

E. Richiardone
s126892

Programmazione di sistema
Progetto del corso - documentazione

Prof. G. Malnati
Politecnico di Torino
Ingegneria informatica – Terza Facoltà
febbraio 2007

Descrizione del funzionamento

Ermes – così è stato chiamato il programma per il progetto del corso – è un applicativo di rete per la trasmissione di un flusso audio e di un flusso video tra due o più sistemi Windows.

L'applicativo può operare in due modalità, a seconda della sua funzione:

- modalità server: esso cattura il flusso audio dal microfono e genera un flusso video da successive immagini dello schermo o di una parte di esso.
- modalità client: si ricostruiscono i due flussi e vengono presentati in sincronismo.

I flussi utilizzano il protocollo *UDP* e vengono trasmessi ad un indirizzo multicast su due porte differenti; così facendo i client possono essere molteplici per uno stesso server senza sovraccaricarlo. Assieme al flusso audio è inviata l'informazione sulla posizione del cursore del mouse. Oltre all'informazione della posizione, anche l'icona del cursore è trasmessa, ogni volta che il cursore cambia e periodicamente; se l'informazione non è giunta, un punto lo sostituisce.

Il flusso video è compresso secondo un algoritmo che all'immagine intera della cattura (il cosiddetto *keyframe*) intervalla una codifica differenziale dei cambiamenti; tale codifica differenziale analizza due catture successive e, suddividendole in blocchi da 16x16 pixel, salva l'informazione sui blocchi in cui vi sono notevoli cambiamenti in forma compressa.

Il tutto è regolabile dalla schermata di configurazione. E' infatti possibile:

- impostare l'area dello schermo da catturare tramite selezione con l'ausilio del mouse oppure digitando le coordinate, può anche essere l'intera schermata.
- definire l'indirizzo multicast e le porte per i due flussi di informazione;
- impostare la banda massima da utilizzare, oppure definire a mano i differenti parametri di compressione video: qualità dei keyframe, intervallo fra keyframe successivi, frequenza di rinfresco (*fps*).

In modalità client è possibile configurare:

- indirizzo multicast e porte per i due flussi di informazione;
- tempo di attesa iniziale (*playout buffer*): variando questo valore si permette un maggiore o un minor tempo di accettazione dei pacchetti.

Il programma è stato realizzato scrivendo diversi moduli di base e una parte centrale che li istanzia e che si occupa dell'interazione grafica con l'utente; il tutto è stato scritto in C# utilizzando Visual Studio 2003 e 2005. Per il funzionamento è quindi necessario installare l'ambiente di esecuzione *.NET*; altre dipendenze sono le librerie di sistema *winmm.dll*, *gdi32.dll* e *user32.dll*, comunque presenti senza installazione da Windows 2000.

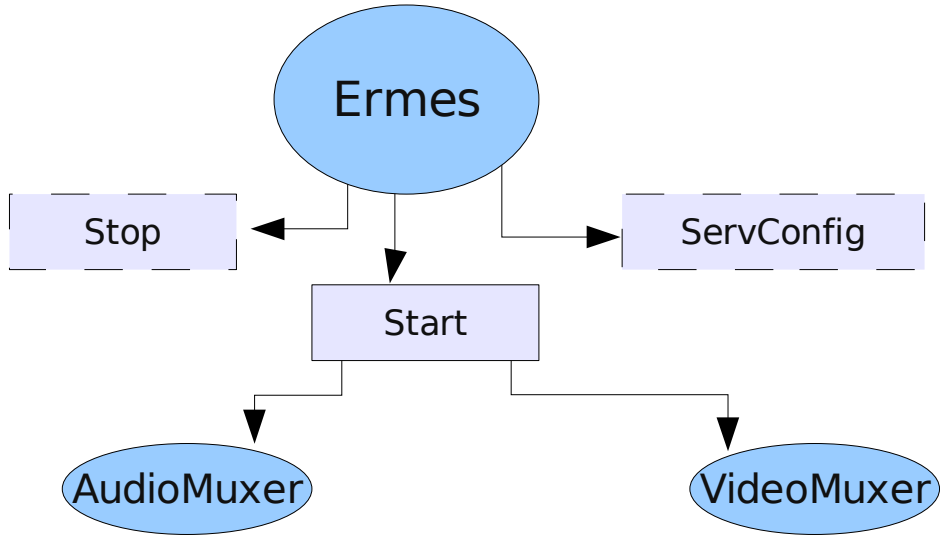
Componenti del programma

Ad ogni modulo corrisponde uno o più namespace; ecco l'elenco dei componenti con le classi:

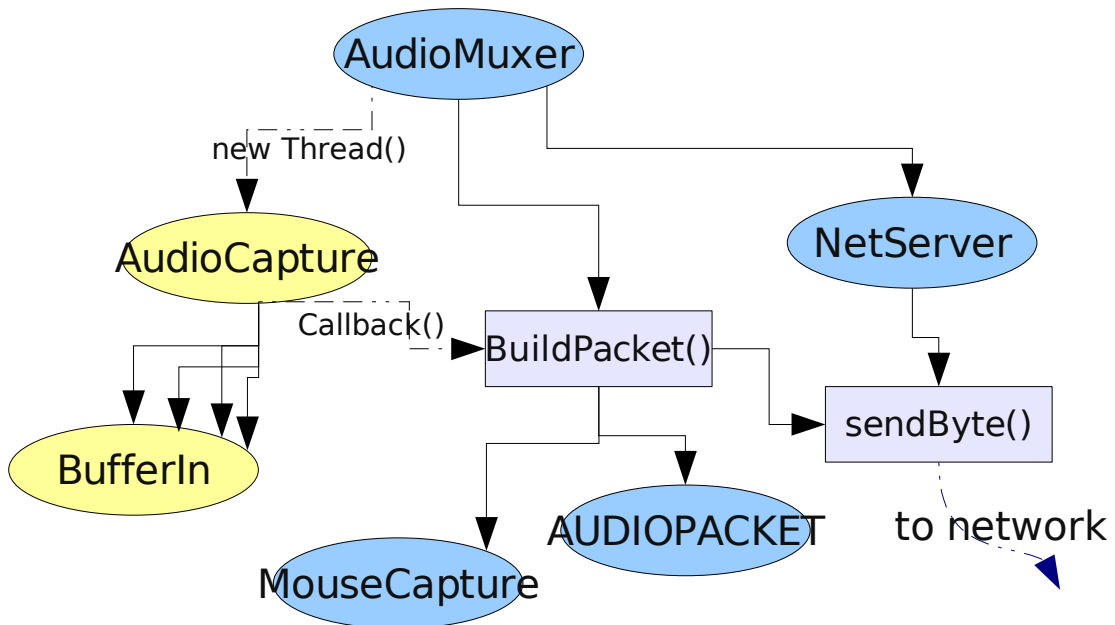
<i>modulo</i>	<i>namespace.class</i>	<i>funzionalità</i>
Audio	AudioLowLevel.Device AudioLowLevel.Wave	scelta del dispositivo audio raggruppa le chiamate p/invoke a <i>winmm.dll</i>
	AudioIn.AudioCapture AudioIn.BufferIn AudioIn.MouseCapture	cattura e gestisce 10 buffer d'ingresso audio rappresenta un buffer d'ingresso audio cattura l'icona del mouse, utilizza <i>user32.dll</i>
	AudioOut.AudioPlayer AudioOut.BufferOut	inizializza e gestisce 2 buffer d'uscita audio rappresenta un buffer d'uscita audio
Mux	Mux.AudioDemuxer Mux.AudioMuxer Mux.VideoDemuxer Mux.VideoMuxer Mux.QueueObj	passa i dati di rete al player, dirige il sinc riceve i blocchi audio e li impacchetta passa i dati di rete al player secondo il sinc riceve i blocchi video e li impacchetta oggetto utilizzato dal sincronismo
Network	Network.NetClient Network.NetServer Network.AUDIOPACKET Network.VIDEOPACKET	client UDP multicast, ricostruisce i frammenti server UDP multicast costruisce e scompone un pacchetto audio costruisce e scompone un pacchetto video
Video	Video.GDICapture Video.VideoCapture Video.VideoCompression Video.VideoPlayer	raggruppa le chiamate p/invoke a <i>gdi32.dll</i> effettua la cattura e comprime i frame algoritmo di compressione, codice <i>unsafe</i> mantiene e presenta in successione i frame
Ermes	Ermes.About Ermes.ClientConfig Ermes.configStr Ermes.Ermes Ermes.Help Ermes.LoadForm Ermes.SelectWin Ermes.ServerConfig	semplice form informativo presenta le impostazioni del client struttura contenente tutte le impostazioni form principale, riceve gli event handler semplice form di aiuto semplice immagine di avvio form trasparente per selezione area di cattura presenta le impostazioni del server

