Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Corso di laurea magistrale in Ingegneria Informatica

CORSO DI RETI DI TELECOMUNICAZIONE (PRINCIPI E LABORATORIO)

PROGETTO C10

Andrea Appiani
Matricola: 1057683

Fabio Filippo Mandalari
Matricola: 1047426

A.A. 2021/2022

RELAZIONE

Il sistema di cui si vuole modellare il comportamento è un a|b|1|K|∞, in cui:

- a è una variabile casuale con distribuzione esponenziale;
- b è una variabile casuale con distribuzione di Rayleigh.

La distribuzione di Rayleigh non era presente di default, perciò è stato necessario implementarne una versione. Il codice si trova all'interno del file rand.hpp:

```
#define GEN_RAY(IS, AVERAGE, RES)                \
    {                                            \
    double pseudonum;                            \
    double sigma = AVERAGE/sqrt(M_PI/2);         \
    PSEUDO(IS,pseudonum);                        \
    RES = sqrt(-2*log(pseudonum))*sigma;         \
    }
```

L'obiettivo del progetto è quello di fornire in output due probabilità:

- Probabilità di rifiuto dovuta al meccanismo di scarto;
- Probabilità di rifiuto dovuta al trabocco della coda.

Per far fronte a questo tipo di calcolo sono stati introdotti quattro diversi contatori all'interno del file buffer.hpp:

- `Tot_arrivals` ⇒ contatore di quanti pacchetti richiedono di entrare in coda;
- `Tot_in_queue` ⇒ contatore di quanti pacchetti tra quelli che fanno richiesta riescono ad entrare in coda;
- `Tot_scarti_casuali` ⇒ contatore di quante volte viene scartato un pacchetto a caso tra quelli presenti in coda;
- `Tot_scarti_trabocco` ⇒ contatore di quanti pacchetti che fanno richiesta per entrare in coda vengono rifiutati perché la coda è già piena.

Tali contatori vengono resettati dopo ogni run.
Le due probabilità richieste vengono definite nel file queue.hpp e implementate nel file queue.cpp. L'implementazione prevede un calcolo così impostato:

```
prob_scarti_casuali = (buf->tot_scarti_casuali / buf->tot_in_queue) * 100;
prob_scarti_trabocco = (buf->tot_scarti_trabocco / buf->tot_arrivals) * 100;
```

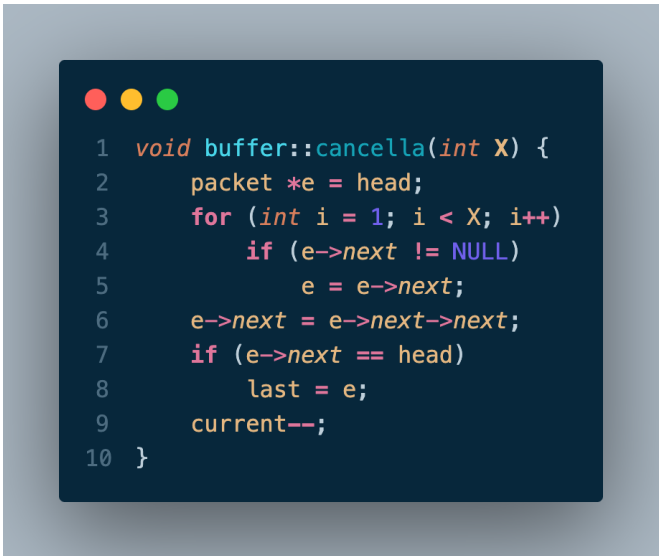
Nel file buffer.hpp viene definito anche un altro contatore: `current`. La sua utilità è da ricercare in una specifica imposta dalla traccia: se la coda ha più di G pacchetti, con G < K, allora deve essere scartato un pacchetto a caso con probabilità p.
Operativamente, quindi, il contatore `current` serve per contare quanti pacchetti ci sono effettivamente in coda in ogni momento.
Per rispettare la specifica è stato necessario definire, oltre a `current`, anche due variabili per la modellazione delle quantità G e K e un'altra variabile P che regoli la probabilità di scarto casuale dovuta al superamento della soglia G. Le tre variabili di cui si sta parlando sono state definite all'interno del file buffer.hpp.
I valori di `P`, `K` e `G` possono essere modificati modificando il file queue.cpp:

```
buf->K = read_int((char*) "\nBuffer Capacity K", 5, 1, 10);
buf->G = read_int((char*) "\nBuffer Threshold G", 3, 1, buf->K - 1);
buf->P = read_double((char*) "\nProbabilita' di scarto casuale P", 0.3, 0.1, 1);
```

Per scelta progettuale è stato deciso che ci deve essere uno scarto casuale se, estraendo mediante funzione `rand()` un numero compreso tra 1 e 10 nel momento in cui si realizza la condizione di "valore G superato", questo sia minore o uguale del valore di P moltiplicato per 10.

```
if (buf->current > buf->G) { // Controllo se la coda ha superato la soglia G
        int p = rand() % 10 + 1; // Estraggo un numero compreso tra 1 a 10
        if (p <= (buf->P * 10)) { // Il valore di P viene stabilito in queue.cpp
            buf->tot_scarti_casuali++;
            int X = rand() % (buf->current); // Estraggo "l'indice" del pacchetto da scartare dalla coda
            if (X == 0) // Se X == 0 estraggo il pacchetto in testa...
                buf->get();
            else // ... altrimenti ne scarto uno di quelli in coda
                buf->cancella(X);
        }
    }
```

