

NS - Network Simulator



A cura di:

Binotto Riccardo

Simonetti Angelo

Supervisore:

Prof. Carlo Combi

Simulazione di Reti di
Calcolatori



Cos'è NS?

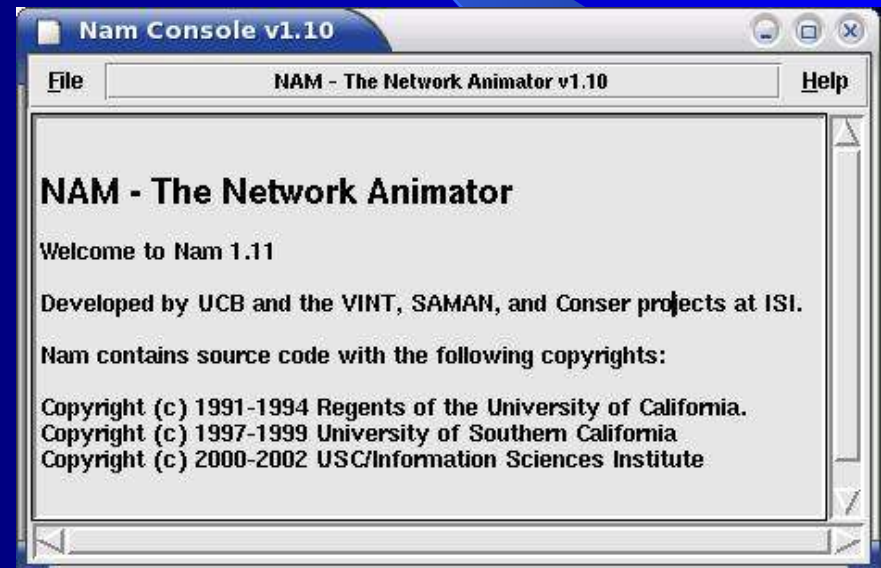
- * Un simulatore di reti ad eventi discreti
- * Viene usato per simulazioni di TCP, algoritmi di routing, reti wired (collegate fisicamente) e wireless (prive di collegamento fisico), ecc.
- * Attualmente il suo sviluppo viene portato avanti dal DARPA

Chi usa NS?

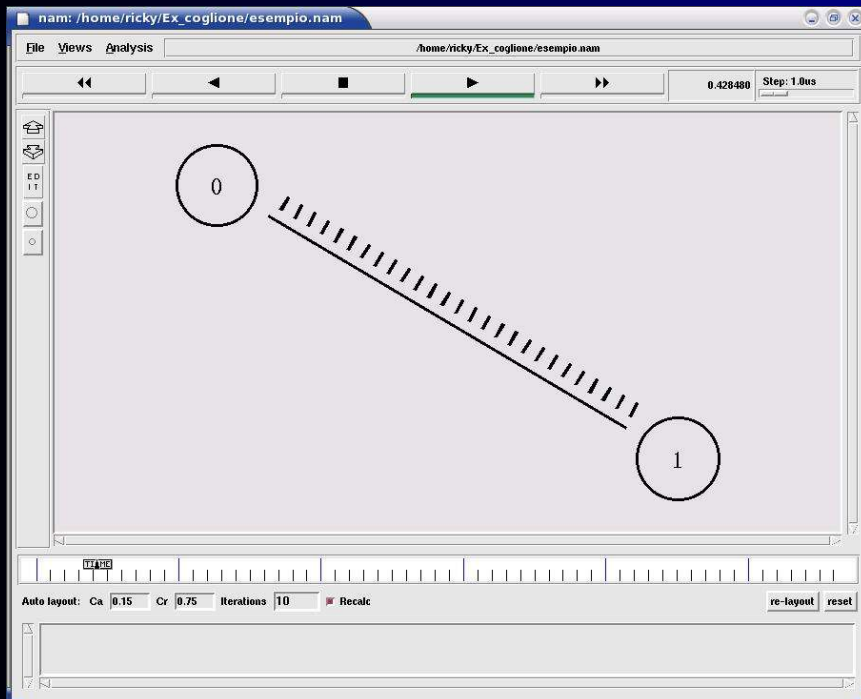
- * Nell'ambito della ricerca, viene utilizzato per testare il funzionamento di nuovi protocolli o modifiche di quelli pre-esistenti
- * Nell'ambito dell'educazione, viene utilizzato per illustrare le modalità di funzionamento degli attuali protocolli di rete
- * Nell'ambito commerciale, per validare i propri risultati con uno strumento comune.