Funzionamento schematico del server

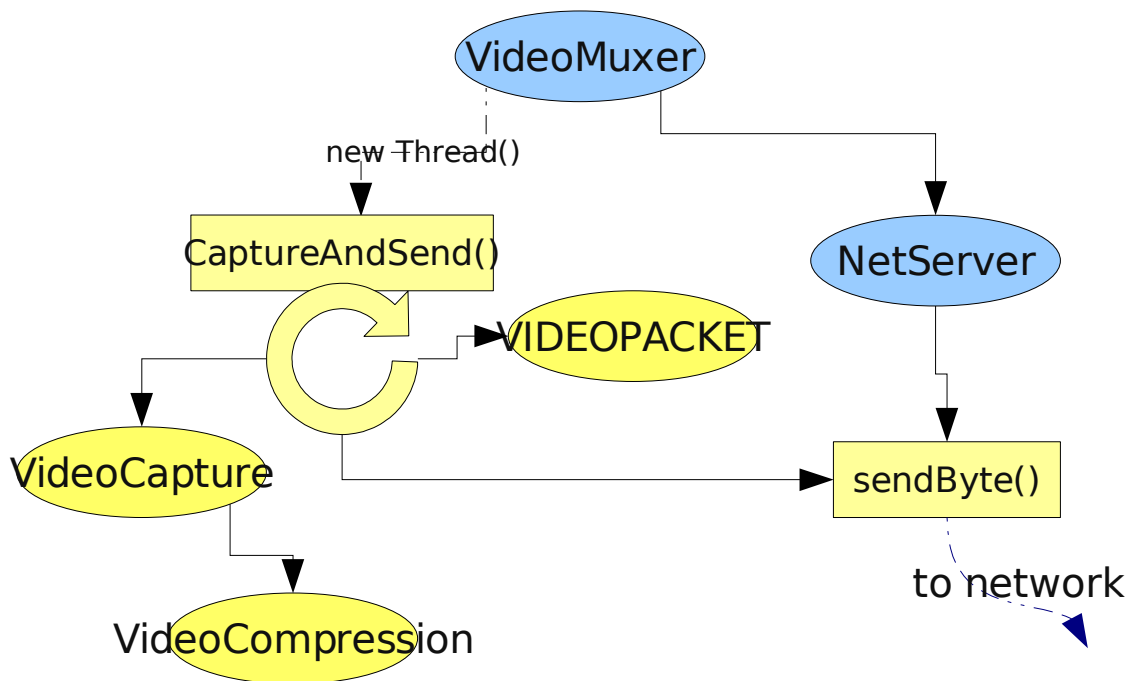
Classi chiamate dal form principale:



Dettaglio di AudioMuxer:

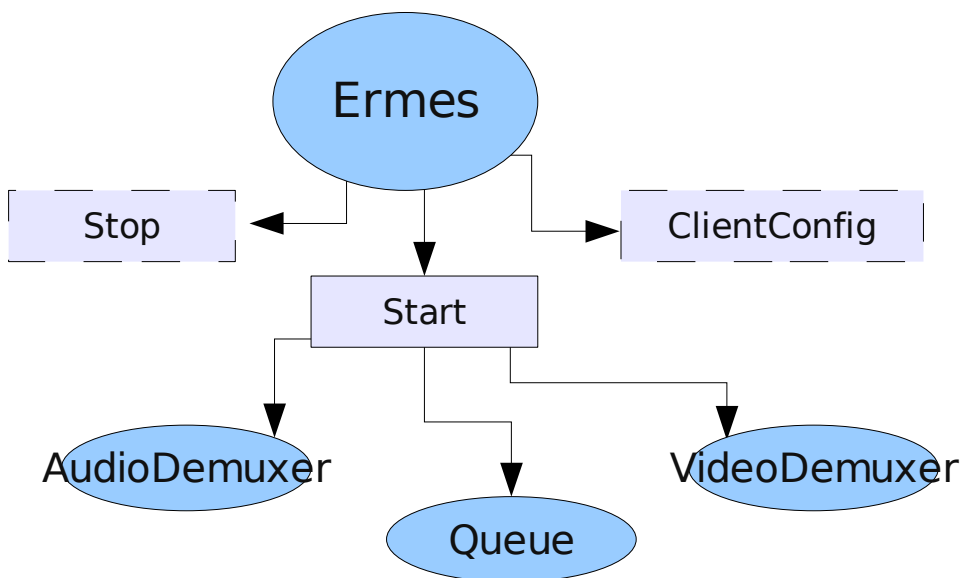


Dettaglio di VideoMuxer:

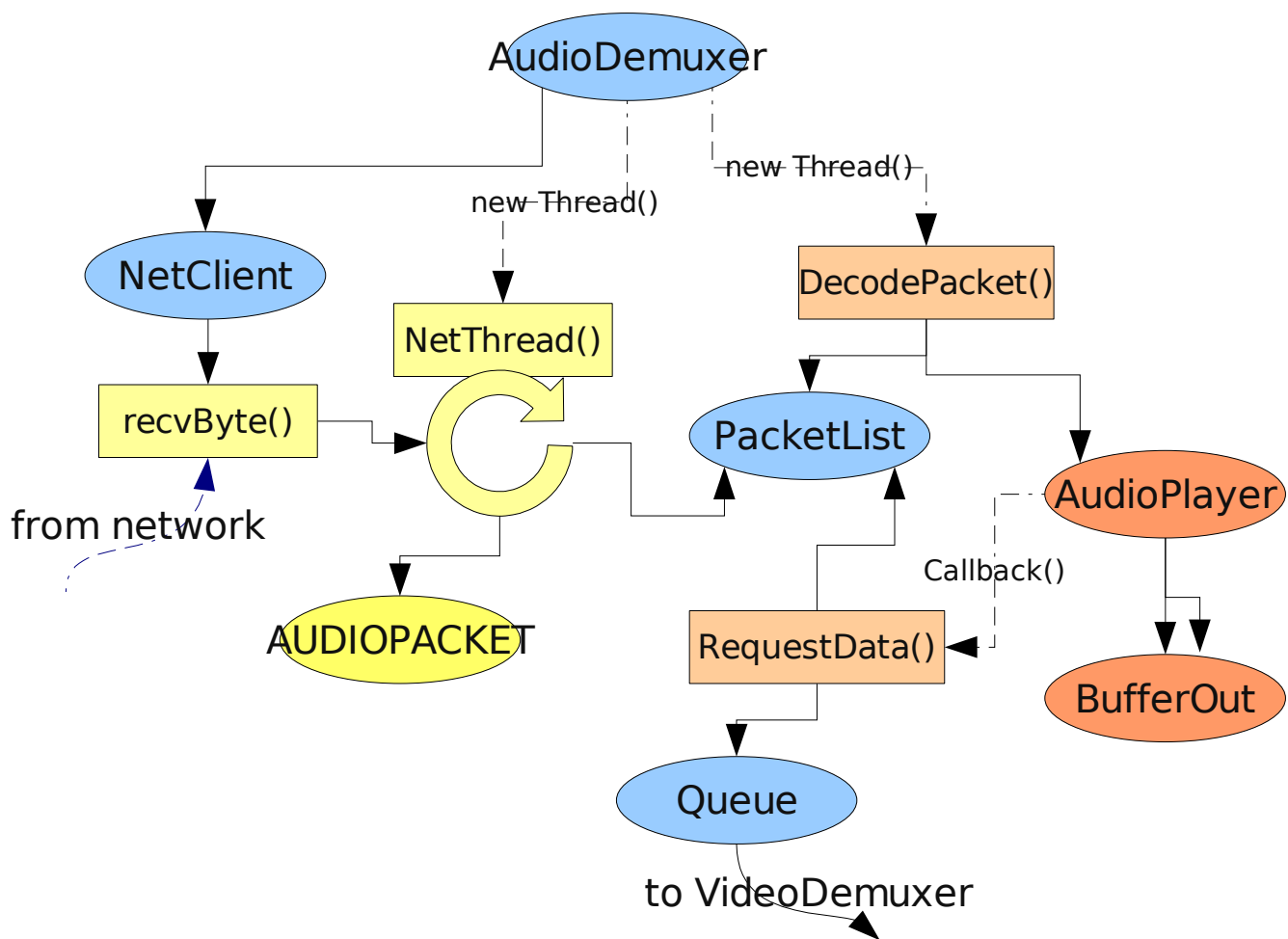


Funzionamento schematico del client

Classi chiamate dal form principale:



Dettaglio di AudioDemuxer:



Dettaglio di VideoDemuxer:

