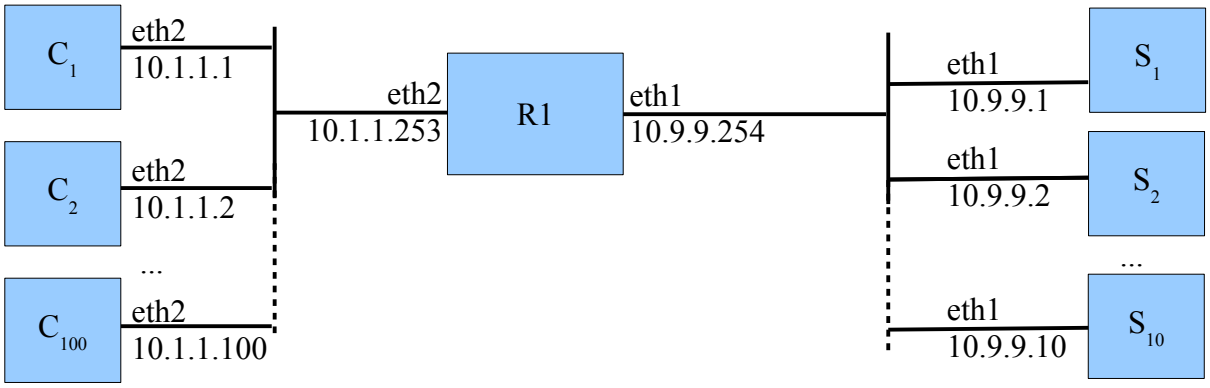
Laboratorio di Amministrazione di Sistemi T

Prova pratica – 15 febbraio 2016

Descrizione generale del problema

Si consideri la rete illustrata in figura, in cui i blocchi C_i (per i compreso tra 1 e 100) rappresentano workstation disponibili all'utenza, da usare come client, e i blocchi S_j (per j compreso tra 1 e 10) rappresentano dispositivi che ospitano server LDAP



Di norma, i client sono raggruppati in 10 blocchi di 10 macchine con indirizzi consecutivi, e ogni blocco è servito da un server. LDAP è utilizzato per memorizzare statistiche di utilizzo dei client, che caricano dati periodicamente.

I server LDAP in questo scenario hanno capacità di elaborazione limitata ed è fondamentale, per evitare malfunzionamenti, che i server sovraccarichi vengano esclusi temporaneamente dal pool, reindirizzando le richieste ad altri.

Il router si occupa dello smistamento delle richieste, in modo che i client non sappiano quale server stanno utilizzando, servendosi del packet filter: i client quindi contattano solamente l'indirizzo del router, che provvede a reindirizzare i pacchetti sul server assegnato.

Il router svolge anche il monitoraggio dei server via SNMP per individuare casi di sovraccarico e rinconfigurare lo smistamento client-server in modo opportuno.

File da consegnare

usage.schema (router) – Definire gli attributi LDAP *ip* (testuale), *disco*, *ram* e *timestamp* (interi), e la classe *client* che li contenga obbligatoriamente tutti. Le entry di classe *client* rappresentano l'informazione che la macchina con indirizzo "*ip*" all'istante "*timestamp*" ha una percentuale di spazio occupato *"disco"* sulla partizione / e una quantità "*ram*" di memoria occupata.

update.sh (client) – Questo script rileva la percentuale di spazio occupato sulla partizione / e la quantità di ram occupata sulla macchina su cui viene lanciato, e memorizza i dati in una nuova entry sul server LDAP.
Indicare nei commenti come eseguire automaticamente lo script ogni 5 minuti;

route.sh (router) – Questo script (utilizzabile da *init.sh* e *monitor.sh*) accetta 3 parametri: il primo può essere la stringa "*off*" o l'indirizzo di un server, gli altri due sono l'indice iniziale e finale di un range di client.
Se invocato con "off" (de)configura il packet filter per eliminare eventuali regole di indirizzamento delle richieste ricevute dai client verso un server; se invocato con un indirizzo di server, configura il packet filter per indirizzare tutte le richieste ricevute da un client appartenente al range al server specificato. Esempi:

`route.sh 10.9.9.7 21 30` fa sì che tutte le richiese LDAP provenienti da 10.1.1.21, 10.1.1.22 .... 10.1.1.30 vengano inviate a 10.9.9.7

`route.sh off 21 30` fa sì che tutte le richiese LDAP provenienti da 10.1.1.21, 10.1.1.22 .... 10.1.1.30 non vengano inoltrate ai server

copy.sh (router) – Questo script (utilizzabile da *monitor.sh*) accetta due parametri, entrambi indirizzi ip di server, e copia dal primo al secondo tutte le entry con *timestamp* maggiore di quello memorizzato in */etc/last_timestamp*. Al termine aggiorna il file con l'ora corrente.

init.sh (router) – Questo script deve configurare il packet filter per

- consentire solo il traffico necessario ai vari script di questo testo;
- predisporre la situazione iniziale di inoltro del traffico LDAP tra i blocchi di client e i server (client da 1 a 10 → server 1, client da 11 a 20 → server 2, ecc.)

Fatto ciò, lo script lancia *monitor.sh* e lo controlla, riavviandolo in caso di terminazione accidentale.
Indicare nei commenti come far eseguire questo script al boot.

monitor.sh (router) – Questo script, a ciclo continuo senza mai arrestarsi, rileva via SNMP il carico di tutti i server. Nel predisporre il controllo, si garantisca che possano essere raccolte tutte le risposte nel giro di pochi secondi, senza incorrere in attese prolungate causate dall'indisponibilità di uno o più server.
Se un server (che chiameremo SC) riporta un carico superiore a 2, individua il server col carico più basso (che chiameremo SL), e solo se dal log risulta che SC e SL non sono stati coinvolti in una commutazione negli ultimi 2 minuti:

- individua i client che stanno utilizzando SC e deconfigura l'inoltro del traffico;
- aggiorna SL col contenuto di SC per mezzo di *update.sh*;
- attiva l'inoltro del traffico verso SL per i client che stavano usando SC;
- logga l'evento attraverso syslog sul file /var/log/commutazioni.

Indicare nei commenti:

- come configurare gli agenti snmp dei server per consentire il controllo;
- come configurare il sistema di logging del router