Come è realizzato NS?

- * NS è realizzato completamente in C++, ed il sorgente viene distribuito liberamente.
- * E' corredato da diversi tools che ne semplificano l'utilizzo, il più importante è NAM
- * NAM è un Network Animator che legge un trace file prodotto da ns e apre una finestra nella quale dà vita ad un'animazione della simulazione



NAM - Network Animator -



- * Mostra la topologia della rete, i flussi di pacchetti, i pacchetti in coda nei buffer e quelli dropped, i link che si guastano o che si aggiustano, e così via; attraverso una comoda interfaccia utente è possibile controllare diversi aspetti della simulazione.

Dove si trova NS?

- * La home page ufficiale di NS:
 - <http://www.isi.edu/nsnam/ns>
- * La home page ufficiale di NAM:
 - <http://www.isi.edu/nsnam/nam>
- * Esiste anche un pacchetto compresso contenente tutti il necessario (librerie, ecc.) per usare NS.
- * E' possibile prelevare i sorgenti da compilare e tutta la documentazione relativa ad NS: NS-Manual (Manuale di riferimento con le principali funzioni), NS-Tutorial (Guida per iniziare ad utilizzare NS).

Struttura di NS

- * **Scheduler degli eventi:**

E' l'elemento chiave di ogni simulatore che gestisce l'interazione tra gli eventi della simulazione e l'andamento temporale della simulazione stessa. Lo scheduler degli eventi di NS è a tempo discreto.

- * **I nodi:**

Sono un elemento topologico della rete, rappresentano tutti i dispositivi di rete (router, switch, ecc.) . Ad ogni nodo è possibile associare un qualsiasi generatore di traffico (Agent) e applicazione (Application).

Struttura di NS (2)

- * **I link:**

Sono un elemento topologico della rete; ogni link è caratterizzato da un tempo di propagazione, da una capacità di canale e da altri parametri. Ad ogni link è possibile associare una probabilità di errore oppure simulare dei guasti.

- * **I pacchetti:**

Vengono trasmessi tramite connessioni tra due nodi diversi e relativi link, secondo le regole imposte dalla topologia della rete. Ogni pacchetto ha un identificatore univoco ed è formato dagli header dei diversi livelli implementati nella topologia di rete; non esiste un campo dati e la lunghezza di un pacchetto viene simulato da un campo Length.

Cosa si può simulare con NS?

- * Si possono simulare connessioni **tcp** e **udp** a livello **trasporto**. LAN con diversi tipi di ascolto del canale (vi è la possibilità di usare messaggi di tipo **multicast** e **broadcast**).
- * Si possono simulare **router di confine** (per LAN ad esempio), si possono usare diversi tipi di **mezzi comunicativi wired o wireless**, dei quali si può settare la **banda**, il **tempo di propagazione della linea** ed altro ancora.
- * Si possono simulare diversi **algoritmi di routing (DV, LinkState)**
- * Si possono **monitorare** i comportamenti delle diverse code implementabili su un router.

Cosa si può simulare con NS?

- * A livello **Applicazione** si può simulare il protocollo
 - ftp,
 - telnet,
 - http
 - si possono settare i diversi andamenti del traffico (ad esempio un aumento del traffico di tipo esponenziale)
 - si possono introdurre rotture dei link temporizzate.

Script di NS

- * Le simulazioni di NS vengono configurate tramite riga di comando oppure mediante comodi script TCL. Ogni script è formato da:
 3. Una parte che descrive la topologia della rete
 4. Una parte che descrive gli eventi della simulazione.

```
File Modifica Visualizza Terminale Vai Aiuto
#Author Binotto Riccardo VR001209
#Author Simonetti Angelo VR001207
set ns [new Simulator]
set nf [open file.nam w]
$ns namtrace-all $nf
set tr1 [open file.tr w]
$ns trace-all $tr1

proc finish {} {
    global ns nf tr1
    $ns flush-trace
    close $nf
    close $tr1
    exec nam file.nam &
    exit 0 }

#Creazione della topologia
for {set i 0} {$i < 2} {incr i} {
    set n($i) [$ns node]
}

$ns duplex-link $n(0) $n(1) 2Mb 100ms DropTail
set udp00 [$ns create-connection UDP $n(0) Null $n(1) 0]
set cbr00 [new Application/Traffic/CBR]
$cbr00 attach-agent $udp00

#Temporizzazione della simulazione
$ns at 0.1 "$cbr00 start"
$ns rtmodel-at 2.15 down $n(0) $n(1)
$ns rtmodel-at 2.16 up $n(0) $n(1)
$ns at 5.5 "$cbr00 stop"
$ns at 6. "finish"
$ns run
```

