

Progetto di "Laboratorio di Programmazione di Rete" a.a. 2007/2008

"QoS per VoIP sfruttando Multi-Homing"

sviluppatori: Alberti Michele, Pomili Luigi.

introduzione:

Questo documento vuole descrivere, in breve, l'implementazione degli applicativi MobileLoadBalancer e FixedLoadBalancer (rispettivamente "mobileLB" e "fixedLB") oggetto della proposta di progetto.

Viene dato maggiore spazio alle scelte e politiche implementative riguardanti le strutture dati utilizzate e in generale la struttura del codice.

strutture di dati:

Le specifiche del progetto prevedono che il mobileLB e il fixedLB comunichino scambiandosi pacchetti UDP che attraversano il processo Monitor (emulatore della rete reale) il quale puo' scartarli/ritardarli.

I nostri due applicativi quindi si interfacciano verso il Monitor tramite due porte UDP ciascuna (in realta' potrebbero essere "n"). Abbiamo percio' pensato di mantenere tutte le informazioni possibili riguardanti le porte nella struttura dati definita "port_monitoring". Queste strutture sono poi aggregate a formare una lista di porte UDP.

Il pacchetto scambiato tra i nostri applicativi e' nominato "udp_packet" che mantiene i dati prelevati dalla rispettiva applicazione (AppMobile o AppFixed) piu' altri byte che servono ad identificare il tipo di pacchetto (type_packet), a mantenere il tempo trascorso nei buffer dell'applicazione (time_buffered) e a determinare il ritardo temporale introdotto dalle applicazioni (the_mighty_fix).

Il campo "type_packet" risulta necessario perche' tra mobileLB e fixedLB non vengono scambiati soltanto i dati veri e propri ma anche "strutture" di controllo riguardanti le prestazioni della rete. Si parla cioe' della struttura "stat_notify".

Da notare anche la struttura "mon_notify" utile per il mantenimento delle notifiche di gestione provenienti dalla connessione TCP del Monitor.

Notare l'utilizzo della direttiva **`__attribute__((packed))`** utilizzato per tutte quelle strutture rappresentanti pacchetti che vanno trasmessi. Inoltre si e' cercato di mantenere il piu' possibile basso l'overhead sui dati, evitando uno spreco eccessivo di banda.

mobileLB e fixedLB:

I due applicativi sviluppati sono molto simili tra di loro, speculari.

Entrambi ricevono i pacchetti dalla rispettiva applicazione, spediscono e ricevono pacchetti dal Monitor. Ovviamente la grossa differenza tra i due e' la presenza o meno della gestione della comunicazione TCP con il Monitor.

Il mobileLB prevede un frammento, non piccolo, di codice per la gestione delle notifiche che provengono dalla connessione TCP con il Monitor, la ricezione di quest'ultime viene fatta in maniera da evitare la possibile attesa dei dati sul canale. In caso di notifica di tipo "A" (ack) il mobileLB elimina il pacchetto notificato "spedito" dal Monitor facendo in questo modo spazio per quelli provenienti dall'AppMobile. Viceversa, in caso di notifica "N" (nack) il mobileLB setta il pacchetto, ancora bufferizzato, come da "rispedire" in modo da forzare il rinvio appena il canale UDP con il Monitor risultasse libero.

Ovviamente questo trattamento viene riservato anche ai pacchetti di tipo "stat_notify", utilizzati per comunicare al fixedLB le prestazioni della rete. Va notato come soltanto il mobileLB spedisca questo tipo di pacchetto, in modo

speculare soltanto il fixedLB li riceve.

Le notifiche al fixedLB vengono spedite ogni 50 pacchetti spediti all'AppMobile, in modo da decifrare abbastanza tempestivamente il cambiamento di comportamento da parte della rete senza però causare eccessivo spreco di banda. Ad ogni porta aperta (canale) nel Monitor corrisponde un pacchetto di notifica da trasmettere; questo poi permette al fixedLB di scegliere, sulla base delle statistiche, quale canale sia il migliore per la comunicazione.

Si noti come ogni notifica sia contraddistinta da un identificativo di pacchetto e da uno di sequenza di pacchetti. Questo per permettere poi al fixedLB di distinguere differenti sequenze di notifiche nel tempo, e più dettagliatamente differenti pacchetti.

I pacchetti di notifica sono stati la nostra risposta al problema dello scenario proposto nel progetto. La mancanza delle notifiche da parte del Monitor dal lato fixedLB portano, senza alcuna forma di protocollo o politica, ad una perdita di traffico che va dal 15% minimo ad un massimo di 18%.

Tramite i pacchetti di notifica che periodicamente il mobileLB trasmette al fixedLB, quest'ultimo può comprendere l'andamento della rete (percentuale dei pacchetti persi) e applicare in questo modo una politica di ritrasmissione più o meno pesante.

In particolare, il fixedLB riceve notifica di quanti pacchetti da lui trasmessi sono stati ricevuti dal mobileLB. Su questa base viene poi determinata una "goodness" della rete e quindi la percentuale di ritrasmissione.

Oltretutto, i pacchetti "stat_notify" sono utilizzati anche per far comprendere al fixedLB i canali rimasti aperti e conseguentemente le nuove porte aperte. Il fixedLB è progettato per rendersi conto dell'apertura di nuove porte anche al momento della ricezione dei pacchetti dal Monitor.

Quest'ultimo non si limita soltanto nello scartare i pacchetti, ma può anche ritardarli. L'idea è quella di bufferizzare i pacchetti provenienti dal Monitor (trasmessi dall'opposto balancer) attendendo quel tempo necessario per far arrivare quelli ritardati.

L'idea da noi implementata è quella di ricostruire il tempo perso dai pacchetti all'interno del Monitor, man mano affinando sempre di più l'approssimazione dell'istante di spedizione dei pacchetti effettuata dall'opposta applicazione (AppMobile o AppFixed).

In generale, quello che si fa è uno studio sulla distanza temporale di ricezione dei pacchetti; in questo modo si riesce a "ricostruire" l'istante "zero", l'invio del primo pacchetto. Ricostruito in modo apprezzabile l'istante "zero", si riesce a prevedere, in modo abbastanza preciso, il momento di arrivo di ogni pacchetto e quindi anche il tempo di attesa più indicato.

note conclusive:

La versione presentata ne prevede una precedente che fa utilizzo (in più) di un protocollo basato su pacchetti di "ACK" e ritrasmissioni. Questa si comporta, in termini prestazionali, in modo migliore. Si riesce a perdere anche soltanto il 4% del traffico, lato fixedLB->mobileLB. Il problema di questa versione del software risiede nella quantità di overhead risultante, troppo.

La versione ultima, oggetto di questa documentazione, non riesce ad ottenere le stesse prestazioni della prima a causa di una politica di "resending" più restrittiva. L'overhead però risulta molto minore, circa il 30%.

Il fatto però che la percentuale delle perdite si abbassi sotto la soglia del 10% soltanto a seguito di un massiccio "resending", denota come la politica di scarto/ritardo del Monitor sia *[fortemente]* restrittiva.