Università di Verona
Dipartimento di Informatica

Architetture Multimediali

Analisi della Qualità del Servizio di flussi simulati mediante NS-2

Davide Quaglia
Giuseppe Di Guglielmo

(parte del SW e di questo materiale è tratto dagli elaborati di Anna Pesarin/Monica Bertagnoli e di Michele Rizzato)

Scopo di questa esercitazione è la simulazione della trasmissione di un flusso multimediale e la valutazione dei parametri di Qualità del Servizio in diversi scenari di rete.

1. Generazione di un file di traccia di NS

E' possibile impostare la scrittura su un file di una traccia completa dei pacchetti che circolano nella rete; si usano i comandi:

```
set traceFile [open out.tr w]
$ns trace-all $traceFile
```

inoltre occorre modificare la procedura "finish" in modo che chiuda anche tale file:

```
proc finish {} {
      global ns nf traceFile
      $ns flush-trace
      close $nf
      close $traceFile
      exec nam out.nam &
      exit 0
}
```

Ciascuna riga del file prodotto segnala un evento avvenuto ad un certo pacchetto ed è organizzata in 12 campi (Figura 1).

Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id

Figura 1. Campi del file di traccia.

I campi hanno il seguente significato:
1) Tipo di evento sul pacchetto: assume 4 possibili valori r, +, -, d corrispondenti a received (all'uscita del link), enqueued (all'ingresso del link), dequeued (all'ingresso del link), dropped (all'ingresso del link).
2) Istante di tempo

3) Primo nodo del link
4) Secondo nodo del link
5) Tipo di pacchetto
6) Dimensione del pacchetto
7) Flags
8) Flow Id
9) Indirizzo sorgente nella forma nodo.porta
10) Indirizzo destinazione nella forma nodo.porta
11) Numero di sequenza all'interno di un flusso
12) ID univoco del pacchetto in NS

Se si vuole monitorare solo un link invece che tutta la rete (per ridurre la dimensione della traccia) si può sostituire

```
$ns trace-all $traceFile
```

con

```
$ns trace-queue $n0 $1 $traceFile
```

La seguente pagina WEB riporta una guida di riferimento per i possibili formati delle tracce generabili in NS
http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats#Normal_trace_formats

Il campo 8 della traccia è il ***Flow Id***, un identificativo di flusso: a ciascun agente della rete può essere assegnato un determinato ID in modo da poter monitorare tutti i pacchetti del traffico da esso generato.

Per assegnare uno specifico Flow Id ad un agente:

```
set $udp0 [new Agent/UDP]
$udp0 set fid_ 1000
```

2. Analisi di un file di traccia di NS mediante script Perl

L'utilizzo di un Flow Id specifico permette di effettuare delle misure sul traffico della rete mediante alcuni dei seguenti script (scaricabili dalla pagina del corso).

`pck-rate-at-node.pl <tracefile> <node> <granularity>`
misura la frequenza di arrivo dei pacchetti al nodo `<node>` in intervalli di tempo sucessivi di ampiezza `<granularity>`

`throughput-at-node.pl <tracefile> <node> <granularity>`
misura il bit-rate arrivato al nodo `<node>` in intervalli di tempo successivi di ampiezza `<granularity>`

`throughput-flow-F-at-node.pl <tracefile> <flow> <node> <granularity>`
misura il bit-rate appartenente al flusso `<flow>` arrivato al nodo `<node>` in intervalli di tempo successivi di ampiezza `<granularity>`

`dpck-at-link.pl <trace> <node0> <node1> <granularity>`
misura il numero di pacchetti persi alla coda del link tra `<node0>` e `<node1>` in intervalli di tempo successivi di ampiezza `<granularity>`

`dpck-flow-F-at-link.pl <trace> <flow> <node0> <node1> <granularity>`
misura il numero di pacchetti appartenenti al flusso `<flow>` e persi alla coda del link tra `<node0>` e `<node1>` in intervalli di tempo successivi di ampiezza `<granularity>`

`flow-F-at-N.pl <tracefile> <flowID> <node>`
estrae la traccia con i pacchetti appartenenti ad un certo flusso `<flowID>` arrivati ad un certo nodo `<node>`

Se si salva l'output di `flow-F-at-N.pl` e poi lo si passa a `pck-rate-at-node.pl` o a `throughput-at-node.pl` si ottiene il pck-rate e throughput per lo specifico flusso.

`jitter-flow-F-at-N.pl <tracefile> <flowID> <node> <period>`
misura la deviazione standard dei tempi di interarrivo `<period>` dei pacchetti del flusso `<flowID>` al nodo `<node>`

`jitter-flow-F-at-N-granularity.pl <tracefile> <flowID> <node> <period> <granularity>`
misura la deviazione standard dei tempi di interarrivo `<period>` dei pacchetti del flusso `<flowID>` al nodo `<node>` in intervalli di tempo successivi di ampiezza `<granularity>`

Per eseguire ciascuno script, dopo aver generato un tracefile, eseguirli come nell'esempio:

```
./flow-F-at-N.pl out.tr 1000 3
```

è possibile visualizzare più comodamente l'output di tali script redirigendolo su file oppure tramite pipe sul comando `less`

```
./flow-F-at-N.pl out.tr 1000 3 > flow-1000.tr
./flow-F-at-N.pl out.tr 1000 3 | less
```

3. Analisi di un file di traccia di NS mediante il tool NStat

Scopo di questo tool è analizzare ed estrarre statistiche dall'output di NS-2. Le statistiche generate si riferiscono alla perdita di pacchetti (Packet Loss Rate, PLR), perdita di byte (Byte Loss Rate, BLR) e alla distribuzione del ritardo end-to-end riferiti ad un determinato flusso di dati indentificato da un FlowID.

3.1 Istruzioni per l'uso

Il tool richiede la presenza di Java. Occorre scompattare il file `nstat.zip` nella propria HOME

```
cd $HOME
unzip nstat.zip
```

Il tool deve essere lanciato nel modo seguente:

```
java -classpath $HOME/nstat statisticNS $PWD
```

All'avvio del programma vengono chiesti:

- il nome (con eventuale percorso) del file di traccia creato da NS-2 da analizzare
- l'ID del flusso da analizzare
- la grandezza (in secondi) della finestra su cui vengono calcolati il PLR e BLR
- la dimensione (in secondi) del quantile utilizzato per creare il grafico della distribuzione del ritardo end-to-end
- il prefisso utilizzato per generare i vari nomi dei file contenenti le statistiche

Il programma creerà i seguenti file (si assume di aver scritto "prefix" come prefisso dei nomi):

- gnuScript.plt - script di GNUplot per la generazione dei grafici
- prefix_b.dat - contenente il BLR nei vari intervalli dati dalla finestra di analisi
- prefix_p.dat - contenente il PLR nei vari intervalli dati dalla finestra di analisi
- prefix_d.dat - distribuzione del ritardo end-to-end in funzione del quantile

Il file gnuScript.plt può essere elaborato con GNUplot nel modo seguente:

```
gnuplot gnuScript.plt
```

verranno creati i grafici ByteLossRate.eps, PackLossRate.eps e RitEndtoEnd.eps che potranno essere visualizzati con comuni tool grafici come Gimp o kview.

4. Creazione di un modello di rete di test

Al fine di provare il comportamento di un flusso multimediale in vari contesti di trasmissione occorre costruire un modello di rete sintetico ma in grado di catturare gli aspetti essenziali di uno scenario reale. Introduciamo quindi la *topologia a bottleneck* raffigurata in Figura 2 e descritta nel file `bottleneck.tcl`.