Script di NS (2)

- * NS crea dei file di trace con la descrizione dei pacchetti generati e trasmessi durante la simulazione.
- * E' possibile tracciare tutti i pacchetti o solo alcune grandezze di interesse.
- * Esistono due tipi di trace file
 - NS trace
 - NAM trace
- * NAM trace serve per la simulazione grafica.

```
File Modifica Visualizza Terminale Vai Aiuto
EVENT | TIME | FROM_NODE | TO_NODE | PK_TYPE | PK_SIZE | FLAGS | FID | SRC_ADDR | DST_ADDR | SEQ_NUM | PK_ID
+ 0.1 0 1 cbr 210 ----- 0 0.0 1.0 0 0
- 0.1 0 1 cbr 210 ----- 0 0.0 1.0 0 0
+ 0.10375 0 1 cbr 210 ----- 0 0.0 1.0 1 1
- 0.10375 0 1 cbr 210 ----- 0 0.0 1.0 1 1
+ 0.1075 0 1 cbr 210 ----- 0 0.0 1.0 2 2
- 0.1075 0 1 cbr 210 ----- 0 0.0 1.0 2 2
+ 0.11125 0 1 cbr 210 ----- 0 0.0 1.0 3 3
- 0.11125 0 1 cbr 210 ----- 0 0.0 1.0 3 3
+ 0.115 0 1 cbr 210 ----- 0 0.0 1.0 4 4
- 0.115 0 1 cbr 210 ----- 0 0.0 1.0 4 4
+ 0.11875 0 1 cbr 210 ----- 0 0.0 1.0 5 5
- 0.11875 0 1 cbr 210 ----- 0 0.0 1.0 5 5
+ 0.1225 0 1 cbr 210 ----- 0 0.0 1.0 6 6
- 0.1225 0 1 cbr 210 ----- 0 0.0 1.0 6 6
+ 0.12625 0 1 cbr 210 ----- 0 0.0 1.0 7 7
- 0.12625 0 1 cbr 210 ----- 0 0.0 1.0 7 7
+ 0.13 0 1 cbr 210 ----- 0 0.0 1.0 8 8
- 0.13 0 1 cbr 210 ----- 0 0.0 1.0 8 8
+ 0.13375 0 1 cbr 210 ----- 0 0.0 1.0 9 9
- 0.13375 0 1 cbr 210 ----- 0 0.0 1.0 9 9
+ 0.1375 0 1 cbr 210 ----- 0 0.0 1.0 10 10
- 0.1375 0 1 cbr 210 ----- 0 0.0 1.0 10 10
+ 0.14125 0 1 cbr 210 ----- 0 0.0 1.0 11 11
- 0.14125 0 1 cbr 210 ----- 0 0.0 1.0 11 11
+ 0.145 0 1 cbr 210 ----- 0 0.0 1.0 12 12
- 0.145 0 1 cbr 210 ----- 0 0.0 1.0 12 12
+ 0.14875 0 1 cbr 210 ----- 0 0.0 1.0 13 13
- 0.14875 0 1 cbr 210 ----- 0 0.0 1.0 13 13
+ 0.1525 0 1 cbr 210 ----- 0 0.0 1.0 14 14
- 0.1525 0 1 cbr 210 ----- 0 0.0 1.0 14 14
+ 0.15625 0 1 cbr 210 ----- 0 0.0 1.0 15 15
- 0.15625 0 1 cbr 210 ----- 0 0.0 1.0 15 15
+ 0.16 0 1 cbr 210 ----- 0 0.0 1.0 16 16
- 0.16 0 1 cbr 210 ----- 0 0.0 1.0 16 16
+ 0.16375 0 1 cbr 210 ----- 0 0.0 1.0 17 17
- 0.16375 0 1 cbr 210 ----- 0 0.0 1.0 17 17
+ 0.1675 0 1 cbr 210 ----- 0 0.0 1.0 18 18
```

Esercitazioni

Script “reti_lab”

- * Per lanciare lo script eseguire dalla shell `./reti_lab nomefile`, dove ***nomefile*** è il nome dell'esercitazione.
- * Al termine dell'esecuzione dello script nella home directory comparirà una directory “Ex_***nomefile***” contenente i file relativi all'esercitazione.

