VALUTAZIONE DI PRESTAZIONI

- E' utile ad ogni fase della vita di un "computer system": progetto, costruzione, vendita, uso, aggiornamento

- I sistemi da valutare sono cosi' diversificati che non e' possibile definire uno standard in tale area. occorre quindi selezionare:
  - - la corretta misura di prestazioni,
  - - il corretto ambiente di misura,
  - - la tecnica di misura piu' adatta.

- Ogni valutazione richiede un'intima conoscenza del sistema modellato ed un'accurata selezione di metodologia, workload, e tools.

- Il primo passo e' la definizione del problema reale e la sua conversione in una forma in cui sia possibile usare le tecniche e tools piu' adatti.

Principali Errori

- Assenza di obiettivi: non esistono modelli general purpose. Ogni modello deve essere sviluppato con un chiaro obiettivo in mente. E' importante capire il problema ed identificare il problema da risolvere.

- Approccio non sistematico: parametri, variabili da misurare e workload non possono essere scelti in modo arbitrario.

- Performance metrics inadatte

- Workload non rappresentativo delle condizioni reali.

- Tecnica di valutazione inadatta. Le tre tecniche utilizzabili sono Simulazione, Modelli analitici, Misure.

- Trascurare parametri importanti.

- Livello di dettaglio non appropriato. Occorre evitare formulazioni del problema troppo dettagliate o troppo generiche.
- Errata analisi dei risultati.
- Assenza di analisi di sensitivita'.
- Trattamento inadatto dei valori singolari (outliers).
- Inadatta presentazione dei risultati.
- Omissione di assunzioni e limitazioni.

Selezionare una tecnica di valutazione

CRITERIO	MODELLI ANALITICI	SIMULAZIONE	MISURE
Stadio del sistema in cui si può usare la tecnica	In qualunque stadio	In qualunque stadio	Dopo che il sistema è stato realizzato
Tempo richiesto dalla tecnica	Breve (per un esperto analista)	Medio	Variabile a seconda la complessità del sistema
I tools usati nelle varie tecniche	Gli analisti	I linguaggi dei computer	Gli strumenti di misura
Accuratezza	Bassa	Moderata	Variabile a seconda i strumenti usati e il tipo di misura (diretta o indiretta)
Compromesso tra complessità della tecnica e bontà della valutazione	Facile	Moderato	Difficile (perché mettere insieme la soluzione è spesso abbastanza complicato)
Costi	Bassi	Medi	Elevati
Vendibilità	Bassa	Media	Elevata

Misure, Modelli Analitici, Simulazione

quando si possono usare:

•Le misure sono possibili solo se esiste qualcosa di simile al sistema da valutare.

•Modelli analitici e simulazioni possono sostituire le misure in assenza di sistemi disponibili.

che precisione hanno:

•I modelli analitici sono imprecisi (impongono semplificazioni)

•Le simulazioni sono piu' precise ma possono richiedere molto tempo.

•Le misure possono non fornire risultati accurati a causa della unicita' (non ripetibilita') di alcuni parametri.

legame fra i vari parametri

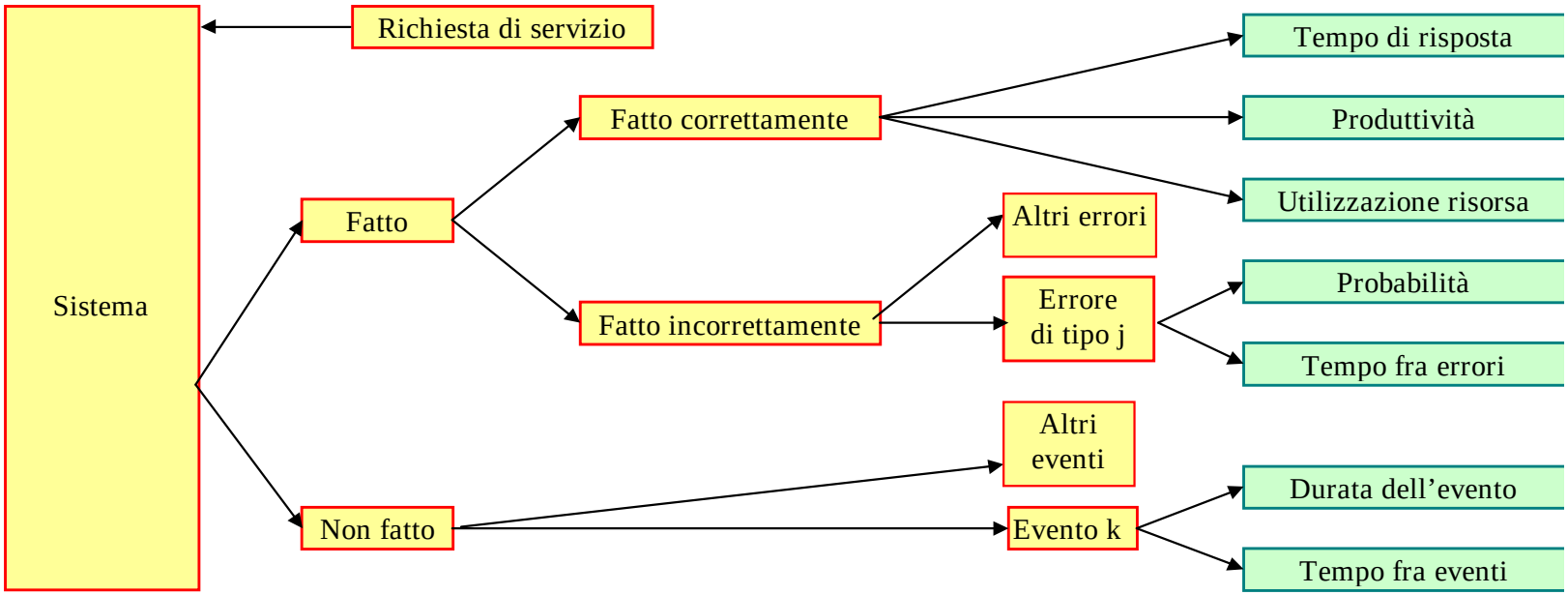
- I modelli analitici permettono di evidenziare l'effetto mutuo di piu' parametri.

- Con le simulazioni a volte non e' chiaro il trade-off fra diversi parametri.

- Le misure rendono difficile interpretare il legame fra vari parametri.

Due o piu' tecniche possono essere usate in modo sequenziale. Ad esempio un modello analitico trova il range adatto dei parametri e la simulazione studia le prestazioni in quel range.

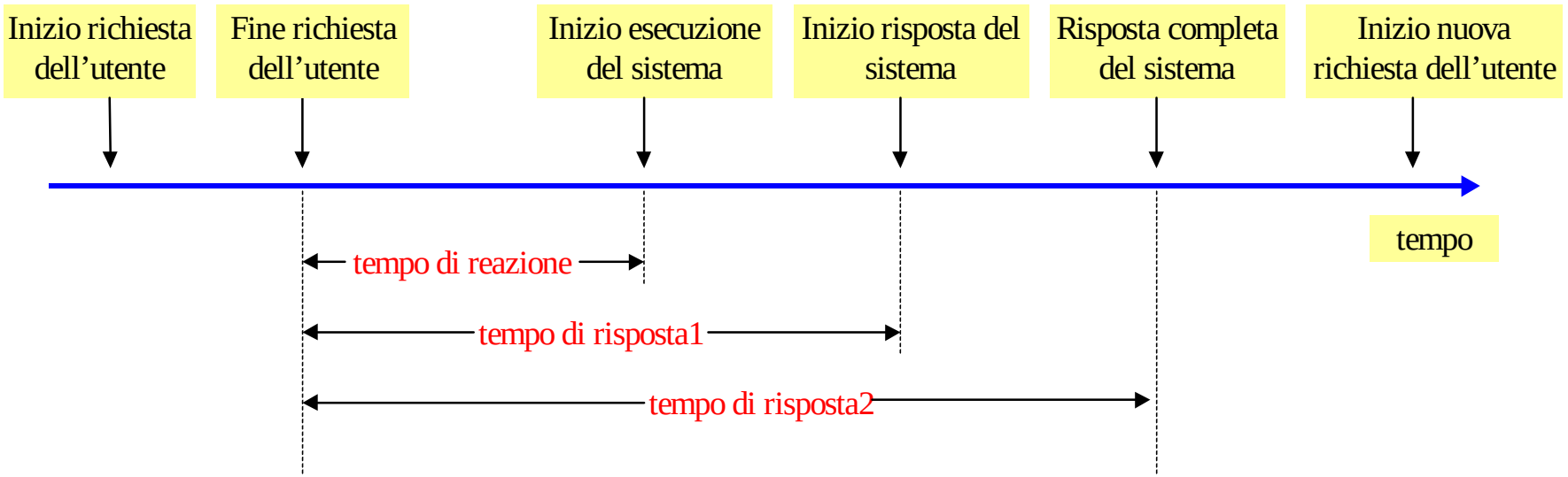
Selezionare le “Performance metrics”.



- Un sistema puo' effettuare un servizio in modo corretto, incorretto, o non effettuarlo.
- Se il sistema esegue il servizio correttamente, le metrics sono chiamate *responsiviness, productivity, utilization.*
- Se il sistema non opera correttamente e' utile classificare gli errori e determinare la probabilita' di ciascuna classe di errori.
- Se il sistema non funziona (unavailable) e' utile classificare i modelli di fallimento e determinare la probabilita' di ciascuna classe.

