

Laboratorio di Reti
Progetto AA. 2006-2007

Fabio Quinzi 0000210574
Alfredo Saglimbeni
Gabriele Verardi 0000119364

Introduzione:

Lo sviluppo del progetto ha interessato tre file:

- ProxyReceiver.c
- ProxySender.c
- ProxyUtil.c

All'interno di ProxyUtil.c abbiamo definito variabili, strutture e funzioni globali, che abbiamo utilizzato per implementare le funzionalità dei due Proxy. Deleghiamo ai commenti all'interno del codice la descrizione dettagliata di strutture e variabili, mentre tratteremo in sede la descrizione delle funzioni.

ProxySender.c contiene il codice relativo alla gestione dell'instradamento dei pacchetti al ritardatore, alla gestione della ricezione degli ack, e all'eventuale rinvio dei pacchetti nel qual caso si siano verificati problemi durante la trasmissione.

ProxyReceiver.c, dualmente a ProxySender.c si occupa della ricezione dei pacchetti dal ritardatore, della gestione del loro inoltramento al ricevitore e dell'invio degli ack.

Comportamento dei Proxy, vita dei pacchetti e scelte progettuali:

La corretta sequenza di startup, dopo aver compilato i file attraverso l'apposito make file, è:

- 1) Ritardatore.exe
- 2) Receiver.exe
- 3) ProxyReceiver.exe
- 4) ProxySender.exe
- 5) Sender.exe

Ritardatore, Sender e Receiver se lanciati senza parametri utilizzeranno quelli di default. Altrimenti è possibile impostare manualmente IP e porte.

I pacchetti vengono generati dal Sender, e inoltrati al ProxySender. Quest'ultimo si occupa, tra l'altro, di smistare i pacchetti sulle tre connessioni instaurate con il ritardatore, in modalità non bloccante. Ovvero, tramite la Select, ogni volta che si hanno byte nell'FD del socket, questi vengono spostati in un buffer interno del ProxySender.

Una volta acquisiti i byte necessari, viene ricomposto il datagram ed inviato al Ritardatore.

La scelta del canale avviene tramite la funzione "selectChannell", che monitora continuamente la situazione dei canali. Pacchetto dopo pacchetto, ad ogni canale viene assegnato un coefficiente di qualità, ottenuto tenendo conto di un punteggio e della valutazione di alcuni casi tipici.

Il punteggio, assegnato dalla funzione "keyquality", viene calcolato tenendo conto del delay medio dei pacchetti e del numero di timeout collezionati dal canale.

Prima dell'invio vengono memorizzati in una struttura chiamata "pCheck" le informazioni riguardanti il pacchetto (l'istante in cui viene inviato, il canale utilizzato, e una variabile di controllo che segnala la corretta ricezione del pacchetto da parte del ProxyReceiver). Inoltre, viene mantenuto in memoria anche l'intero datagram, che verrà reinoltrato in caso di errore.

Terminate queste operazioni, il pacchetto viene inviato al Ritardatore, e vengono aggiornati alcune informazioni sulla qualità dei canali.

Il passo successivo è controllare la presenza di ack ricevuti dal ProxyReceiver, relativi a pacchetti precedentemente inviati. Successivamente viene aggiornato l'array di pCheck con le nuove informazioni acquisite, e viene ripetuto lo stesso procedimento per ogni pacchetto ceduto dal Sender .

Il Ritardatore ricevuti i pacchetti agisce su di loro imponendogli un ritardo o lasciandoli semplicemente proseguire verso il ProxyReceiver.

Il ProxyReceiver, così come il ProxySender, riceve i pacchetti in modalità non bloccante, prendendo i byte dall'FD del socket non appena disponibili, per poi ricostruire il datagram non appena sono in quantità sufficiente.

Ottenuto il datagram, l'ack relativo a quest'ultimo viene spedito al ProxySender e successivamente il carico utile di datagram viene spedito al Receiver.

Si ripete il tutto per ogni pacchetto ricevuto dal ProxySender.

14/11/2007