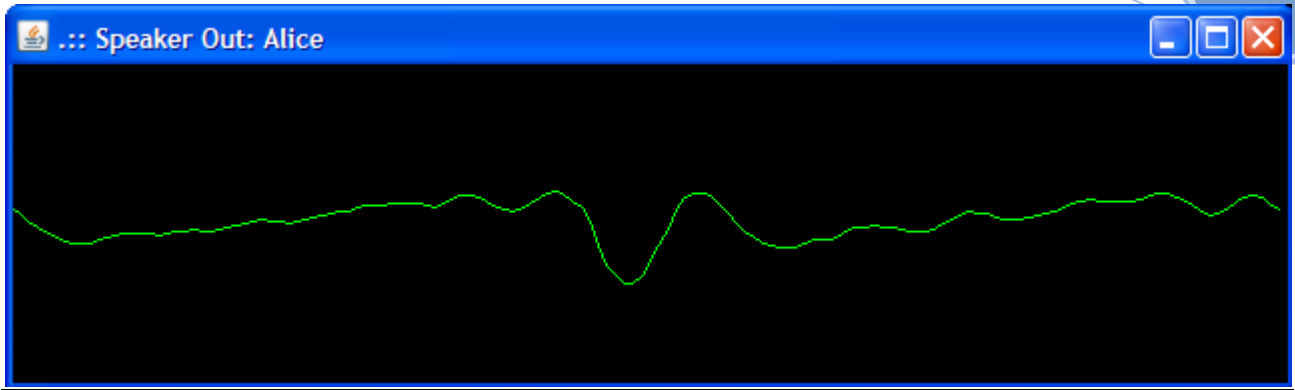
GUI di MjSIP

Ogni refresh visualizza 160 campioni di flusso audio Linearizzati.

Ecco qua di seguito alcuni esempi di grafici:

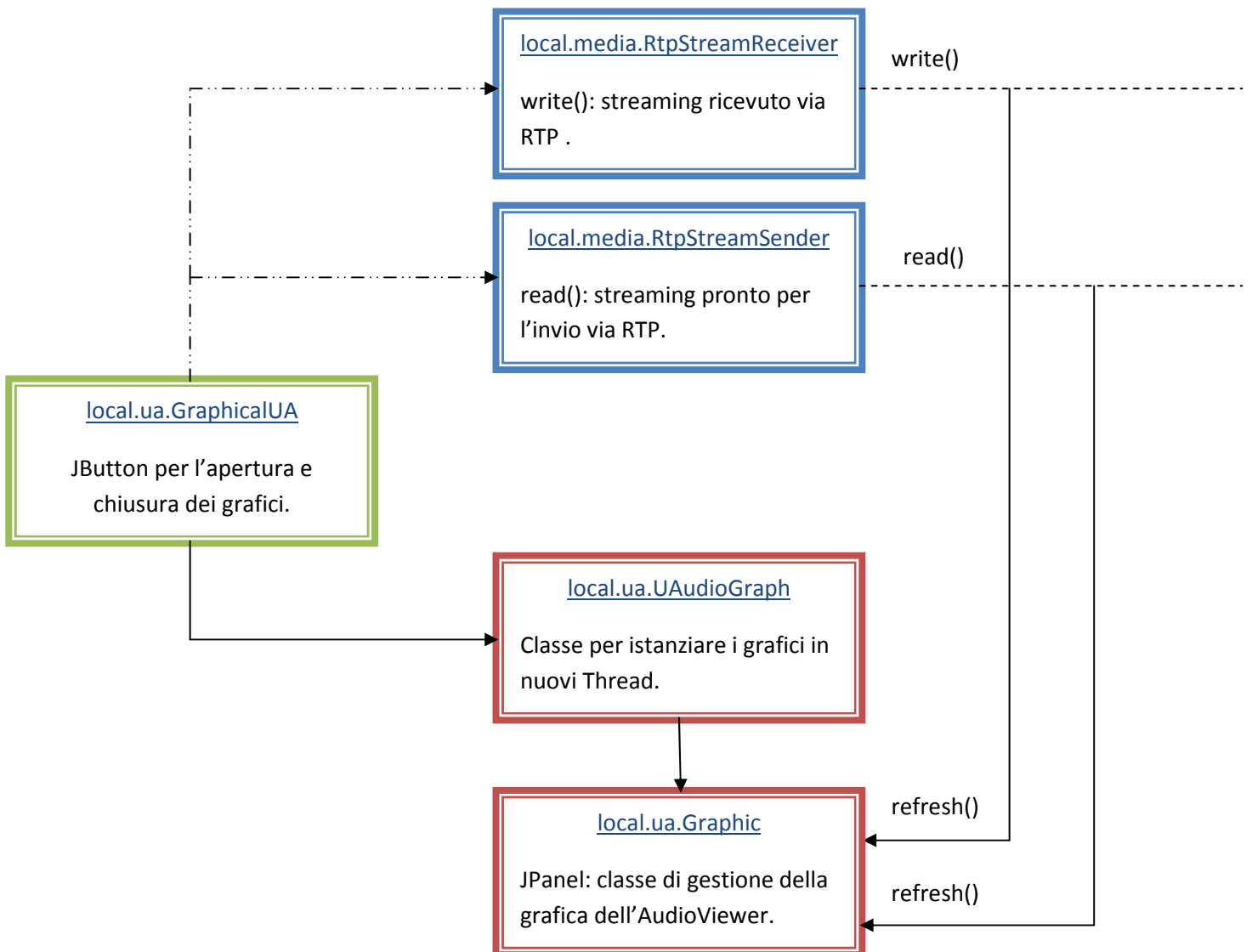


Audio catturato dal microfono di Alice



Audio ricevuto via RTP da Alice

Variazioni apportate a MjSIP



Conclusioni

Il progetto svolto per il corso di Reti di Telecomunicazioni C ha dato buoni risultati per quanto riguarda la velocità ed il sincronismo tra voce e sua graficazione mantenendo buoni valori di utilizzazione della CPU.

Tale risultato è stato ottenuto istanziando nuovi thread per la gestione di ogni singolo grafico e utilizzando il meccanismo dei listener per il refresh dei valori stessi. Il grafico è disegnato velocemente tramite l'utilizzo degli strumenti messi a disposizione da Java2D.

Problemi riscontrati: nella fase di test si erano riscontrati problemi di rallentamento nella graficazione dello stream in ricezione. Tale problema, già noto a priori, è stato risolto scartando un pacchetto a caso ogni 20 ricevuti. Con tale valore l'effetto della perdita di un pacchetto non interferisce sulla qualità della comunicazione.