Performance metrics più usate

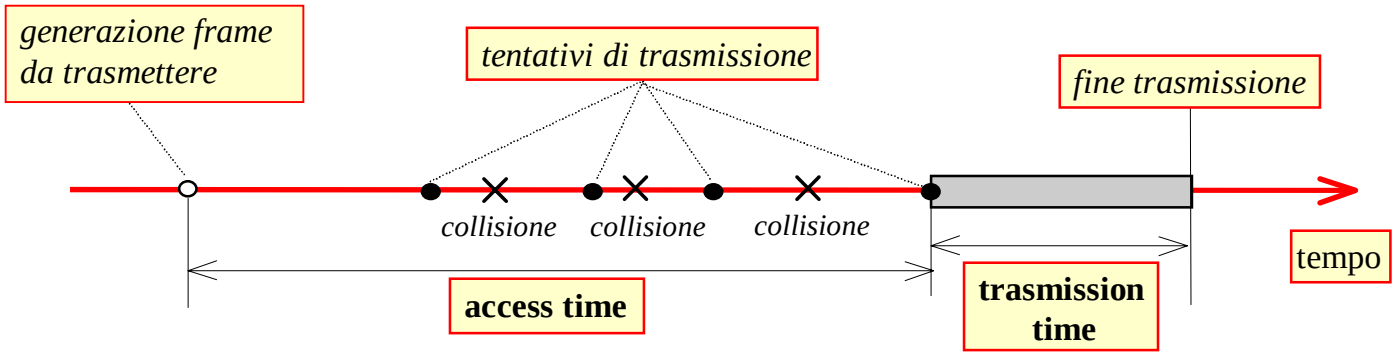
- Response time: intervallo fra la richiesta dell'utente e la risposta del sistema.
- Reaction time: intervallo fra la sottomissione di una richiesta e l'inizio della sua esecuzione.
- Stretch factor: rapporto fra il Response time ad un certo carico e quello a minimo carico.



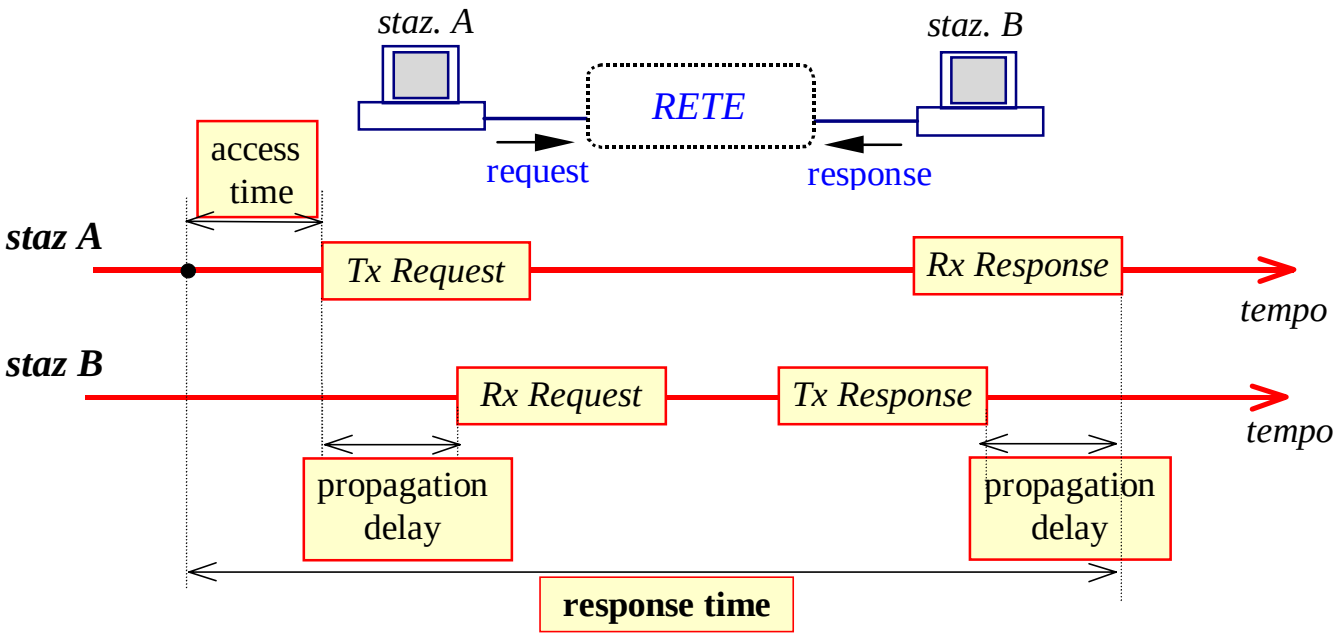
- Con riferimento alle LAN è utile definire “l’Access delay” cioè il tempo che intercorre fra quando un host ha un pacchetto da trasmettere e quando acquisisce l’uso del canale.

- Tale ritardo dipende dalle condizioni del traffico sulla rete e dal particolare protocollo di MAC utilizzato.

- Nel caso di protocolli di tipo CSMA/CD, le collisioni possono influenzare fortemente il tempo di risposta complessivo.



- Il tempo di risposta, nel caso delle reti, può essere definito come il tempo necessario perché un utente riceva la risposta ad una richiesta inviata ad un altro host. Esso è composto da:
- tempo necessario per trasmettere la richiesta (*access time + Tx time*).
- tempi di propagazione del segnale (*andata e ritorno*)
- tempo necessario al destinatario per trasmettere la risposta (*access time + Tx time*)
- Tempo di reazione alla richiesta, da parte del destinatario. (spesso trascurabile)



•Throughput : frequenza a cui le richieste possono essere servite dal sistema.

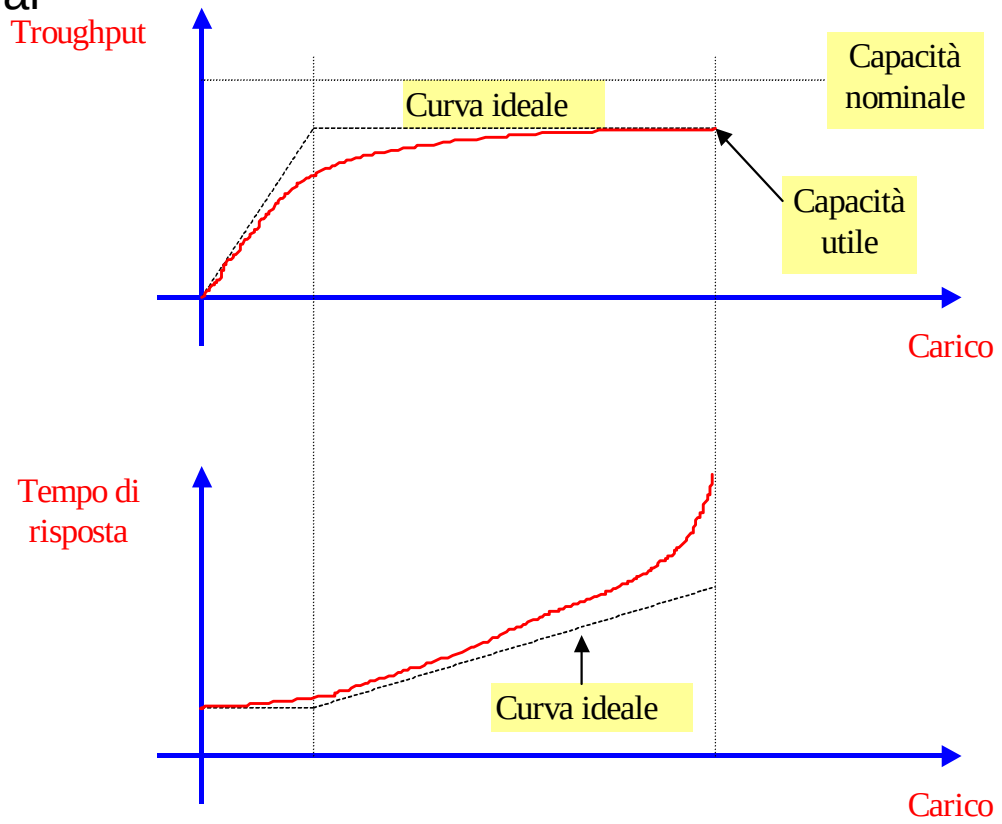
•Nominal capacity: massimo thrpoughput in condizioni ideali (bandwidth per le reti).

•Usable capacity: massimo throughput ottenibile senza superare un delay prefissato.

•Efficienza: rapporto fra massimo throughput (usable capacity) e "nominal capacity".

•Utilizzazione: frazione di tempo in cui una risorsa e' impegnata per servire una richiesta.

•Reliability: tempo medio fra gli errori.

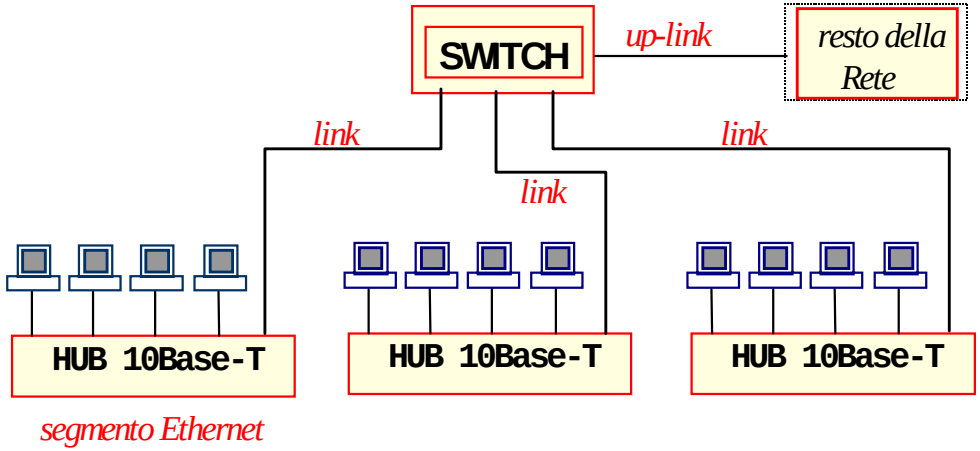


Availability:frazione del tempo in cui un sistema è disponibile per le richieste dell'utente.

WORKLOAD

•Real workload: e' quello osservato su un sistema durante le normali operazioni. Non e' ripetibile.

•Syntetic workload: ha caratteristiche simili a quelle del "Real workload" di cui costituisce un modello. Puo' essere applicato ripetutamente, in maniera controllata.



System under test (SUT): denota l'insieme di componenti che si stanno valutando.

Component under test: denota il singolo componente del SUT considerato.

Il workload va selezionato in base al sistema e non al singolo componente

- Addition instructions: usato nei vecchi sistemi in cui l'addizione era 'operazione piu' usata.

- Instruction mixes: e' una specifica delle varie istruzioni insieme alla loro frequenza d'uso. (Gibson mixes 1959). Misurano l'efficienza della CPU in MIPS o in MFLOPS.

- Kernels: e' rappresentato da una particolare funzione, frequentemente usata, composta da un gruppo di istruzioni. Non usa le istruzioni di I/O.

- Syntetic programs: programmi di test che attraverso dei loop eseguono un numero definito di istruzioni che fanno riferimento anche ad operazioni di I/O.

Implementazione di un simulatore

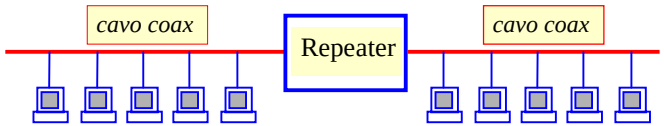
Esempio: rete Ethernet

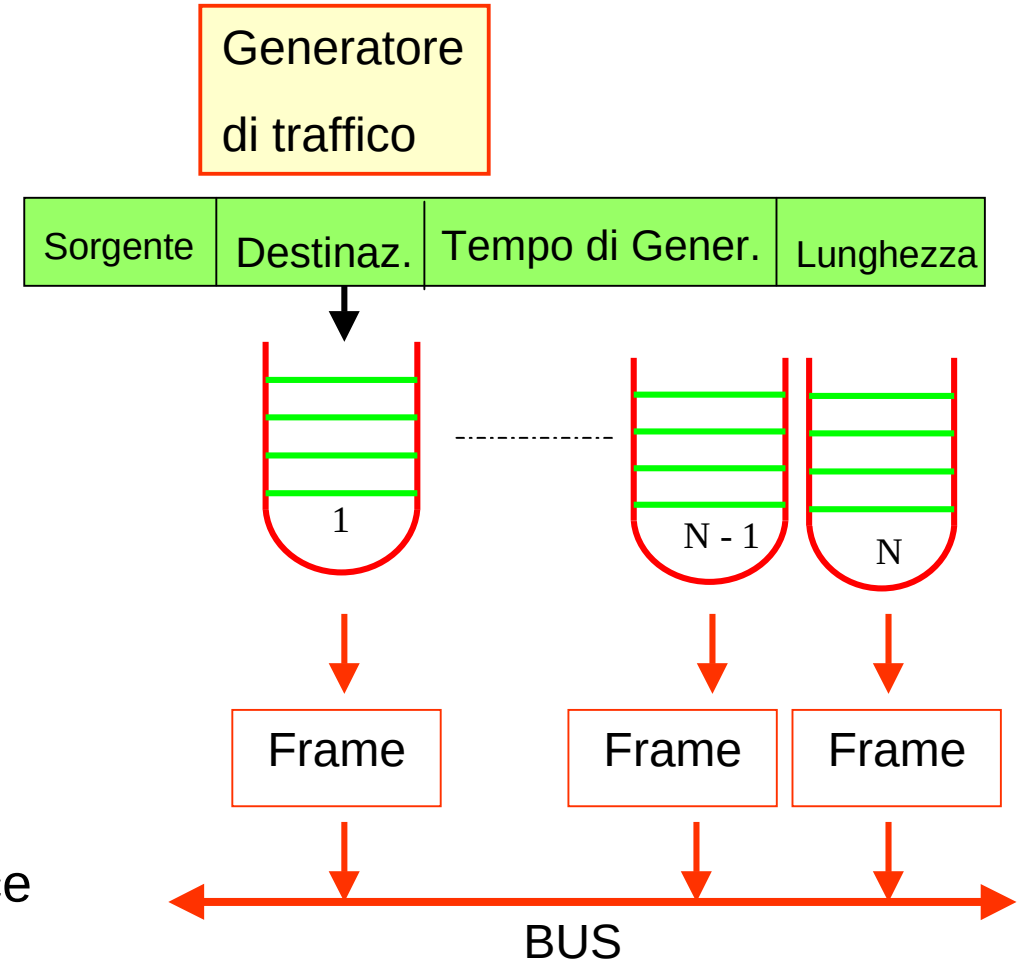
Elementi critici da modellare:

- Generatori di traffico
- code in ogni host
- trasmissione/rivelazione delle collisioni.
- Event scheduler.
- Time.

Le imput routines, initialization routines e report generator non rappresentano un problema.

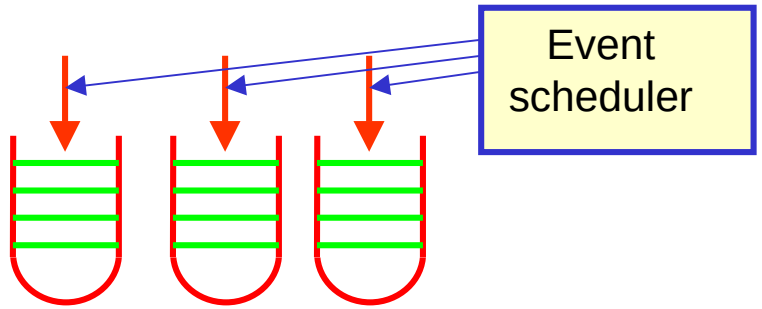
- Il generatore di traffico è una routine che, utilizzando una opportuna funzione di distribuzione temporale, inserisce in coda le frame da spedire.





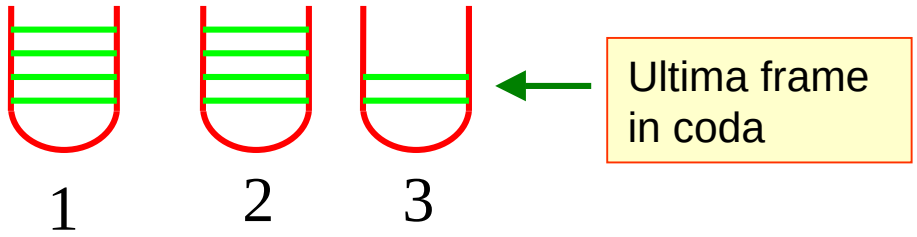
La generazione parallela del traffico

- Tutti i generatori devono generare il traffico in parallelo, in modo indipendente fra loro, in tempi diversi.



Lo schedulatore deve attivare le routines che generano le frames in modo che ciascuna produca una frame al tempo giusto. Poiché i tempi interframe sono variabili, la schedulazione delle generazioni non può essere sequenziale.

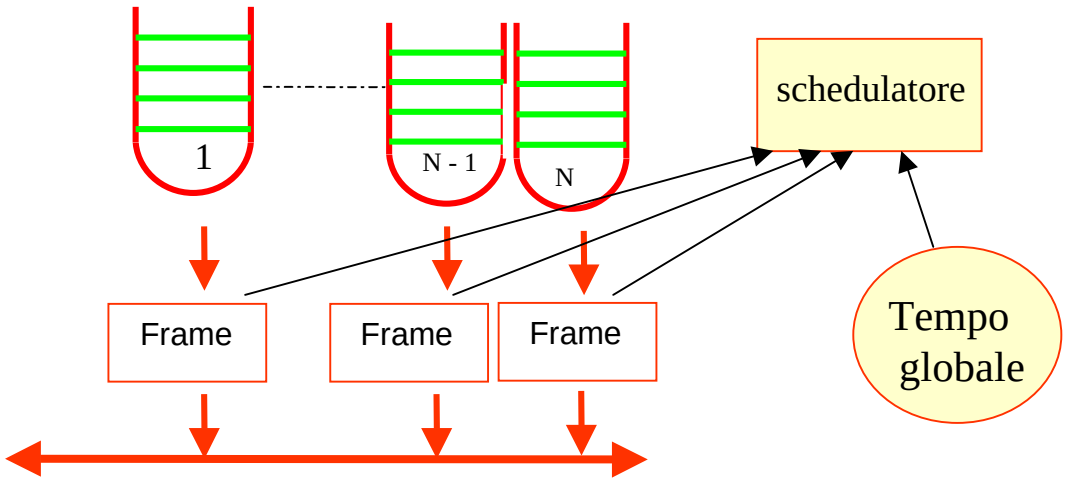
- Poiché il tempo avanza in modo discreto, e non può tornare indietro, lo schedulatore dovrebbe attivare in maniera sequenziale tutti i generatori, misurare i tempi di generazione, ordinarli in ordine crescente e riempire di conseguenza le code.



Soluzione: Lo schedulatore quando la coda è vuota , la riempie per intero (chiamando la funzione generatore). La coda può essere implementata con una struttura statica

Trasmissione delle frames

- Quando il bus è libero, tutti gli host che hanno un messaggio in coda cercano di trasmetterlo.
- Lo schedulatore deve capire quali host hanno un messaggio pronto da trasmettere. Lo fa confrontando il tempo globale con il tempo di generazione dei messaggi in testa in ogni coda.

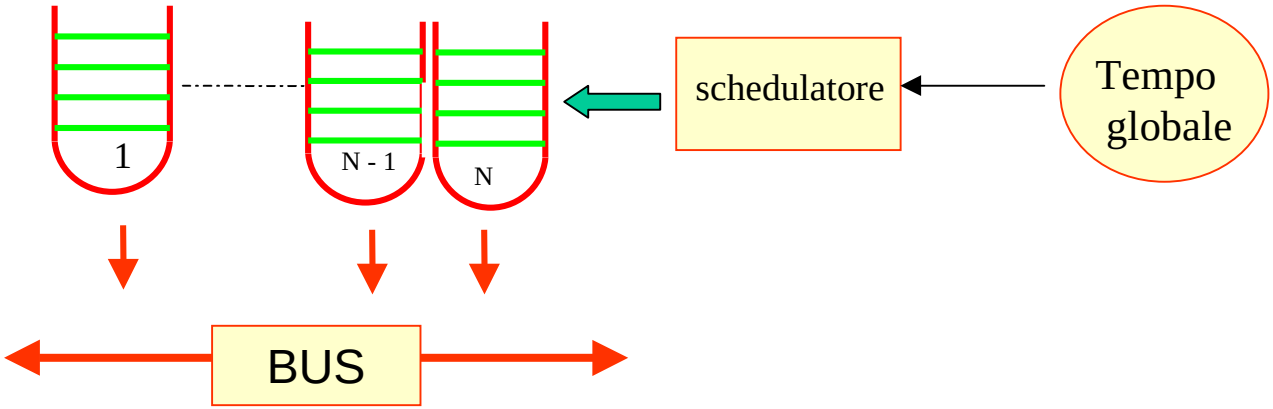


I messaggi in coda, esistono realmente?

Come si capisce se il BUS è libero?

Il messaggio col tempo di generazione minore viene trasmesso, inserendolo nel buffer che rappresenta il bus

Rivelazione delle collisioni



- Quando due o più frames sono trasmesse contemporaneamente (o quasi) si ha una collisione.

- Lo schedulatore rivela una collisione confrontando il tempo di inizio trasmissione con quello di generazione delle frames in testa. Deve esaminare tutte le code.

Tempo_inizio_trasmiss. - tempo di generaz. **<** slot_time ⟹ COLLISIONE

- Algoritmo di backoff esponenziale
- Il tempo di generazione va aggiornato come:

istante_di_collisione + ritardo_di_backoff

Clock e Scheduler

- Il clock è l'elemento di riferimento del simulatore.
- Avanza a scatti, di una quantità pari alla durata dell'evento attivato.
- Lo scheduler confronta il clock con gli eventi possibili e lo incrementa della durata dell'evento da attivare.
- Se l'evento attivato è una trasmissione, verifica se c'è una collisione, incrementa il clock della durata della frame (o del tempo perduto nella collisione), ed effettua le operazioni sui dati, necessarie per le statistiche finali.
- Effettua una nuova scansione delle code per individuare la nuova frame da trasmettere.

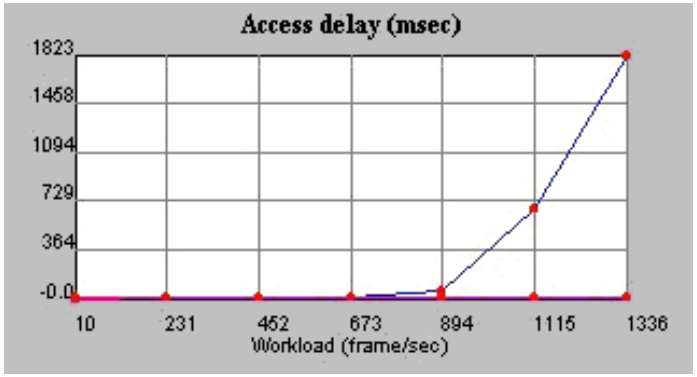
Simulazione stazionaria e transitori

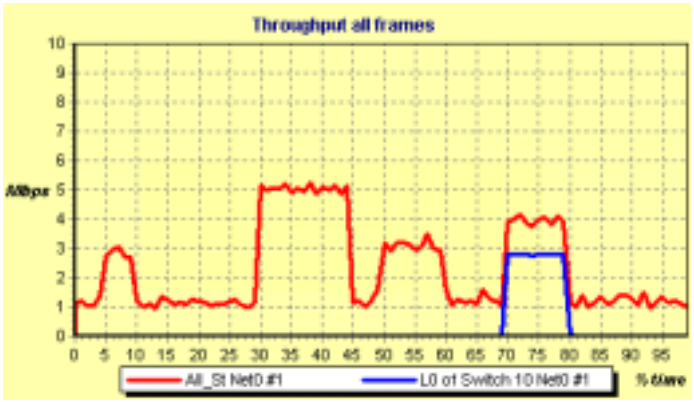
- La simulazione stazionaria ricava il valore degli indici prestazionali prescelti, a regime, per un prefissato punto di lavoro.

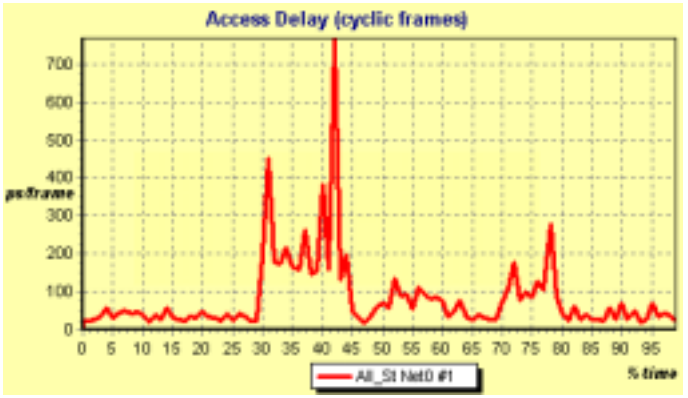
  La presenza di transitori non può essere evidenziata, e può falsare i risultati ottenuti.

- Una simulazione che evidenzi la risposta di un sistema ad eventi transitori deve rappresentare il comportamento del sistema nel tempo.

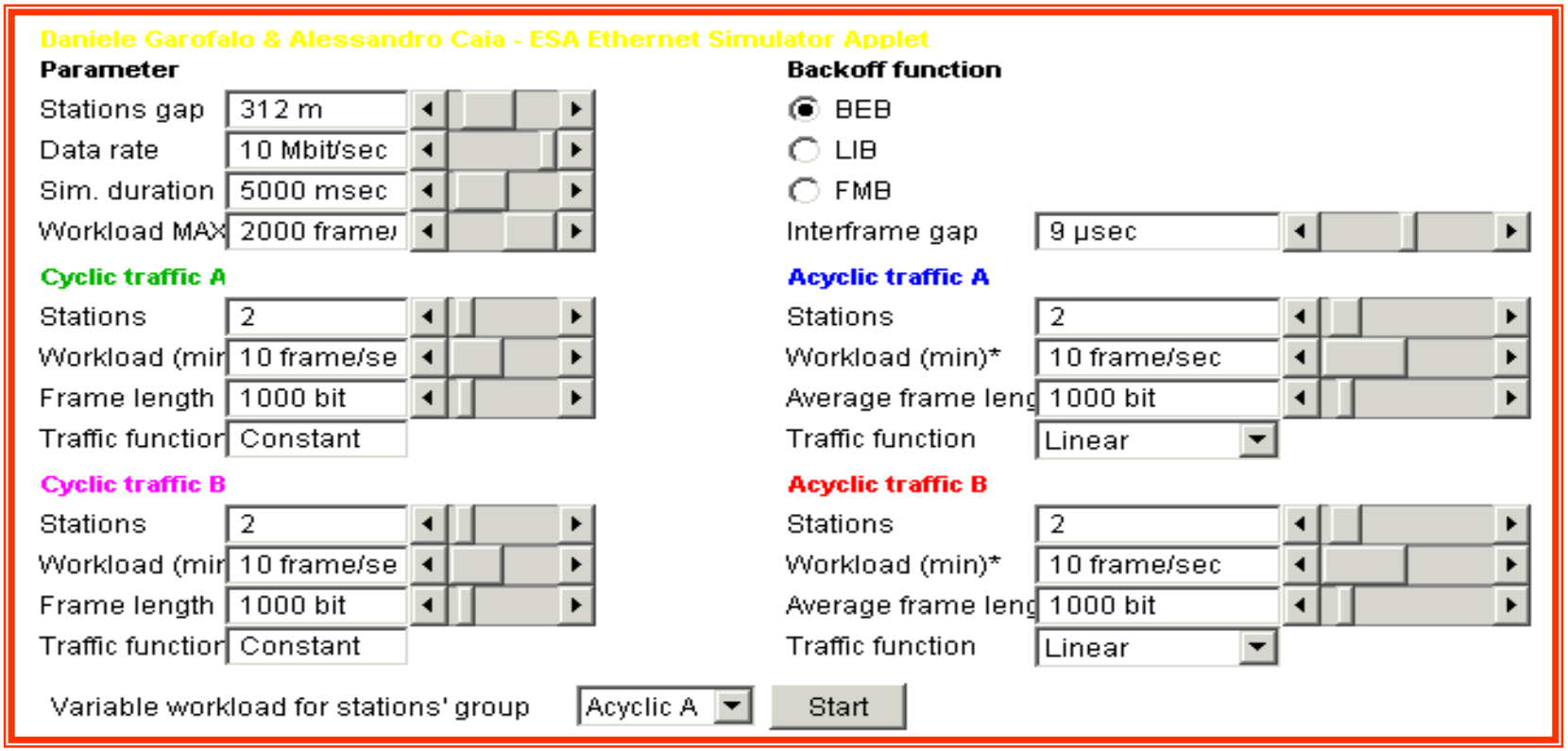
  Ciò non esclude la realizzazione di statistiche o regime, a fine simulazione.



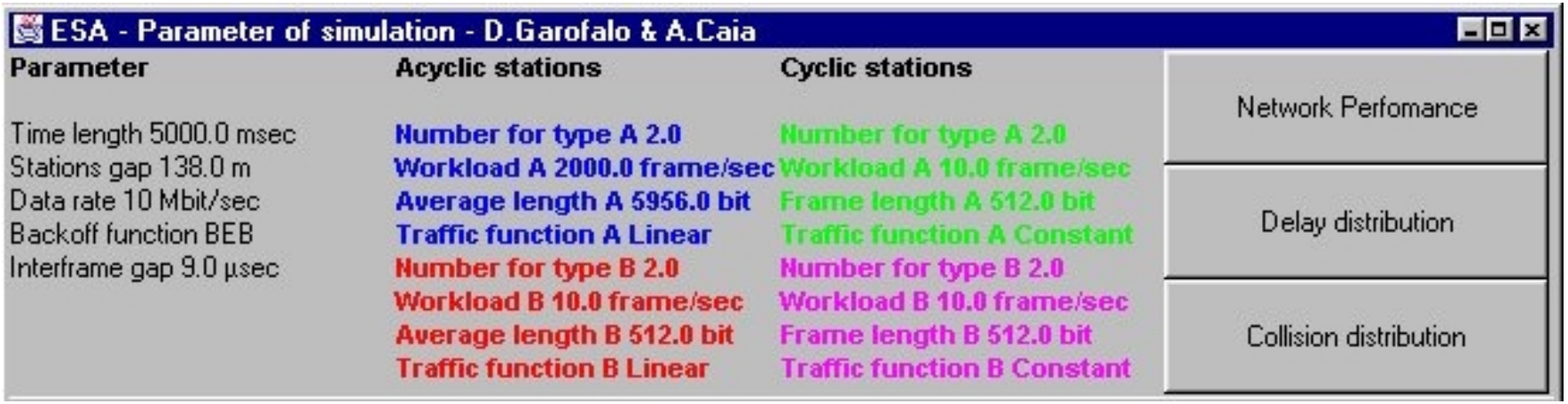


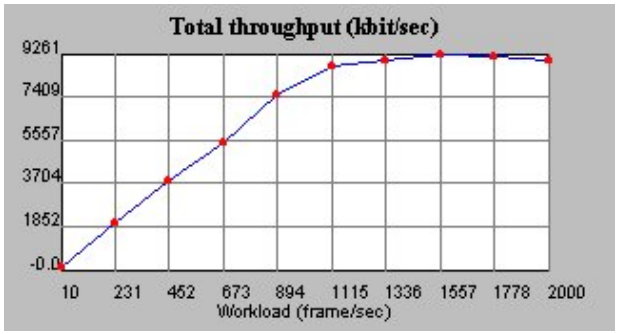


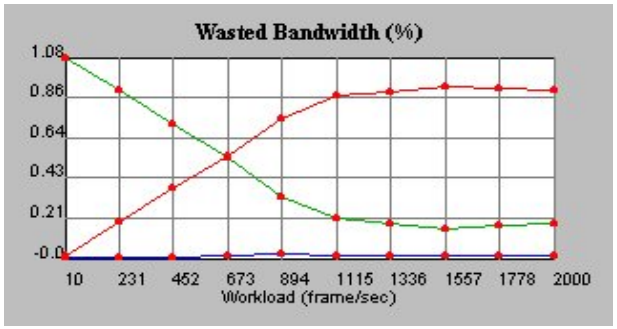
La simulazione stazionaria richiede un'interfaccia di input che permetta di specificare sia la struttura del sistema che le caratteristiche operative a regime.

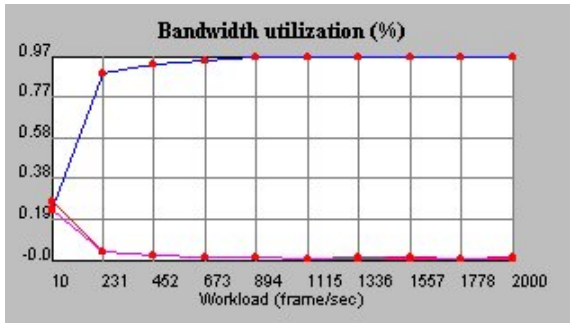


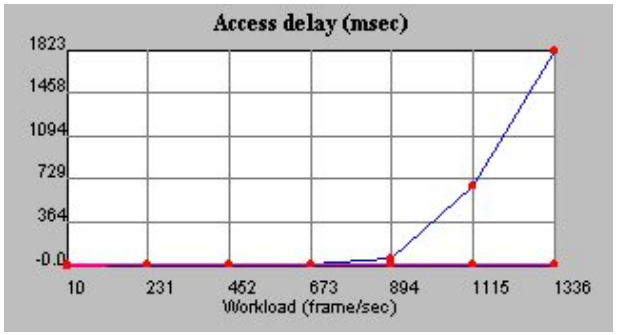
Principali indici prestazionali

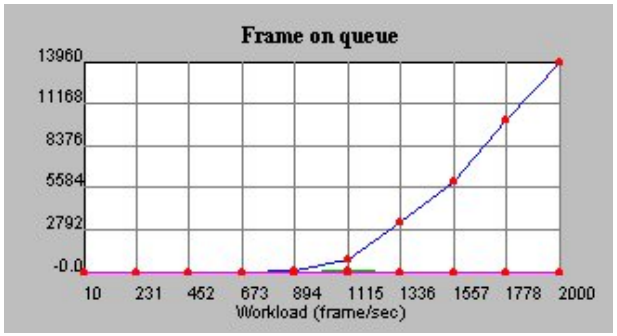


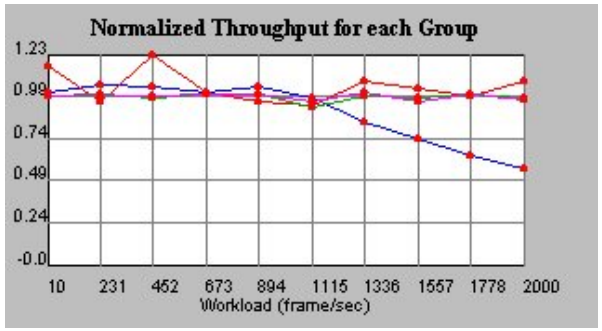




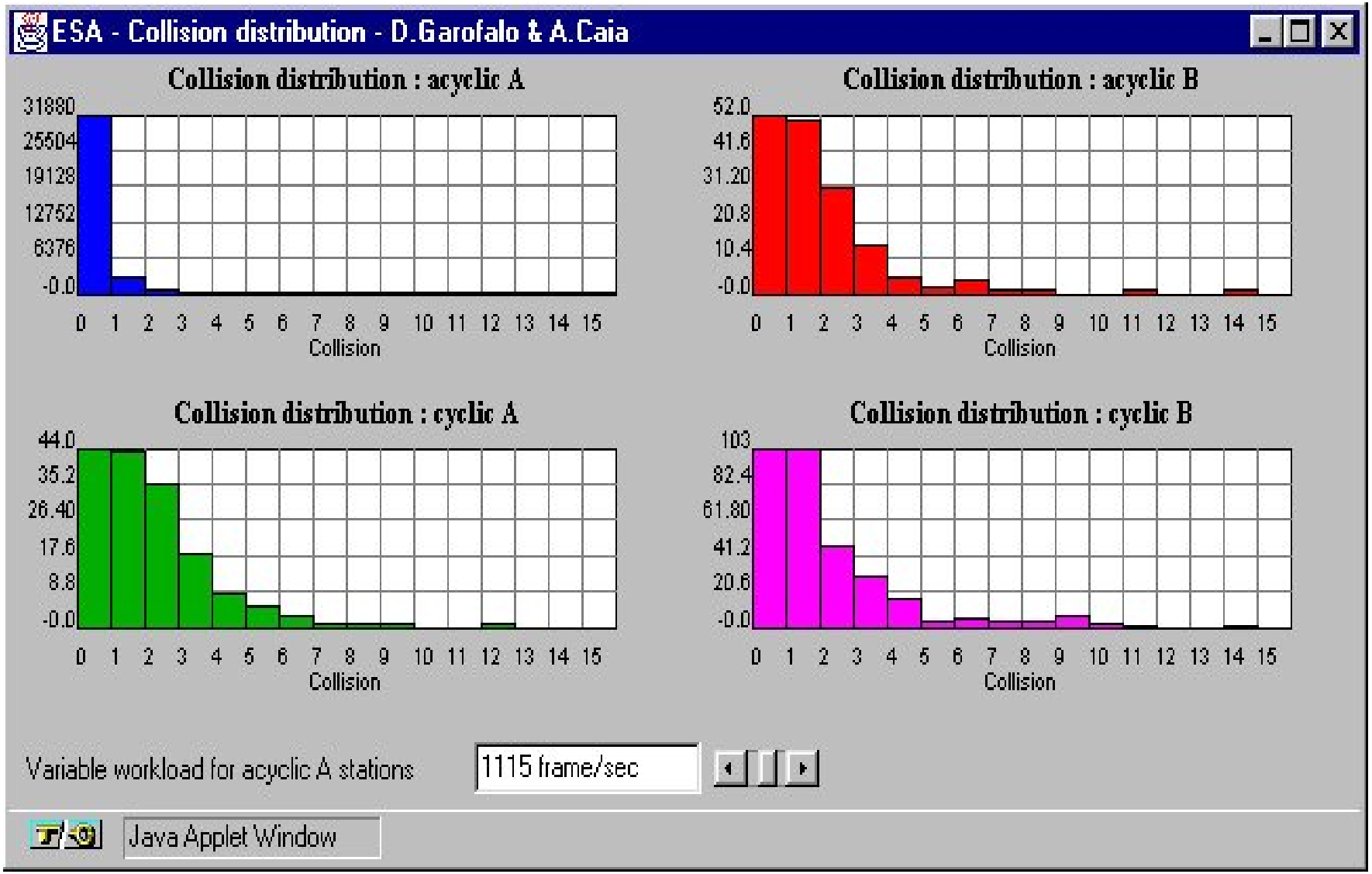




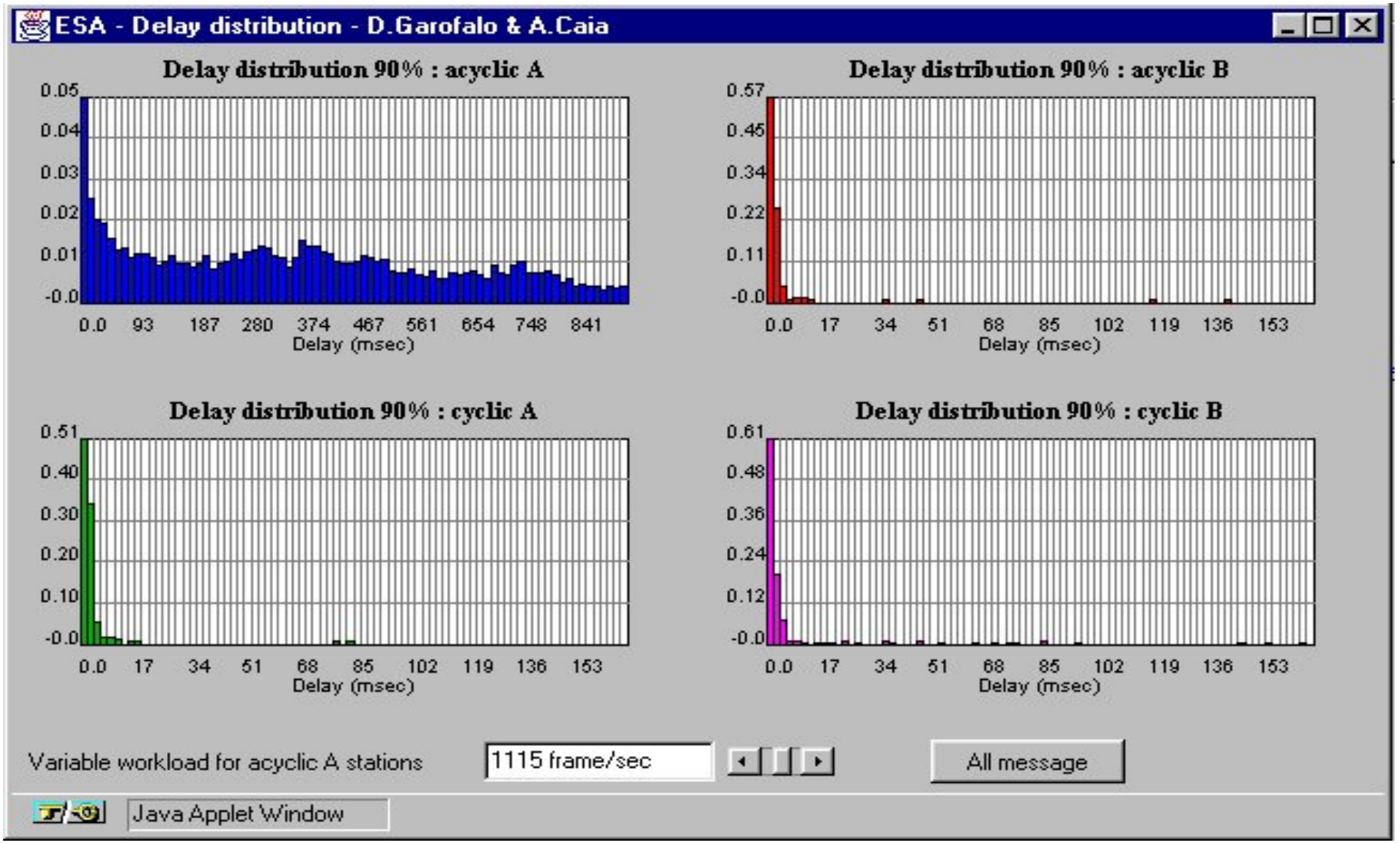




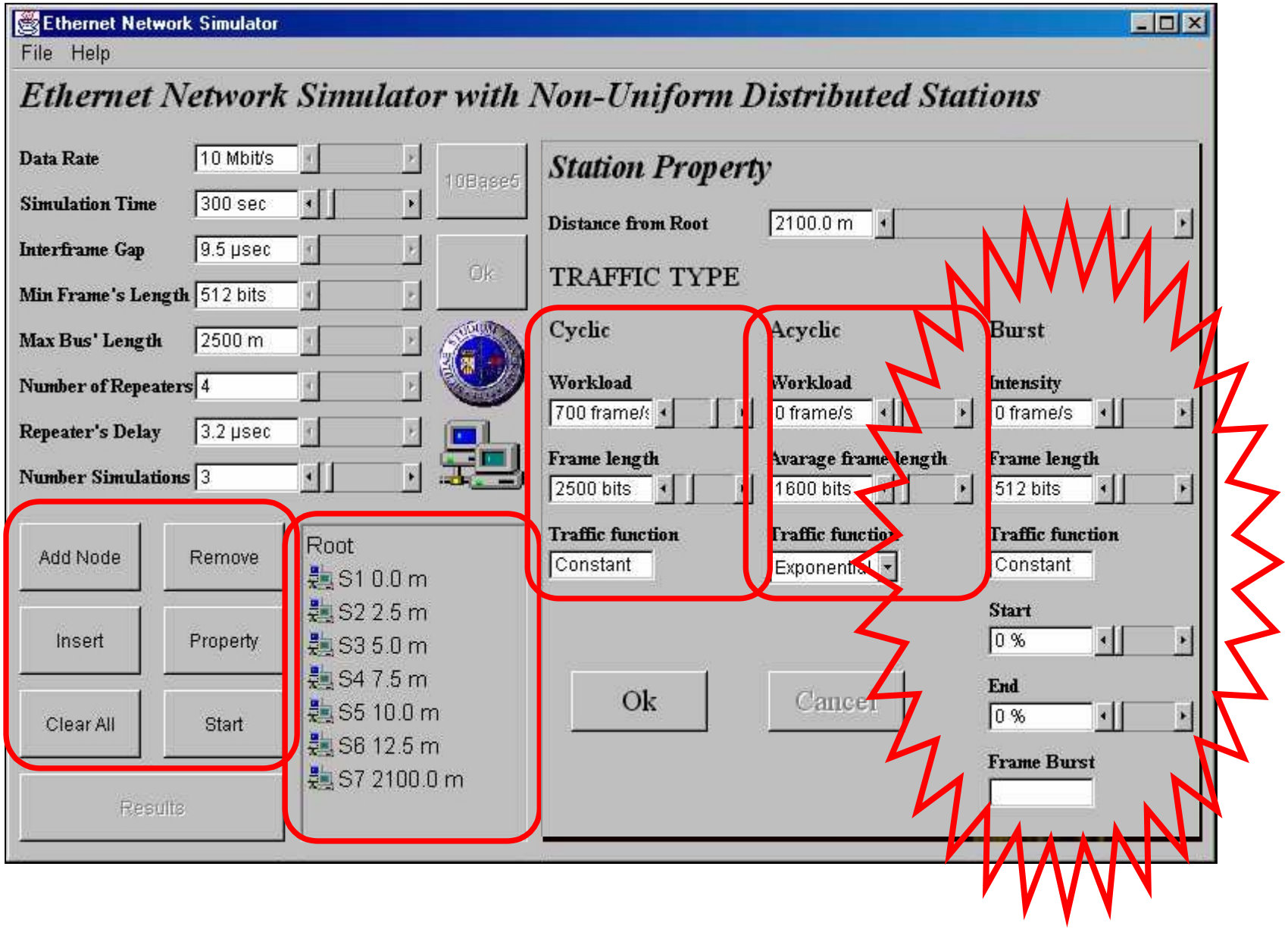
Distribuzione delle collisioni



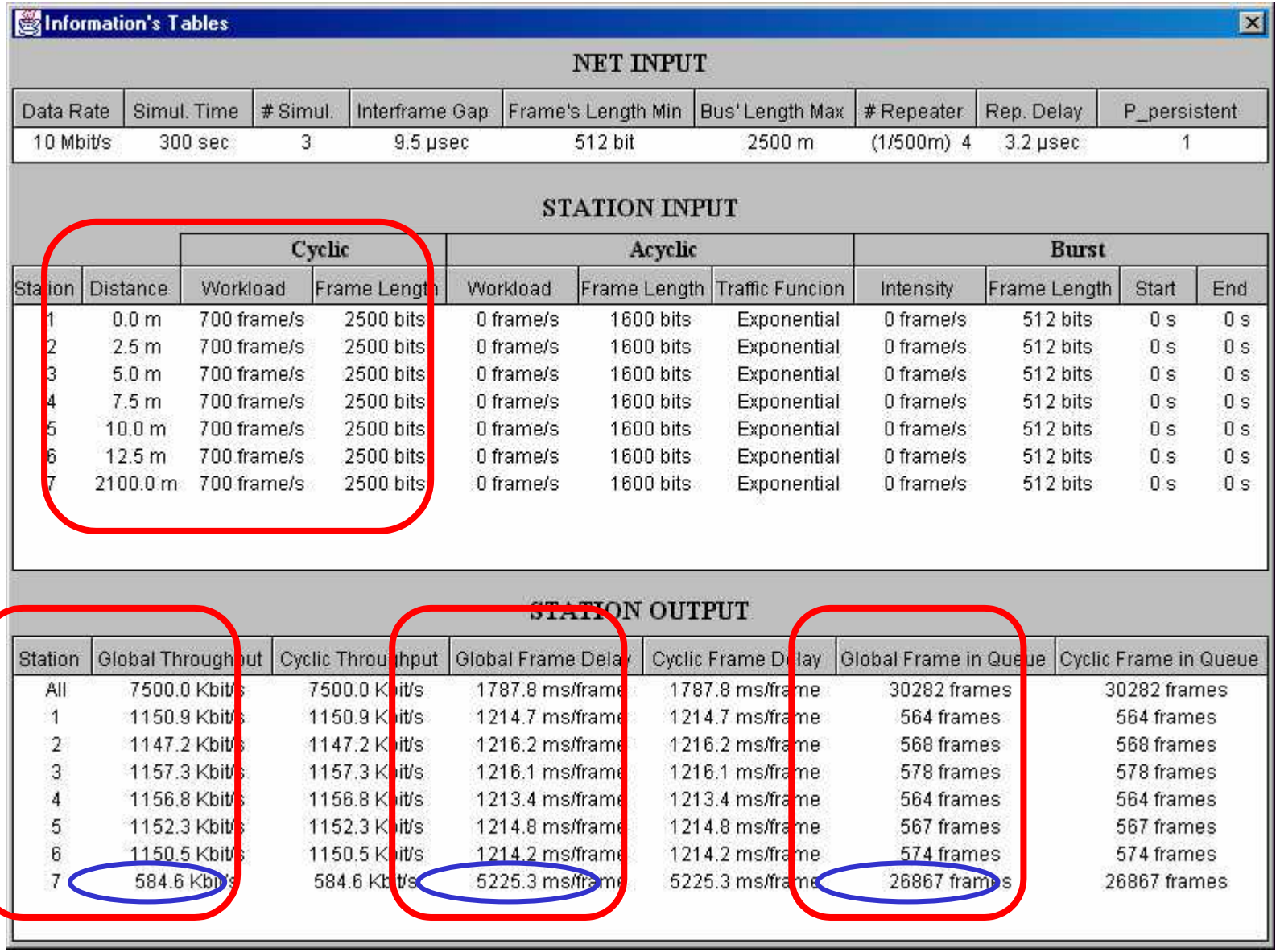