Il meccanismo di scarto avviene estraendo innanzitutto un intero tra 0 e il numero attuale di pacchetti in coda -1, affidandogli il compito di "indice" che segna quale pacchetto scartare dalla coda. Se il numero estratto è X=0 viene scartata la testa tramite la funzione `get()` presente di default, mentre è stata definita una funzione ad-hoc `cancella()` (nel file buffer.cpp) nel caso in cui X sia diverso da zero:

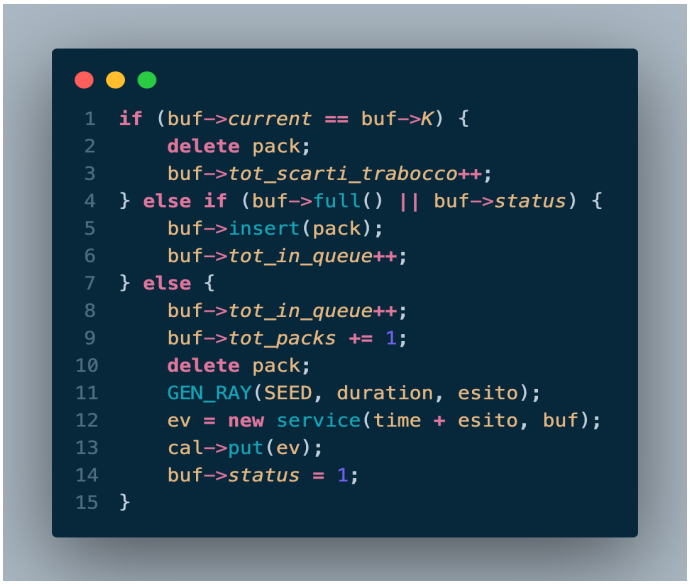


```
1  void buffer::cancella(int X) {
2      packet *e = head;
3      for (int i = 1; i < X; i++)
4          if (e->next != NULL)
5              e = e->next;
6      e->next = e->next->next;
7      if (e->next == head)
8          last = e;
9      current--;
10 }
```

Condizione necessaria e sufficiente perché ci sia uno scarto è che ci sia un arrivo. Di default il metodo `body()` presente nel file event.cpp gestisce i due casi in cui:

- C'è qualche pacchetto in coda, ma la coda non è satura;
- La coda è vuota.

Per l'implementazione del presente progetto è stato necessario aggiungere un ramo al costrutto `if` già presente. Tale ramo serve per gestire il caso in cui nella coda si sia raggiunta la capacità `K`. Sono state apportate altre modifiche ai due rami dell'`if` già presenti in modo tale che vengano incrementati i contatori di cui si è parlato a pag. 2.



```
1  if (buf->current == buf->K) {
2      delete pack;
3      buf->tot_scarti_trabocco++;
4  } else if (buf->full() || buf->status) {
5      buf->insert(pack);
6      buf->tot_in_queue++;
7  } else {
8      buf->tot_in_queue++;
9      buf->tot_packs += 1;
10     delete pack;
11     GEN_RAY(SEED, duration, esito);
12     ev = new service(time + esito, buf);
13     cal->put(ev);
14     buf->status = 1;
15 }
```

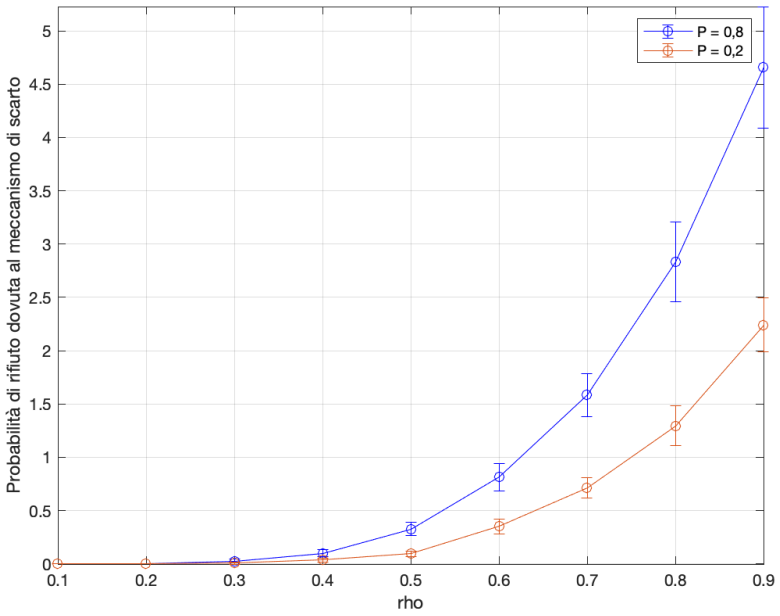
SIMULAZIONE

Le simulazioni sono state effettuate impostando i seguenti parametri:

- Numero di run: 50;
- K = 6;
- G = 4;
- $\rho \in [0.1, 0.9]$.

Fissati i parametri di cui sopra si è deciso di considerare come casi rappresentativi per il parametro P i valori P = 0.2 e P = 0.8. Gli intervalli di confidenza sono al 95%.

PROBABILITÀ DI RIFIUTO DOVUTA AL MECCANISMO DI SCARTO



PROBABILITÀ DI RIFIUTO DOVUTA AL TRABOCCO DELLA CODA