Script “reti_lab” (2)

- * Avviato lo script vengono poste delle domande, a cui bisogna rispondere in modo corretto, come illustra lo script...

Script “reti_lab” (3)

- * Viene chiesto:
 - * Il numero di nodi della rete
 - * La capacità del canale (in Mb)
 - * Tempo di propagazione (in ms)
 - * La coppia di nodi su cui fare il collegamento.
 - * Il guasto di un canale
 - * ...

Esempio (1)

Inserire il numero di nodi:

4

Inserire la capacità del canale espresso in Mb :

-- ATTENZIONE: Inserire solo il valore numerico senza unità di misura!

2

Inserire il tempo di propagazione espresso in ms :

-- ATTENZIONE: Inserire solo il valore numerico senza unità di misura!

100

Inserire la COPPIA di nodi su cui fare il collegamento e PREMERE INVIO:

[--N.B. Scrivere 'ok' per fine collegamenti!!]

(Esempio: 0 2 , collega il nodo 0 con il nodo 2)

0 1

Inserire la COPPIA di nodi su cui fare il collegamento e PREMERE INVIO:

[--N.B. Scrivere 'ok' per fine collegamenti!!]

(Esempio: 0 2 , collega il nodo 0 con il nodo 2)

1 2

Inserire la COPPIA di nodi su cui fare il collegamento e PREMERE INVIO:

[--N.B. Scrivere 'ok' per fine collegamenti!!]

(Esempio: 0 2 , collega il nodo 0 con il nodo 2)

0 3

Esempio (2)

```
Inserire la COPPIA di nodi su cui fare il collegamento e PREMERE INVIO:  
[--N.B. Scrivere 'ok' per fine collegamenti!!]  
(Esempio: 0 2 , collega il nodo 0 con il nodo 2)
```