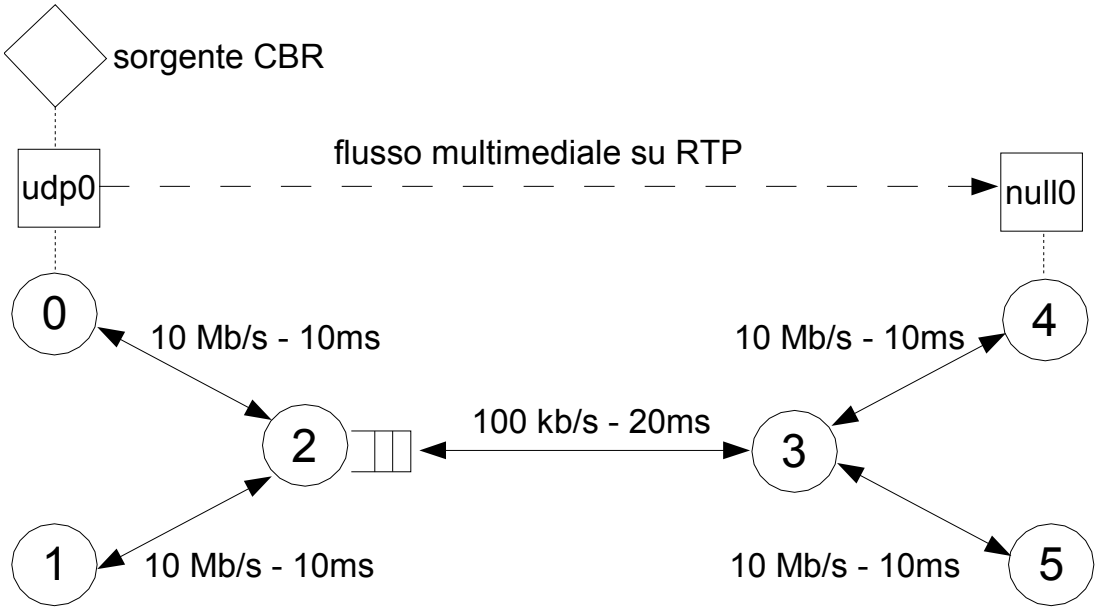

Figura 2. Modello di rete con topologia a bottleneck.

In tale modello è possibile notare link periferici con alta capacità e basso ritardo che rappresentano la rete vicina agli utenti finali (nodi0/1/4/5) e un link a bassa capacità e

maggior ritardo che rappresenta cosa gli utenti “vedono” del collegamento geografico che li collega e che rappresenta il “collo di bottiglia” della rete. La presenza di 4 nodi utenti permette di creare due coppie TX-RX: una coppia nodo0-nodo4 sarà interessata dal traffico multimediale che vogliamo studiare mentre la coppia nodo1-nodo5 sarà interessata da traffico cosiddetto “interferente” perché andrà ad interferire col traffico multimediale contendendo la capacità del bottleneck.

Tra i nodi 0 e 4 viene installata una connessione UDP e sopra la sorgente UDP viene installata una sorgente a bitrate costante che genera pacchetti di 400 byte ogni 0.040 sec (cioè 80 kb/s); tale flusso rappresenta il modello semplificato di un flusso multimediale su RTP (ad es. un flusso video compresso – 25 frame/s – 1 frame/pacchetto). In ingresso al link nodo2-nodo3 è visualizzata la coda che andrà a riempirsi quando il link geografico non riuscirà a smaltire i pacchetti in ingresso; tale coda conterrà al massimo 15 pacchetti e i pacchetti in entrata nella coda piena vengono persi.

Le statistiche su tale coda vengono rilevate ogni 0.1 sec. e salvate nel file `bottleneck.queue` con il seguente formato:

- istante di tempo di riferimento
- nodi estremi del link di cui la coda fa parte
- dimensione della coda in byte
- dimensione della coda in numero di pacchetti
- numero tot di pacchetti entrati nella coda
- numero tot di pacchetti usciti dalla coda
- numero tot di pacchetti persi per coda piena
- numero tot di byte entrati nella coda
- numero tot di byte usciti dalla coda
- numero tot di byte persi per coda piena.

Analizzando il file di traccia è possibile verificare che:

- i pacchetti partono dal nodo 0 distanziati di 0.040 sec
- nessun pacchetto viene perso (non ci sono eventi di tipo “d”) in quanto la capacità del bottleneck uguaglia il bitrate della sorgente

Analizzando il jitter del flusso multimediale al ricevitore col comando:

```
./jitter-flow-F-at-N-granularity.pl  bottleneck.tr 1000 4 0.040 1
```

si nota una variazione molto bassa (praticamente trascurabile) del tempo di interarrivo dei pacchetti multimediali.

