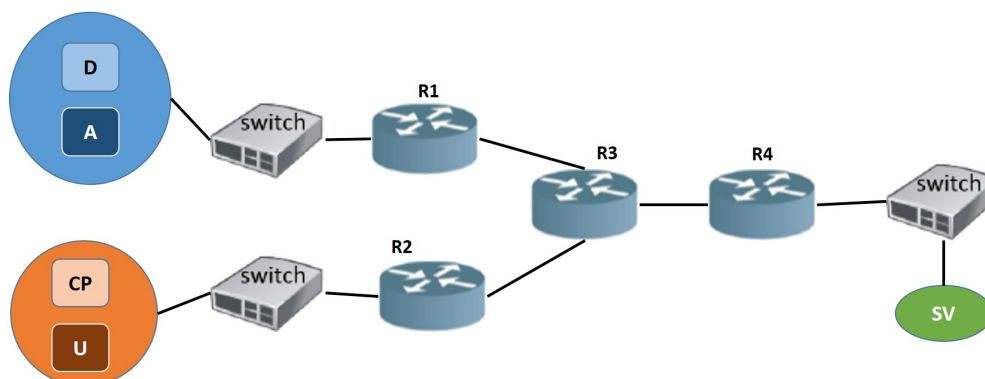


Esame di Laboratorio di Reti di Calcolatori – gennaio 2022

Prova Packet Tracer

La rete di un'azienda ha indirizzo base 188.205.74.0/23 ed è organizzata in tre edifici collegati tra loro da switch e router come mostrato dal seguente schema:



La rete blu comprende due VLAN: **D** (dirigenti) comprendente 28 host, e **A** (amministrazione) comprendente 54 host.

La rete rossa comprende due VLAN: **CP** (catena di produzione) contenente 87 host, e **U** (uffici) che può contenere fino a 151 laptop.

La rete verde **SV** è una VLAN contenente 6 server.

Per tale infrastruttura di rete lo studente è richiesto:

TASK 1 (20 minuti a disposizione)

- Calcolare i parametri di ogni sotto-rete dell'azienda così che sia della dimensione minima sufficiente per contenere il numero di host indicato. Inserire i parametri nel file *grigliaPT.xlsx* da ri-denominare come *griglia_CognomeNome.xlsx* e caricare in <http://upload.di.unimi.it> entro il termine di tempo indicato (autenticandosi con login name e password di Ateneo, cioè @studenti.unimi.it).

TASK 2 (60 minuti a disposizione)

- Scaricare da <http://upload.di.unimi.it> il file *griglia-J22-ITA.xlsx* fornito dalla docente all'inizio del Task, con i parametri di rete.
- Configurare almeno due host per ogni rete come da parametri forniti.
- Configurare opportunamente switch e router in base ai parametri forniti e alla specifica sopra data, così da consentire la corretta comunicazione sia tra host appartenenti alla medesima VLAN sia tra host appartenenti a VLAN differenti. *Non aggiungere altri apparati oltre a quelli mostrati in figura.*
- Configurare i router in modo che usino il protocollo di instradamento OSPF. Si faccia in modo che il protocollo OSPF non inondi di messaggi le sotto-reti non contenenti altri router.
- Configurare un web server in SV così che sia accessibile da tutti.

Modalità di consegna: entro il limite di tempo indicato, lo studente deve consegnare la sola activity, nominata *cognome-matricola.pkt*, via sito <http://upload.di.unimi.it> autenticandosi con login name e password di Ateneo (cioè @studenti.unimi.it).

Prova Java Socket (90 minuti a disposizione)

Si realizzi un sistema distribuito che implementa un servizio di calcolatrice in notazione polacca (prefissa). Il comportamento dei processi è definito dalle seguenti specifiche:

CLIENT:

1. Comunica con il server utilizzando i servizi di TCP.
2. Alla partenza, chiede l'introduzione da tastiera del proprio nickname. Contatta il server presentandosi con il proprio indirizzo di rete e nickname. Poi si pone in attesa di ricevere un messaggio dal server.
3. Richiede l'introduzione da tastiera del numero di operandi (max 5). Dopo di che invia questa informazione al server, identificandosi con il proprio nickname, e ne attende in risposta uno ack.
4. Richiede l'introduzione da tastiera del simbolo di operazione da eseguire (scelto tra +, *, -), e lo invia al server, attendendone in risposta uno ack.
5. Richiede l'introduzione da tastiera degli operandi (numeri reali), nel numero dato in input al punto 3. Quando tutti gli operandi sono stati inseriti, invia al server un messaggio che li contiene. Dopo di che attende il risultato dell'operazione dal server.
6. Ricevuto il risultato, lo stampa a video. Quindi richiede da tastiera se si vuole continuare con altra operazione, oppure terminare.
 - a. Nel primo caso, riprende l'esecuzione dal punto 3
 - b. Nel secondo caso, manda al server una notifica di chiusura della conversazione. Quindi chiude la propria socket e termina.

SERVER:

1. E' un server singolo processo e singolo thread che comunica con i client utilizzando i servizi di TCP.
2. Alla partenza si pone in attesa di richiesta di connessione di un client. Quando riceve un messaggio da un nuovo client, ne inserisce il nickname e l'indirizzo di rete in una propria struttura. Quindi manda al client un riscontro.
3. Successivamente si pone in attesa di ricever dal client il numero di operandi: lo memorizza e invia uno ack al client.
4. Quindi attende di ricevere dal client il simbolo di operazione da eseguire: lo memorizza e invia uno ack al client.
5. Quindi attende di ricevere dal client un messaggio con tutti gli operandi (nel numero indicato al punto 3). Una volta ricevuto, computa l'operazione e invia al client il risultato.
6. Si pone in attesa di messaggio dal client:
 - a. Se il client decide di proseguire con una nuova operazione, il server riprende l'esecuzione dal punto 3.
 - b. Se il client decide di terminare, il server chiude la socket con esso e riprende l'esecuzione dal punto 2.
7. Il server non termina mai.

La definizione del formato dei messaggi, così come i dettagli implementativi non specificati sopra, sono a discrezione dello studente. Tutti i messaggi ricevuti da ogni processo devono essere mostrati a video. Devono essere gestite le situazioni di terminazione improvvisa del processo remoto con cui si sta comunicando.

Modalità di consegna:

Lo studente deve consegnare due file sorgenti cliente.java e server.java in accordo alle seguenti istruzioni; qualunque altro file presente nella consegna non verrà considerato per la correzione.

1. Prima di consegnare, assicurarsi di *salvare il contenuto di tutti i file sorgente* prodotti
2. comprimere i file sorgenti in un *archivio cognome-matricola.zip*; altri formati di compressione non verranno accettati
3. Collegarsi al sito <http://upload.di.unimi.it>
4. Autenticarsi con login name e password di Ateneo (cioè @studenti.unimi.it)
5. Fare upload del *file .zip*

6. Fare logout dal sito.

NB: in caso si procedesse ad effettuare la consegna dell'elaborato più di una volta, concatenare un numero progressivo al nome file, così che sia possibile individuare l'ultima versione. Per la correzione verrà considerata solo l'ultima versione dell'archivio caricata.