Distribuzione dei tempi di ritardo



Ethernet Network Simulator



Condizioni di traffico uniforme, 7 host, una stazione distante dalle altre,



Information's Tables

NET INPUT

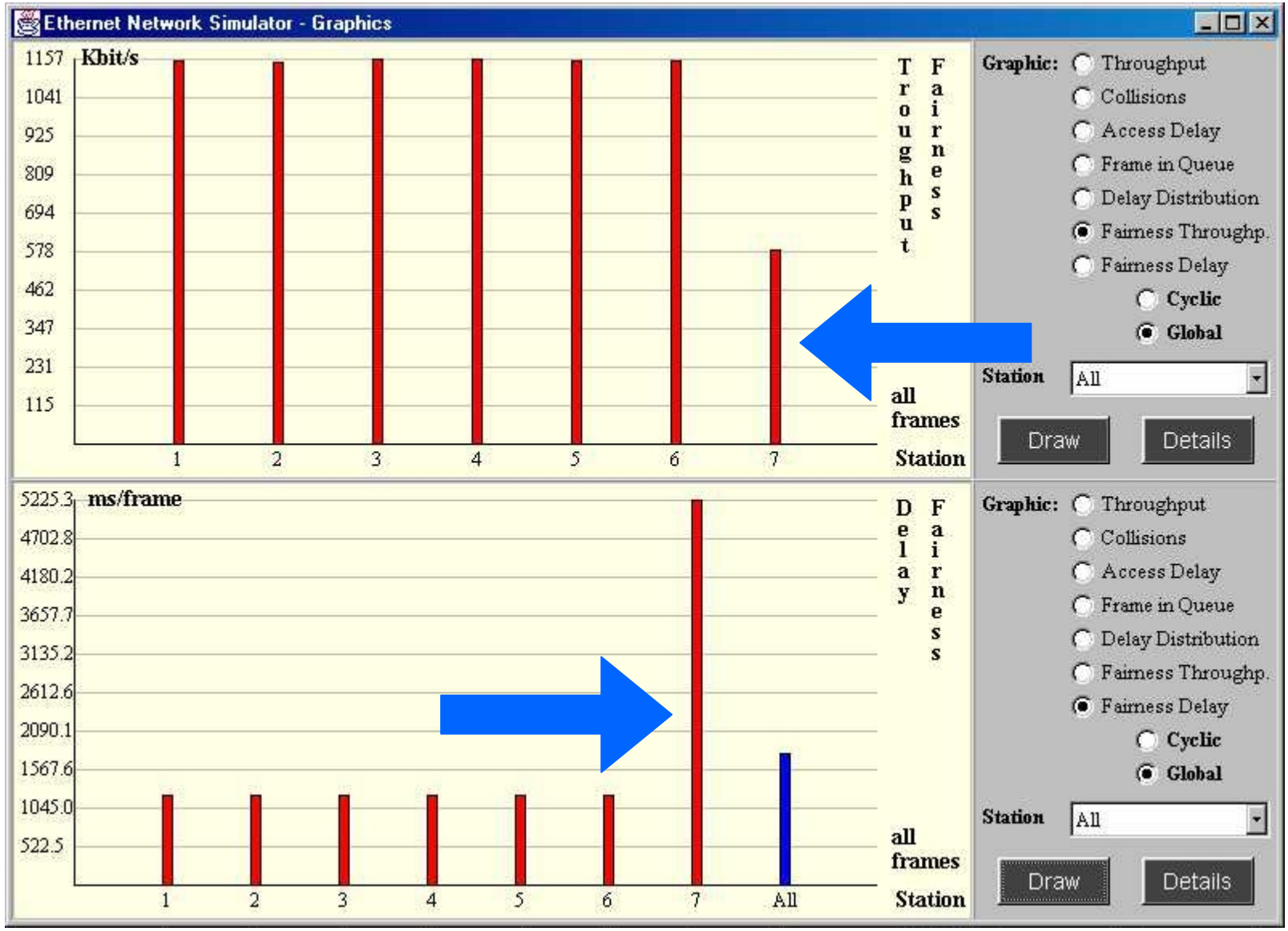
Data Rate	Simul. Time	# Simul.	Interframe Gap	Frame's Length Min	Bus' Length Max	# Repeater	Rep. Delay	P_persistent
10 Mbit/s	300 sec	3	9.5 µsec	512 bit	2500 m	(1/500m) 4	3.2 µsec	1