Analizzando la dimensione (in pacchetti) della coda al nodo 2 (colonna 5) col comando:

```
less bottleneck.queue
```

si nota che la coda è vuota in quanto i pacchetti in arrivo trovano subito il bottleneck libero.

5. Generazione di una congestione

Si vuole analizzare il comportamento del flusso multimediale in presenza di traffico concorrente che contende l'uso del bottleneck. Per questo motivo all'istante 5 viene avviata una sessione TCP (ad es. un nuovo utente accede al web e i suoi pacchetti passano nella stessa porzione di rete del traffico multimediale). Il file `congestione.tcl` contiene tale scenario.

Analizzando il file di traccia si nota che entrambi i flussi perdono pacchetti in corrispondenza della coda in ingresso al bottleneck nei secondi successivi all'attivazione del flusso TCP.

Si può analizzare l'andamento della perdita dei pacchetti in corrispondenza del bottleneck per il flusso multimediale con il comando:

```
./dpck-flow-F-at-link.pl congestione.tr 1000 2 3 1
```

si può notare che:

- la perdita non avviene immediatamente all'attivazione del TCP ma dopo un certo tempo (5.8 sec)
- la perdita è inizialmente accentuata ma poi va decrescendo e quasi sparisce.

Visualizzando la statistica della coda in corrispondenza del bottleneck con il comando:

```
less congestione.tr
```

si può notare che:

- a partire dall'istante 5.0 sec. la lunghezza della coda comincia a crescere fino alla lunghezza di 13 pacchetti
- da tale momento la coda non cresce più ma vengono persi pacchetti
- successivamente la lunghezza della coda rimane sui 10-12 pacchetti e vengono persi sporadicamente dei pacchetti.

Si può analizzare il throughput del flusso multimediale a monte e a valle del bottleneck coi comandi:

```
./throughput-flow-F-at-link.pl congestione.tr 1000 2 1
./throughput-flow-F-at-link.pl congestione.tr 1000 4 1
```

si può vedere che:

- il throughput “offerto” alla rete dal trasmettitore è sempre costante a 80 kb/s
- fino all'avvio del TCP il bottleneck è sfruttato solo al 80%
- il throughput che effettivamente arriva al ricevitore varia nel tempo
- all'avvio della trasmissione TCP scende a causa della perdita di pacchetti sul bottleneck ma poi torna al suo valore di 80 kb/s.

Si può analizzare il throughput del flusso TCP a monte e a valle del bottleneck coi comandi:

```
./throughput-flow-F-at-link.pl congestione.tr 2000 2 1
```

```
./throughput-flow-F-at-link.pl congestione.tr 2000 5 1
```

si può vedere che:

- il throughput “offerto” alla rete varia nel tempo; esso scende gradualmente mentre è attivo il flusso multimediale;
- si alza dopo la fine della trasmissione multimediale fino ad occupare il 100% del bottleneck

Si può analizzare il jitter del flusso multimediale al ricevitore col comando:

```
./jitter-flow-F-at-N-granularity.pl  congestione.tr 1000 4 0.040 1
```

Si nota che la variazione del ritardo:

- è trascurabile fino all'attivazione del TCP
- è considerevole in prossimità dell'attivazione del TCP
- rimane su valori significativi, anche se minori, per tutto il resto della trasmissione multimediale contemporanea alla trasmissione TCP

6. Esercizi

Si valuti la distribuzione dei ritardi della trasmissione multimediale nello scenario congestionato utilizzando il tool NStat. Che dimensione dovrebbe avere il buffer di playout al ricevitore nell'ipotesi che esso venga riempito a metà prima dell'inizio della riproduzione del brano ?

Si ripetano le misure effettuate sullo scenario congestionato sostituendo il traffico TCP con un flusso CBR analogo a quello già presente. Che tipo di considerazioni si possono fare ?