

[open stringa mode] restituisce il descrittore di un file aperto (di nome *stringa*) in modalità “riscrittura” (se *mode* è w) o in “append” (se *mode* è a)

puts \$file_handler "stringa1 \$var1 stringa2 \$var2 [...]" scrive nel file indirizzato da *file_handler* (opzionale, il default è la scrittura a schermo) la sequenza data di stringhe e variabili

\$argc contiene il numero di parametri della linea di comando

\$argv è una lista di parametri della riga di comando

[lindex \$lista num] restituisce l'elemento di posto *num* della *lista* (il primo ha posto 0)

\$argv0 (diverso da **[lindex argv 0]**) restituisce il nome del file .tcl corrente

[\$ns get-ns-traceall] restituisce il descrittore del file di tracce di *ns* (diverso da quello di nam!), eventualmente creandolo se non esiste

[\$ns monitor-queue \$Nodo1 \$Nodo2 \$file_handler \$sample_interval] crea un oggetto (“QueueMonitor”) che tiene traccia di alcune variabili di stato relative al link che collega *\$Nodo1* a *\$Nodo2*, scrivendo sul file *\$file_handler* e con intervallo di tempo tra un campione e l'altro dato da *\$sample_interval* (opzionale, il default è 0.1s)

[\$QueueMonitor0 set pdepartures_] restituisce il numero di pacchetti inviati sul link su cui è definito *QueueMonitor0*

[\$QueueMonitor0 set parrivals_] restituisce il numero totale di pacchetti giunti sulla coda del link su cui è definito *QueueMonitor0* (sono compresi i pacchetti eventualmente scartati per overflow del buffer)

[\$QueueMonitor0 set pdrops_] restituisce il numero di pacchetti scartati per overflow del buffer del link su cui è definito *QueueMonitor0*

[\$QueueMonitor0 set pkts_] restituisce il numero di pacchetti che attualmente si trovano nel buffer del link su cui è definito *QueueMonitor0*

\$QueueMonitor0 set-delay-samples \$delay0 fa in modo che l'oggetto *delay0* (di tipo Samples) tenga traccia dei ritardi di accodamento del link su cui è definito *QueueMonitor0*

[\$delay0 mean] restituisce la media di *delay0* (di tipo Samples)

\$delay0 reset resetta l'oggetto *delay0* di tipo Samples

\$ns rtproto Static seleziona un protocollo di routing statico

\$ns rtproto Session seleziona un protocollo di routing dinamico ideale

\$ns rtproto DV seleziona un protocollo di tipo Distance Vector

\$ns rtproto LS seleziona un protocollo di tipo Link State

\$ns rtmodel-at 4.0 down \$N0 \$N1 all'istante t=4s fa cadere il link tra il nodo N0 ed il nodo N1 *insieme al link opposto*

\$ns rtmodel-at 6.0 up \$N0 \$N1 all'istante t=6s ristabilisce il link tra il nodo N0 ed il nodo N1 e il link opposto

\$ns cost \$N0 \$N1 2 assegna costo 2 al link (di andata) tra il nodo Node0 e il nodo Node1 e...

\$ns cost \$N1 \$N0 3 ...costo 3 al link opposto

[expr es] restituisce il risultato dell'espressione matematica *es*

(esempio d'uso: **set a [expr 1/[expr \$c+\$d]]**)

esempi di indirizzamento di array: **\$var(0)** **\$var(\$index)**

esempi di costrutti di base (if, for), simili a quelli del linguaggio C:

```
if {$argc == 2} {
    set buffer_len [lindex $argv 0]
    set load [lindex $argv 1]
} else {
    puts "Usage: ns $argv0 <traffic_load>"
    exit 1
}

for {set sources 0} {$sources<20} {incr sources} {
    set ExpOnOff0($sources) [new Application/Traffic/Exponential]
}
```