STATION INPUT

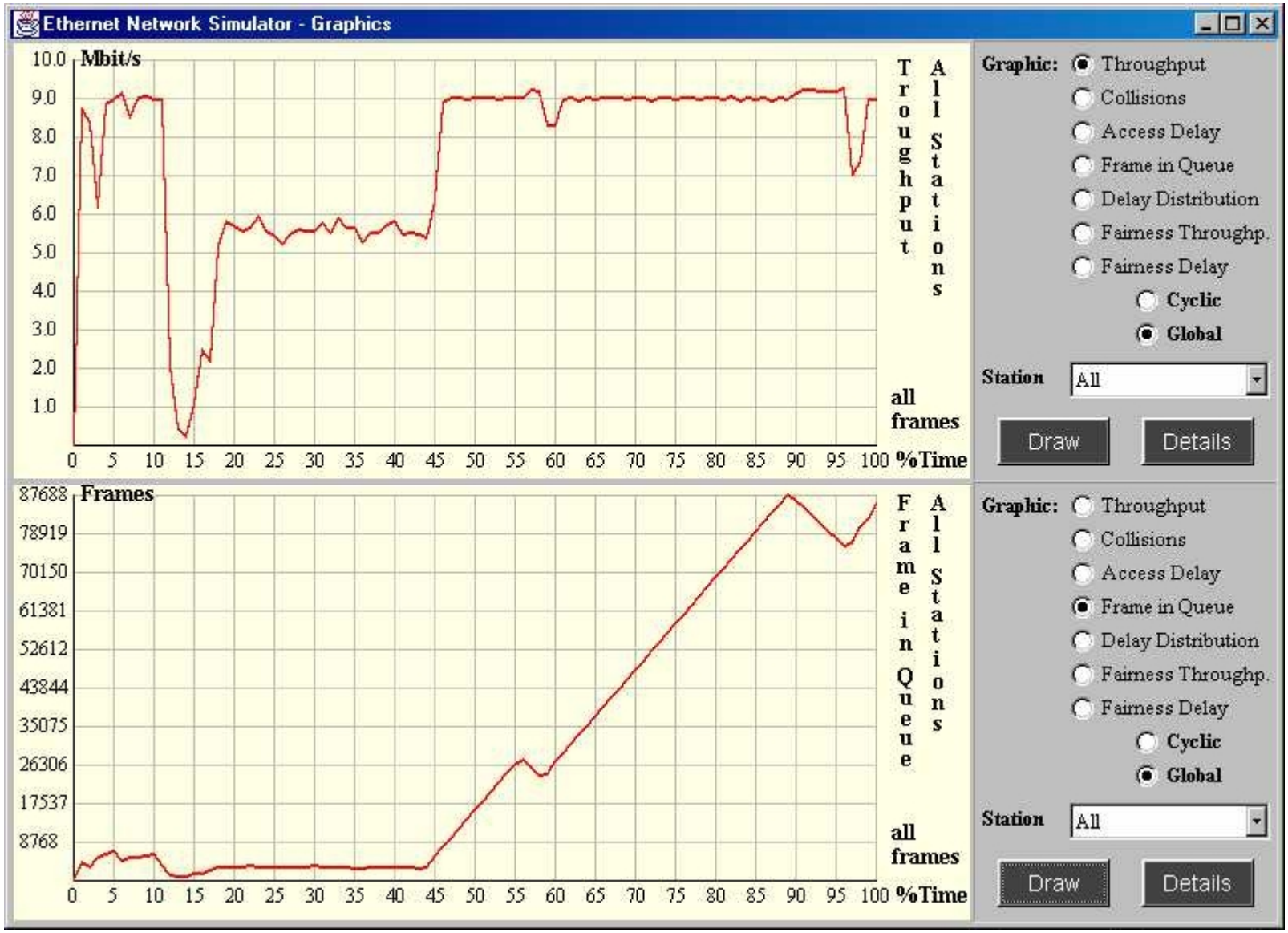
		Cyclic		Acyclic			Burst			
Station	Distance	Workload	Frame Length	Workload	Frame Length	Traffic Funcion	Intensity	Frame Length	Start	End
1	0.0 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s
2	2.5 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s
3	5.0 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s
4	7.5 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s
5	10.0 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s
6	12.5 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s
7	2100.0 m	700 frame/s	2500 bits	0 frame/s	1600 bits	Exponential	0 frame/s	512 bits	0 s	0 s

STATION OUTPUT

Station	Global Throughput	Cyclic Throughput	Global Frame Delay	Cyclic Frame Delay	Global Frame in Queue	Cyclic Frame in Queue
All	7500.0 Kbit/s	7500.0 Kbit/s	1787.8 ms/frame	1787.8 ms/frame	30282 frames	30282 frames
1	1150.9 Kbit/s	1150.9 Kbit/s	1214.7 ms/frame	1214.7 ms/frame	564 frames	564 frames
2	1147.2 Kbit/s	1147.2 Kbit/s	1216.2 ms/frame	1216.2 ms/frame	568 frames	568 frames
3	1157.3 Kbit/s	1157.3 Kbit/s	1216.1 ms/frame	1216.1 ms/frame	578 frames	578 frames
4	1156.8 Kbit/s	1156.8 Kbit/s	1213.4 ms/frame	1213.4 ms/frame	564 frames	564 frames
5	1152.3 Kbit/s	1152.3 Kbit/s	1214.8 ms/frame	1214.8 ms/frame	567 frames	567 frames
6	1150.5 Kbit/s	1150.5 Kbit/s	1214.2 ms/frame	1214.2 ms/frame	574 frames	574 frames
7	584.6 Kbit/s	584.6 Kbit/s	5225.3 ms/frame	5225.3 ms/frame	26867 frames	26867 frames



La stazione lontana risulta penalizzata in quanto a throughput



Quando la rete si congestiona, a causa di vari burst, le code si allungano

NET INPUT

Data Rate	Simul. Time	# Simul.	Interframe Gap	Frame's Length Min	Bus' Length Max	# Repeater	Rep. Delay	P_persistent
10 Mbit/s	300 sec	5	9.6 µsec	512 bit	2500 m	(1/500m) 4	3.2 µsec	1

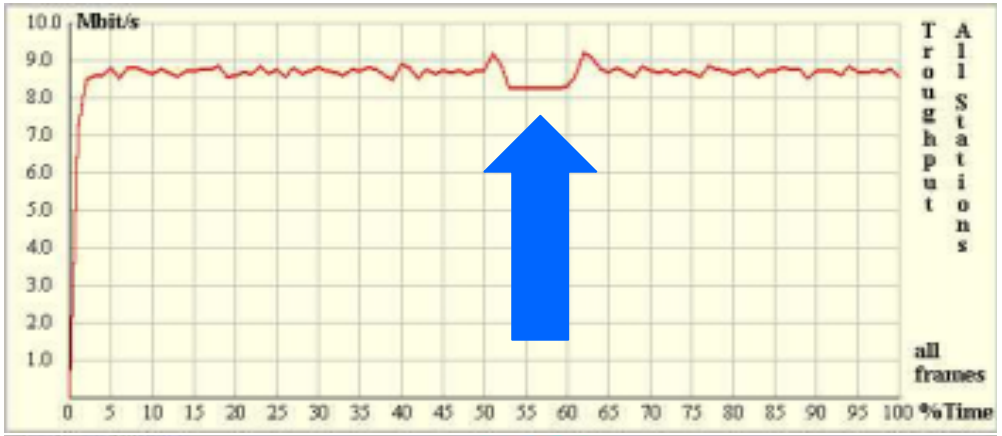
STATION INPUT

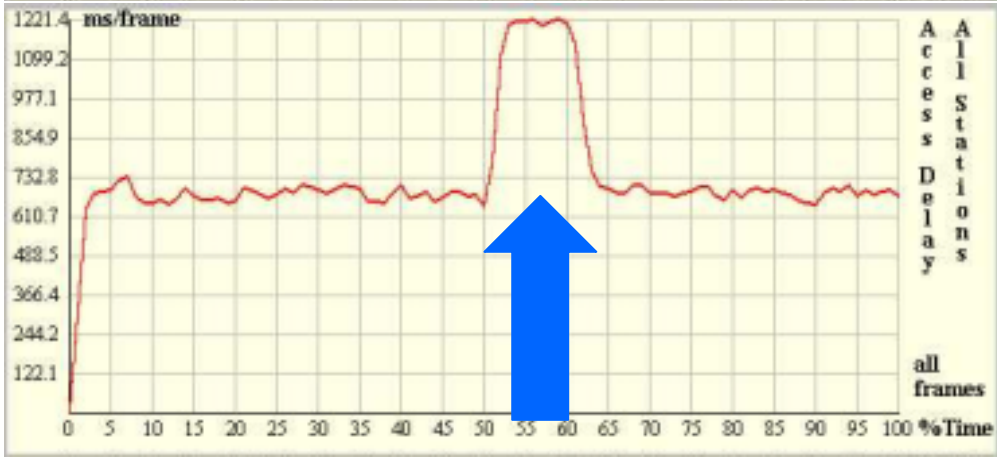
		Cyclic		Acyclic			Burst			
Station	Distance	Workload	Frame Length	Workload	Frame Length	Traffic Function	Intensity	Frame Length	Start	End
1	0.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
2	10.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
3	20.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
4	30.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
5	40.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	250 frame/s	10000 bits	150 s	180 s
6	50.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
7	60.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
8	70.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	0 s
9	80.0 m	250 frame/s	2000 bits	300 frame/s	1500 bits	Exponential	0 frame/s	512 bits	0 s	150 s

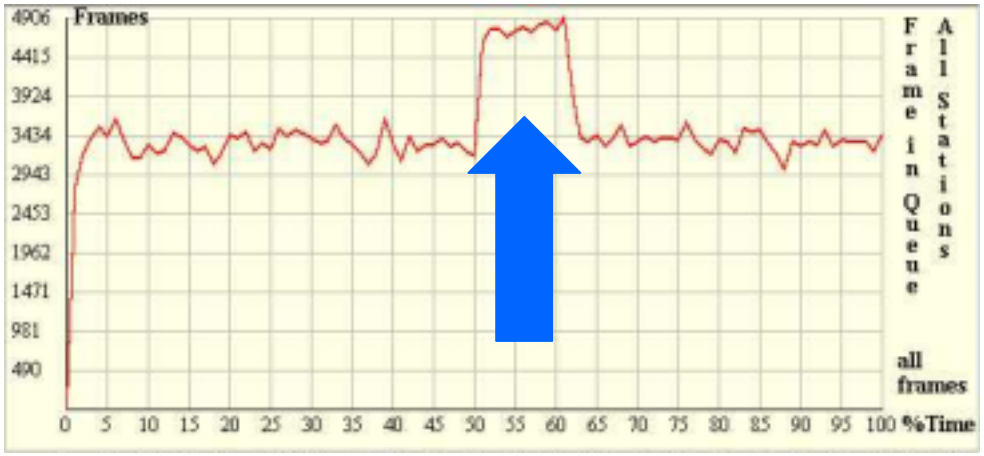
La presenza e durata dei burst viene evidenziata nelle tabelle riassuntive

Il throughput totale diminuisce durante i burst a causa della congestione della rete.

Contemporaneamente si ha un aumento dei tempi di ritardo e del numero di frame in coda.







organizzazione di una simulazione di sistemi modellati attraverso reti di Petri generalizzate.