

Specifiche del Progetto “trasporto affidabile multi-percorso” per Reti di Calcolatori a.a. 2011/12, prof. Vittorio Ghini

Scadenza consegna, entro fine febbraio 2013.

(Versione Probabilmente Definitiva, 16 aprile 2012)

Scenario:

Si vuole realizzare un sistema come descritto nella figura 1 qui sotto, con

- i due processi Sender e ProxySender che eseguono su uno stesso host,
- i due processi Receiver e ProxyReceiver che eseguono su uno stesso host, ed
- il processo **Ritardatore** che esegue su un altro host.

Il **Sender** svolge il classico ruolo di client TCP ed instaura una connessione TCP con il **ProxySender** che era in attesa sulla porta TCP numero 59000. Il **Receiver** svolge il ruolo di server TCP ed attende sulla sua porta 64000 che il **ProxyReceiver** gli si connetta per instaurare una connessione TCP.

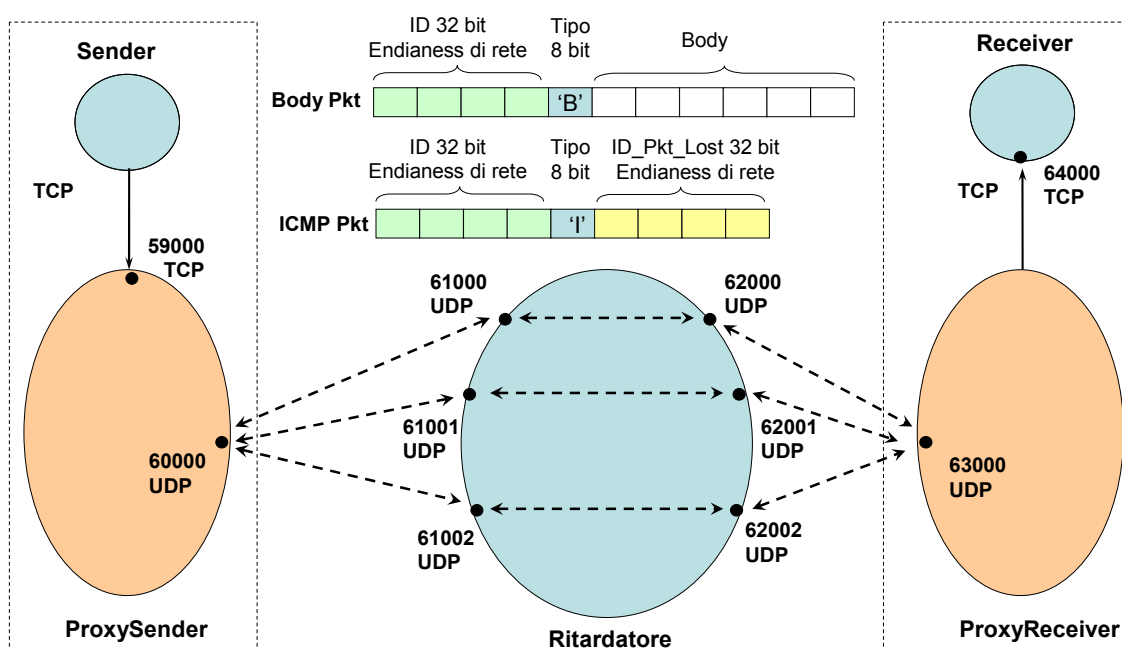


Figura 1: I processi componenti il sistema.

Il ProxySender usa un socket UDP con cui può comunicare (ricevere e spedire) con 3 porte note UDP del Ritardatore. Il ProxyReceiver usa un socket UDP con cui può comunicare (ricevere e spedire) con altre 3 porte note UDP del Ritardatore.

Il Ritardatore mantiene una sorta di **canale tra ciascuna coppia di porte UDP**, una dal lato ProxySender e l'altra dal lato ProxyReceiver. Il Ritardatore inoltra al ProxyReceiver tutto quello che riceve da ciascuna porta UDP dal lato ProxySender utilizzando in trasmissione l'analoga porta UDP dal lato ProxyReceiver. Analogamente nel verso opposto, cioè il Ritardatore inoltra al ProxySender tutto quello che riceve da ciascuna porta UDP dal lato ProxyReceiver utilizzando in trasmissione l'analoga porta UDP dal lato ProxySender.

Il Ritardatore emula il funzionamento di una rete reale **tra ciascuna coppia di porte UDP**, ritardando di alcuni millisecondi i pacchetti UDP prima di inoltrarli o anche scartandoli senza inoltrarli. In ciascun suo canale UDP il Ritardatore può scartare alcuni pacchetti isolati oppure può scartare una sequenza consecutiva di pacchetti. Tipicamente, la sequenza di pacchetti scartati dura tra i 2 e i 4 secondi. Le sequenze di pacchetti non scartati durano tra i 5 e i 20 secondi. All'interno delle sequenze non scartate, alcuni pacchetti isolati possono essere scartati.

Quando il Ritardatore scarta un pacchetto che ha ricevuto (su una sua porta) dal ProxySender o dal ProxyReceiver allora il Ritardatore invia al Proxy da cui ha ricevuto il pacchetto scartato un messaggio di avviso di perdita del pacchetto detto ICMP, incapsulato in UDP. Anche questi pacchetti ICMP possono essere ritardati e scartati, quindi alcune perdite di pacchetti potrebbero non essere notificate al mittente.

Il Sender trasmette verso il ProxySender un flusso di dati in quantità imprecisata sfruttando la connessione TCP instaurata. Il ProxySender riceve dal Sender questi dati e deve trasmetterli man mano verso il ProxyReceiver sfruttando esclusivamente i tre flussi UDP che passano attraverso il Ritardatore.

Il ProxyReceiver riceve dai 3 flussi UDP del Ritardatore i pacchetti provenienti dal ProxySender e deve ricostruire il flusso dati originale trasmesso dal Sender e spedirlo man mano al Receiver mediante la connessione TCP che connette il ProxyReceiver al Receiver.

Obiettivo del Progetto

L'obiettivo di ProxySender e ProxyReceiver è di realizzare un protocollo che sfrutti i datagram UDP, per trasferire invariato il flusso proveniente dal Sender fino al Receiver attraverso i 3 flussi UDP del Ritardatore, nonostante i ritardi e le perdite introdotte dal Ritardatore stesso.

IL PROTOCOLLO DEVE ASSICURARE L'INTEGRITA' DEL FLUSSO.

Poiché i datagram UDP possono essere scartati dal Ritardatore, le due applicazioni ProxySender e ProxyReceiver hanno il compito di realizzare un meccanismo di controllo e di eventuale ritrasmissione dei datagram UDP.

E' considerato **un parametro di efficacia del sistema il tempo che trascorre tra il momento in cui ciascun byte viene trasmesso dal Sender al momento in cui quel byte arriva al Receiver.**

Formato dei Pacchetti

Il Ritardatore accetta solo pacchetti UDP in cui i primi 5 bytes sono così organizzati:

- i primi 4 bytes contengono un intero senza segno memorizzato secondo l'endianess predefinita di rete (big endian). Quell'intero rappresenta un numero (ID) che identifica univocamente il pacchetto.
- il 5° byte è un char senza segno che contiene il valore **'B'** (Body).
- il contenuto del resto del pacchetto può essere qualunque.

Qualunque altro pacchetto viene scartato senza nessun avviso.

I pacchetti non scartati vengono inoltrati invariati.

Quando il Ritardatore scarta un pacchetto, il Ritardatore può spedire al Proxy mittente del pacchetto un messaggio ICMP così fatto:

- i primi 4 bytes contengono un intero senza segno memorizzato secondo l'endianess predefinita di rete (big endian) che rappresenta un numero (ID) che identifica univocamente il pacchetto generato dal Ritardatore.
- Il 5° byte è un char senza segno che contiene il valore 'I' (ICMP)
- i successivi 4 bytes contengono un intero senza segno memorizzato secondo l'endianess predefinita di rete (big endian). Quell'intero memorizza l'ID del pacchetto che è stato scartato.

Applicazioni utilizzate

Dei cinque applicativi, uno (il Ritardatore) viene già fornito assieme a queste specifiche, altri due (Sender, Receiver) verranno forniti al momento della demo in laboratorio, mentre gli altri due (ProxySender ProxyReceiver) debbono essere implementati dagli studenti.

Le applicazioni Sender e Receiver usate durante la demo sono implementate in maniera tale da fornire traffico TCP in diversi modi e soprattutto effettuano dei controlli per verificare l'integrità dei dati ricevuti e monitorare i tempi di trasmissione. In particolare, verificano che i pacchetti siano ricevuti rispettando l'ordine con cui sono stati creati. NON è ammesso che un pacchetto sia consegnato fuori ordine.

Vincoli di progetto:

L'elaborato dovrà essere svolto in linguaggio ANSI C (devono essere passati al compilatore almeno questi parametri: `-ansi -Wall -Wunused -pedantic`), si dovrà privilegiare la portabilità su sistemi POSIX.

