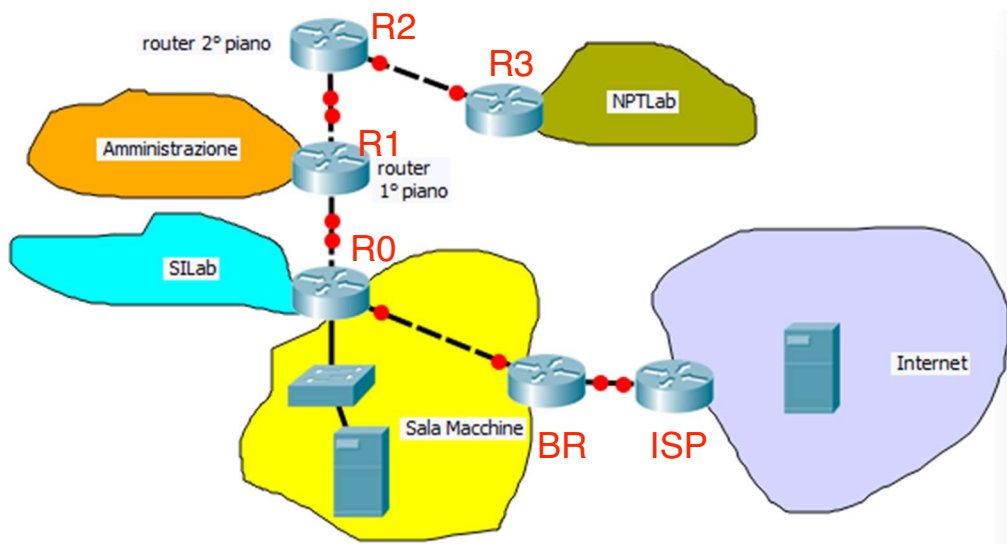


Esame di Laboratorio di Reti di Calcolatori – febbraio 2022

Prova Packet Tracer

Si considerino la rete interna al Dipartimento di Informatica e il suo collegamento a Internet, schematizzati come in figura, dove sono raffigurate le sottoreti del Laboratorio di Reti (NPTLab), dell'Amministrazione, del SILab e di Sala Macchine.



Si progetti e si configuri tutta la rete considerata (inclusa la parte Internet), partendo dalle seguenti assunzioni:

- Si assegni a Internet la rete 84.0.0.0/8 e vi si inseriscano uno host e un server web. Il boundary router di confine alla rete del Dipartimento (che connette Sala Macchine al router di Internet) è collegato al router dello Internet Service Provider con indirizzi nello stesso range. Internet deve essere della massima dimensione possibile.
- Tutti gli indirizzi all'interno del Dipartimento sono presi dalla rete 197.104.0.0/23.
- La rete di Sala Macchine (SM) contiene 25 PC, incluso il server web del Dipartimento (come raffigurato)
- La rete dell'Amministrazione (AM) contiene 14 PC
- La rete dello NPTLab (NL) contiene 11 PC
- La rete del SILab (SL) è suddivisa in due sottoreti:
 - SLS contiene 100 PC con indirizzi statici.
 - SLD permette di assegnare dinamicamente 100 indirizzi ai laptop degli studenti, via protocollo DHCP con server localizzato su uno dei 100 PC con indirizzo statico in SLS.
- All'interno di ognuna delle reti di Sala Macchine, Amministrazione, NPTLab e Internet utilizzare un unico switch, opportunamente configurato, e non inserire ulteriori router oltre a quelli mostrati in Figura.
- All'interno della rete del SILab inserire un unico switch, utilizzato sia per la VLAN dei PC con indirizzo statico sia per la VLAN dei PC con indirizzo dinamico.

Utilizzare il protocollo RIP version 2 per l'instradamento nell'intera rete in Figura. Si faccia in modo che il protocollo RIP non inondi di messaggi di controllo le sottoreti non contenenti altri router.

Per ognuna delle sottoreti definite si inseriscano due endpoint (PC o server) a scopo di test di connettività. Si garantisca la raggiungibilità di entrambi i web server da ogni punto della rete.

Per tale infrastruttura di rete lo studente è richiesto:

TASK 1 (20 minuti a disposizione)

- Calcolare i parametri di ogni sotto-rete 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-F22.xlsx* fornito dalla docente all'inizio del Task, con i parametri di rete.
- Configurare almeno due end system 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 descritti nelle specifiche.*
- Configurare i router in modo che usino il protocollo di instradamento RIPv2. Si faccia in modo che il protocollo RIP non inondi di messaggi le sotto-reti non contenenti altri router.
- Configurare i web server così che siano accessibili 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 composto da un server connectionless e un numero arbitrario di client, che implementi il gioco della morra cinese aderendo alle seguenti specifiche:

SERVER:

- È implementato in linguaggio C, come server singolo processo, e comunica con i client sfruttando i servizi del protocollo UDP.
- Mantiene memoria del numero di clienti serviti (*cs*) dal momento della sua partenza, e del numero di clienti contro cui ha vinto (*cw*).
- Scambia con i client tutti messaggi composti da un solo carattere
- Quando riceve da un client un messaggio 'p' ("let's play"), lo interpreta come segnale di inizio partita con un nuovo client. Incrementa *cs* e risponde con messaggio 'k' ("Ok!"). Stampa a video la notifica di arrivo di un nuovo client con relativo indirizzo.
- Quando riceve da un client un messaggio con contenuto 'f', 's' oppure 'c', estrae casualmente uno dei tre valori forbice / sasso / carta, e risponde al client con il carattere associato al valore estratto. Stampa a video la propria mossa e quella del client.
- Quando riceve da un client un messaggio 'y' ("you win") oppure 'i' ("I win"), risponde con un messaggio 'b' ("bye bye"). Se riceve 'y' incrementa *cw*. In ogni caso stampa a video il numero di partite vinte sul numero di partite giocate.

CLIENT:

- È implementato in linguaggio C, come server singolo processo, e comunica con il server sfruttando i servizi del protocollo UDP.
- Nel suo ciclo di vita gioca una singola partita con il server, tenendo traccia del numero di mosse in cui risulta vincitore (*score*)
- Alla partenza invia un messaggio 'p' al server e ne attende in risposta un messaggio 'k'.
- Entra quindi in un ciclo in cui legge un carattere da tastiera e opera come segue:
 - Se legge 'f', 's', oppure 'c': manda il carattere al server, ne attende in risposta la mossa e stampa a video chi ha vinto; '0' se pari. In caso di propria vittoria, incrementa *score* di 1; in caso di

vittoria del server, decrementa *score* di 1; in caso di parità, *score* resta inalterato. In tabella si rammentano le mosse vincenti (X denota parità):

	f	s	c
f	X	s	f
s	s	X	c
c	f	c	X

- Se legge ‘.’: invia al server ‘y’ se $score \leq 0$, ‘i’ se $score > 0$. Ne attende in risposta ‘b’ e termina.

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.

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.