ESEMPI DI AGENTI E APPLICAZIONI UTILIZZATI

Agenti di tipo sorgente

UDP

```
set UDP0 [new Agent/UDP]                ;# crea un agente (sorgente) UDP
                                           ;#   di nome UDP0
$UDP0 set packetSize_ 50                 ;# UDP0 frammentera' pacchetti di
                                           ;#   dimensione maggiore di 50 byte
$UDP0 set fid_ 1                         ;# assegna l'identificativo 1
                                           ;#   al flusso dati uscente da UDP0
```

TCP

```
set TCP0 [new Agent/TCP/RFC793edu]      ;# crea un agente (sorgente) TCP
                                           ;#   "semplificato" di nome TCP0
$TCP0 set window_ 20                    ;# imposta la dimensione della fi-
                                           ;#   nestra di TCP0 a 20 pacchetti
$TCP0 set packetSize_ 50                ;# TCP0 frammentera' pacchetti di
                                           ;#   dimensione maggiore di 50 byte
$TCP0 set fid_ 1                         ;# assegna l'identificativo 1
                                           ;#   al flusso dati uscente da TCP0
```

Agenti di tipo ricevente

Null

```
set Null0 [new Agent/Null]              ;# crea un agente Null di nome
                                           ;#   Null0
```

LossMonitor

```
set LossMonitor0 [new Agent/LossMonitor] ;# crea un agente LossMonitor
                                           ;#   di nome LossMonitor0
...
$ns at 5.0 "set p_arrivati [$LossMonitor0 set npkts_]"
                                           ;# assegna alla variabile p_arrivati
                                           ;#   il numero di pacchetti arrivati
                                           ;#   all'agente LossMonitor0(di tipo
                                           ;#   LossMonitor)fino all'istante 5s
$ns at 5.0 "set p_persi [$LossMonitor0 set nlost_]"
                                           ;# assegna alla variabile p_persi il
                                           ;#   numero di pacchetti persi fino
                                           ;#   all'istante 5s tra quelli
                                           ;#   diretti all'agente LossMonitor0
                                           ;#   (di tipo LossMonitor)
```

TcpSink

```
set TCPSink0 [new Agent/TCPSink]        ;# crea un agente (ricevente) TCP
                                           ;#   "semplificato" di nome TCPSink0
```

Applicazioni

CBR

(constant bit rate)

```
set CBR0 [new Application/Traffic/CBR]   ;# crea una sorgente CBR
                                           ;#   di nome CBR0
$CBR0 set interval_ 0.1s                 ;# imposta a 0.1s l'intervallo di
                                           ;#   tempo tra la trasmissione di
                                           ;#   un pacchetto (della sorgente
                                           ;#   CBR0) e quella del successivo
$CBR0 set rate_ 100k                      ;# imposto a 100 kbps la velocita'
                                           ;#   media di trasmissione di CBR0
$CBR0 set packetSize_ 100                ;# imposta a 100 byte la dimensione
                                           ;#   del pacchetto di CBR0

;# I tre parametri interval_, rate_ e packetSize_ non vanno mai impostati insieme.
;# Noti due di essi, il terzo e' determinato automaticamente dalla relazione:
;#   rate_ = packetSize_ / interval_
```

Exponential
(ExpOnOff e Poisson)

```
set Exp0 [new Application/Traffic/Exponential] ;# crea una sorgente
;#   ExpOnOff di nome Exp0
$Exp0 set burst_time_ 1.0 ;# imposta a 1s il tempo medio di
;#   ON di Exp0
$Exp0 set idle_time_ 1.0 ;# imposta a 1s il tempo medio di
;#   OFF di Exp0

$Exp0 set rate_ 100k ;# imposto a 100 kbps la velocita'
;#   media di trasmissione del solo
;#   tempo di ON di Exp0
$Exp0 set packetSize_ 100 ;# imposta a 100 byte la dimensione
;#   del pacchetto di Exp0
```

FTP

```
set FTP0 [new Application/FTP] ;# crea una sorgente FTP di nome FTP0
...
$FTP0 set packetSize_ 100 ;# imposta a 100 byte la dimensione
;# del pacchetto di FTP0
```