

## Istruzioni:

---

- L'elaborato può essere svolto in gruppi di massimo 3 persone (si raccomanda l'aggregazione).
- NON dovete annunciarmi preventivamente che elaborato volete fare: sceglietene uno e fatelo.
- E' necessario inviare via mail sia i file modificati (cpp, h) sia una relazione sul lavoro svolto (quali parti di codice si sono modificate, commenti, grafici...).
- L'elaborato verrà letto, compilato e "fatto girare" per verificare la correttezza della sintassi e delle operazioni svolte.
- Evitate di richiedere via mail consigli sulla compilazione del software o sulla soluzione: ogni risposta a questo riguardo si trova molto facilmente su Internet.
- In caso di parametri mancanti, fate voi un'assunzione e inserite un commento nel codice e nella relazione.
- Evitate relazioni di dimensioni troppo elevate: 2 Megabytes complessivi (codice+relazione) e' il massimo che accettero' nella mia casella di posta.

- progetto C1 (valore max. 1 punto)

---

- Si consideri un sistema a coda con un servente e coda infinita. Gli arrivi siano di Poisson con tasso  $\lambda$ , mentre i tempi di servizio siano una variabile casuale esponenziale negativa con valor medio  $1/\mu$ .

- All'arrivo di un pacchetto, se il numero di pacchetti presenti nel sistema è minore di  $G$ , il pacchetto entra sempre; se invece è maggiore o uguale a  $G$ , il pacchetto viene accettato con probabilità  $q$  e rifiutato con probabilità  $1-q$ .

- Il software accetta in input  $\lambda$ ,  $\mu$ ,  $G$  e  $q$ , e calcola:
- la frazione di pacchetti rifiutati sul totale degli arrivi (probabilità di rifiuto)
- il tempo medio di attraversamento dei soli pacchetti ammessi.

- progetto C2 (valore max. 2 punti)

---

- Si consideri un sistema a coda con un servente e coda infinita. Gli arrivi siano di Poisson con tasso  $\lambda$ , mentre i tempi di servizio siano esponenziali negativi con valor medio  $1/\mu$ .

- Quando il sistema si svuota, il servente entra immediatamente in modalità *sleep*. Al primo arrivo successivo, il servente non può riprendere subito il servizio, ma deve attraversare un tempo di setup, variabile casuale esponenziale con valor medio  $1/\sigma$ . Durante il setup gli arrivi continuano ad accumularsi in coda.

- Il software accetta in input  $\lambda$ ,  $\mu$  e  $\sigma$ , e calcola il tempo medio di attraversamento dei pacchetti, le frazioni di tempo in cui il servente si trova negli stati di *sleep* e di *setup*, e il numero medio di riattivazioni del servente per unità di tempo.

- progetto C3 (valore max. 3 punti)

---

- Si consideri un sistema composto da  $N$  server, ciascuno con una propria coda FIFO infinita. Gli arrivi al sistema siano di Poisson con tasso  $\lambda$ .
- Il pacchetto in arrivo viene instradato verso una delle  $N$  code secondo una delle seguenti politiche, scelta in input: instradamento casuale, round-robin, oppure assegnazione alla coda più corta. I tempi di servizio del server  $i$ -esimo siano esponenziali negativi con valore medio  $1/\mu_i$ .
- Il software accetta in input il numero  $N$  di server, il tasso di arrivo  $\lambda$ , i tassi di servizio  $\mu_i$  e una politica di assegnazione degli arrivi alle  $N$  code. Il software calcola il tempo medio di permanenza dei pacchetti nel sistema, la lunghezza media di ogni coda e una misura della non uniformità/sbilanciamento del carico tra i server (come indice di sbilanciamento si consideri la differenza tra la massima e la minima lunghezza media di coda).

- Progetto OPL-1 (max. 1.5 punti)

---

- Si scriva un modello in OPL in cui, per ciascuna delle  $K$  commodity, si determinino due cammini tra sorgente  $s_k$  e destinazione  $t_k$ : un cammino primario e un cammino di backup. I due cammini relativi alla stessa commodity devono essere link-disjoint.
- Per realizzare il file .dat si utilizzi la topologia Nobel-US di SNDlib. Sulla topologia si individuino  $K = 10$  commodity con relativa domanda  $d_k$ . I costi e le capacità dei link siano scelti dallo studente in modo ragionevole.
- Si minimizzi il costo complessivo di instradamento rispettando i vincoli di capacità. Si comparino i risultati con il caso non protetto, in cui per ogni commodity si sceglie un solo cammino.

*Riferimento SNDlib: [home page](#) - Topologia consigliata: [Nobel-US](#) (14 nodi, 21 link, 91 domande)*

- Progetto OPL-2 (max. 3 punti)

---

- Si consideri una rete di telecomunicazione rappresentata dalla topologia Nobel-US di SNDlib. Ciascun nodo genera una domanda di traffico  $d_i$  diretta a un medesimo servizio applicativo, che può essere erogato da un insieme di edge server da attivare nella rete.
- Si scelga un sottoinsieme di nodi da attivare come edge server, ciascuno caratterizzato da un costo fisso di installazione e da una capacità massima. Ogni nodo di domanda deve essere assegnato a un solo edge server attivo, e il traffico risultante deve essere instradato sulla rete rispettando le capacità dei link. Per il file .dat si usino 10 domande ragionevoli, derivate dalla topologia scelta.
- Il modello OPL deve minimizzare il costo totale dato dalla somma dei costi di attivazione dei server e dei costi di trasporto del traffico. Si ri-risolve il problema variando il numero massimo di server attivabili oppure il livello di domanda, e si confrontino i risultati con il caso non capacitivo o con un numero fisso di server.

Riferimento SNDlib: [home page](#) - Topologia consigliata: [Nobel-US](#) (14 nodi, 21 link, 91 domande)