Lo scopo del progetto è acquisire conoscenza delle tecniche di progettazione mediante I/O MULTIPLEXING, e per questo motivo è vietato l'utilizzo di tecniche di progettazione basate su I/O asincrono e segnali. In particolare:

- 1) Non possono essere usate le `signal()`, `sigvec()`, `sigaction()`, e simili.
- 2) Non possono essere usate named pipe (dette anche FIFO), cioè quelle create con `mknod(,,)`.
- 3) Non possono essere usati i socket unix named, cioè creati con `socket(AF_UNIX, ,)`
- 4) Possono essere usate le pipe, cioè quelle create con `int pipe(int fildes[2])`, ricordando però che le pipe non permettono di essere utilizzate mediante le invocazioni alle system call `send` e `recv`, e quindi non possono essere utilizzati i comportamenti definiti dai flags `MSG_DONTWAIT`, `MSG_PEEK`, etc.
- 5) Possono essere usati i socket unix anonimi, cioè quelli creati mediante la system call `socketpair(AF_UNIX,SOCK_STREAM,0)`.
- 6) Potrebbero essere usati pthread e processi ausiliari, se ritenuti necessari evitando però di utilizzarne troppi contemporaneamente per evitare sovraccarico di memoria o di CPU.
- 7) Non possono essere sfruttate tecniche di compressione dei dati.

Valutazione del progetto

La valutazione del progetto realizzato si baserà su diversi criteri, in particolare terrà conto di:

- impostazione del progetto,
- performance in termini di:
 - affidabilità (integrità del flusso dati TCP). E' INDISPENSABILE !!!!
 - interattività realizzata (latenza end-to-end dei bytes),
 - uso di CPU,
 - occupazione di memoria,
 - consumo di bandwidth (calcolata dal Ritardatore, che conta i byte scambiati),
- chiarezza del codice realizzato,
- numero di componenti del gruppo.

Fase di test - l'emulatore di rete Ritardatore

Per aiutare l'analisi prestazionale, il Ritardatore colleziona informazioni sui dati che lo hanno attraversato. In particolare, killando il Ritardatore (premere CTRL+C) vengono visualizzati il numero di byte trasmessi.

Il Ritardatore prende in input a riga di comando alcuni parametri, indicati qui di seguito. In grassetto è indicato il valore di default.

- Indirizzo IP del ProxySender (**127.0.0.1**)
- Porta UDP del ProxySender (**60000**)
- Prima Porta UDP del Ritardatore dal lato ProxySender (**61000**)
- Prima Porta UDP del Ritardatore dal lato ProxyReceiver (**62000**)
- Indirizzo IP del ProxyReceiver (**127.0.0.1**)
- Porta UDP del ProxyReceiver (**63000**)
- Percentuale Pacchetti Scartati (moltiplicata per 100. (un intero tra 0 e 100)) (**15**)
Si noti che scegliendo come percentuale di perdita il valore 0 non viene perso nessun pacchetto (né isolatamente né in sequenza) e quindi questa modalità è la più adatta a testare il progetto durante le prime fasi di implementazione.
- Seme di Inizializzazione del Generatore di numeri casuali (un intero. -1 indica che viene inizializzato casualmente) (**-1**).
- Output di Debug (un singolo carattere, y o n) (**y**)

Se non viene specificato nessun parametro, il Ritardatore usa i parametri di default.

Demo del progetto:

Quando un gruppo riterrà' di avere concluso il progetto, prenderà appuntamento, via email, per la demo del progetto. **Il codice prodotto e la relazione NON devono essere inviati via mail ma solo presentati al momento della demo.** Tale demo consiste nella valutazione del prodotto realizzato. Si svolge utilizzando i calcolatori dei laboratori didattici del corso di studio. Il gruppo si presenterà portando una stampa del codice realizzato, ed una breve relazione (5-10 pagine) che descriva come il progetto e' stato realizzato, i dettagli del protocollo realizzato, **e quali test sono stati effettuati per valutare le prestazioni.**

Infine, **il codice del progetto verrà compilato ed eseguito sulle macchine dei laboratori, per valutarlo.**

Voto del progetto

Il voto ottenuto per il **progetto** di Reti di Calcolatori, concorrerà a formare il voto del Corso di Reti di Calcolatori mediante la media aritmetica con il voto ottenuto nello scritto del corso di Reti di Calcolatori.

Scadenza del progetto

Il progetto di Reti di Calcolatori per l'anno accademico 2011/12 dovrà essere consegnato entro fine febbraio 2013.