ok

```
Inserire il NODO mittente che trasmetterà i pacchetti:
```

0

```
Inserire il NODO destinatario che riceverà i pacchetti:
```

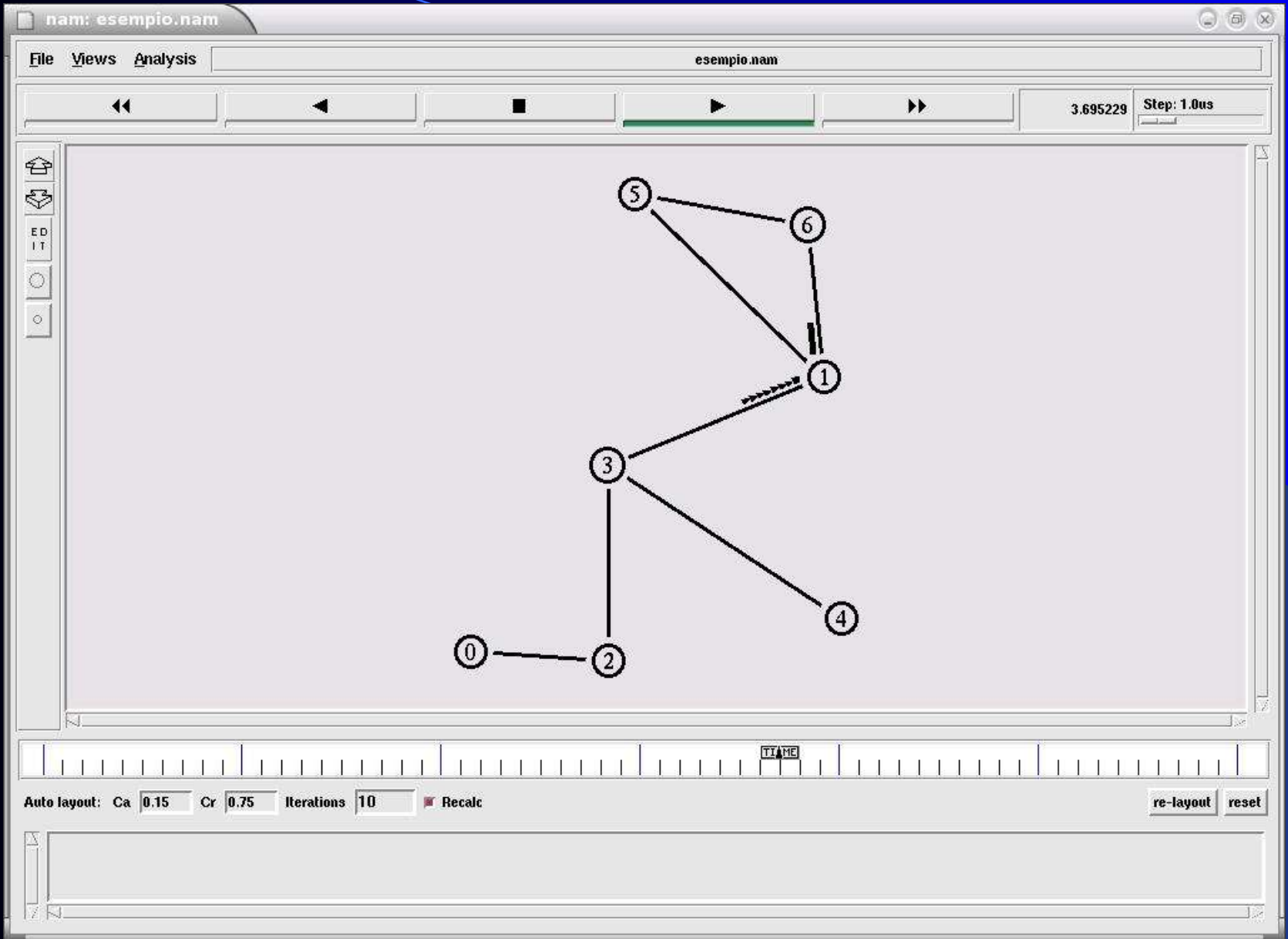
2

... e la simulazione prosegue...

Il nodo mittente che trasmette i pacchetti DEVE poter raggiungere il nodo destinatario che riceverà i pacchetti

Script “reti_lab” (4)

- * Dopo l’inserimento dei dati viene avviato in automatico NAM, che gestisce la simulazione grafica.
- * Il tasto **play** serve per avviare la simulazione grafica.
- * In alto a destra è presente l’unità di misura della velocità della simulazione.
- * In basso è presente la scala dei tempi assieme ad
- * una **barra che indica gli avvenimenti** (inizialmente vuota, se scegliete di interrompere il link, li verranno segnalati i tempi di rottura e con un doppio clic la simulazione viene portata nel tempo corrispondente dell'avvenimento selezionato).



Interfaccia grafica di NAM

- * Una funzione importante è quella implementata dal pulsante *re-layout*, che dispone i nodi nel modo migliore possibile per una vista grafica ottimale. Altrimenti è possibile effettuare la disposizione manualmente, premendo il pulsante a sinistra denominato *Edit*.

Trace File di NS

Tabella dei risultati

Event	Time	from_node	to_node	pk_type	pk_size	flags	fid	scr_addr	dst_addr	seq_num	pk_id

....

Spiegazione dei campi

* **event:**

- + indica un evento di accodamento alla coda
- - indica una partenza dalla coda
- d indica un pacchetto perso(coda piena)
- r indica un evento di ricezione (di un pacchetto ad un certo nodo)

* **time:**

- indica il tempo (simulato) dell'occorrenza dell'evento

* **from node e to node:**

- indica il nodo a cui ´e destinato il pacchetto.La coppia dei campi from node
- to node consente di indentificare il link coinvolto

* **pk type:**

- indica il tipo di pacchetto

* **pk size:**

- indica la dimensione del pacchetto (in byte)

* **flags:**

- contiene dei flags; NS implementa solo il bit Excplicit Congestion Notification
- (ECN), i rimanenti bit non sono utilizzati

Spiegazione dei campi (2)

- * **fid:**
 - indica il flow id(fid) di un IP. Questa informazione pu`o essere specificata di solito per NAM (Network AniMator) ed utilizzata per scopi d'analisi
- * **src addr e dst addr:**
 - indicano rispettivamente la porta sorgente e di destinazione (node.port)
- * **seq num:**
 - indica il sequence number del pacchetto. Anche se UDP non usa il seq num, ns tiene traccia dei pacchetti UDP per scopi d'analisi
- * **pk id:**
 - contiene un identificatore unico del pacchetto