Valutazione delle prestazioni

Questa dispensa si compone di 10 capitoli che affrontano i vari aspetti della valutazione di prestazioni. Nel seguito, un breve indice descrive gli argomenti trattati in ogni capitolo:

Indice.

1. Introduzione

Con il termine **Valutazione di prestazioni** (performance evaluation) si intende un insieme di attività rivolte alla determinazione delle caratteristiche di un sistema sulla base del suo comportamento.

Attraverso la valutazione di prestazioni di un sistema è possibile sapere “cosa” esso può offrire sotto determinate condizioni operative, in modo da delimitare l’area applicativa dove può essere utilizzato con successo.

La valutazione delle prestazioni di un sistema è utile ad ogni fase della sua vita , cioè nella fase di progetto, costruzione, vendita, uso ed aggiornamento, perché in ognuna di queste fasi ci si trova spesso di fronte alla necessità di fare delle scelte, confrontando le prestazioni di un sistema con quelle di altri sistemi, oppure di capire quali siano le condizioni che permettano di rendere massime le prestazioni.

I sistemi da valutare sono in genere così diversificati uno dall’altro che non è possibile definire una metodologia standard. In particolare, in questa dispensa faremo riferimento solo alle problematiche inerenti le Reti di Calcolatori. Trascureremo quindi gli aspetti legati alla potenza di calcolo dei processori, al set di istruzioni, linguaggio e Sistema operativo usati e ci concentreremo sui problemi

legati alle caratteristiche dei mezzi fisici, protocolli di comunicazione, applicazioni, ambiente operativo. Va comunque sottolineato che molte delle problematiche e tecniche di valutazioni che esamineremo sono applicabili anche ai sistemi di elaborazione non distribuiti.

In generale per una corretta valutazione di prestazioni occorrerà ogni volta selezionare:

- ***la corretta misura di prestazioni***: occorre che i risultati ottenuti dalla valutazione siano quelli giusti, cioè quelli che descrivono in modo completo le prestazioni fornite dal sistema. Ciò richiede una accurata analisi degli indici prestazionali in modo da individuare quelli più significativi. Individuare gli indici prestazionali adatti non è operazione semplice. A parte alcuni indici comunemente usati, uno dei compiti di chi esegue la valutazione di prestazione è verificare se gli indici usati descrivono in maniera completa le caratteristiche del sistema e ricercare eventualmente altri indici prestazionali, utili per lo specifico sistema sotto esame.
- ***il corretto ambiente di misura***: occorre cioè definire bene le condizioni al contorno del sistema in modo da operare in condizioni simili a quelli reali in uso. E' questo un altro punto critico della valutazione di prestazioni che è fortemente influenzata dalle condizioni operative. Per tale motivo, è necessario uno studio approfondito per individuare i parametri che caratterizzano le condizioni operative, in modo da assegnare ad essi i valori corretti.
- ***la tecnica di valutazione più adatta***: le tecniche più usate sono quelle di tipo analitico e quelle di tipo simulativo. Più raramente si effettuano misure sul sistema reale perché in genere sono più difficili da eseguire, anche se i risultati ottenuti attraverso misure reali sono molto più credibili di quelli ottenuti con l'analisi o la simulazione. La scelta di una o dell'altra tecnica è legata al tipo di problema e al tipo di valutazione che si vuole realizzare. Quando la valutazione viene realizzata attraverso il monitoraggio di un sistema reale, bisogna risolvere diversi problemi pratici legati alla difficoltà di analizzare variabili fisiche interne al sistema e alla difficoltà di effettuare misure sui parametri temporali nei sistemi distribuiti (cioè le reti).

Prima di gettarsi a capofitto nella valutazione di un sistema bisogna ricordarsi che ogni valutazione richiede di riflettere su alcuni aspetti chiave:

- ***una intima conoscenza del sistema modellato***: non è possibile valutare un sistema se prima non abbiamo capito veramente a fondo come esso funziona. Infatti, solo così è possibile realizzare un modello che tenga conto di tutti gli aspetti del sistema, anche quelli più subdoli e sottili che spesso sfuggono ad una prima analisi superficiale. Va sottolineata la criticità di questo punto, poichè è sufficiente un errore nella rappresentazione di un aspetto apparentemente secondario, per falsare in maniera significativa i risultati della simulazione. Anche quando si effettua una valutazione mediante misure sul sistema vero è fondamentale sapere come esso è fatto, come funziona e che tipo di comportamento è lecito aspettarsi.
- ***un'accurata selezione della metodologia, del workload, e dei tools***: La metodologia è importante poichè sistemi diversi (o fasi diverse nella vita di un sistema) possono essere analizzati meglio con una metodologia piuttosto che con un'altra. Il workload costituisce il carico che viene offerto al sistema per la valutazione. A seconda del tipo di sistema in esame, il workload assume aspetti diversi; se ad esempio si sta valutando una rete per applicazioni industriali il carico sarà costituito prevalentemente da messaggi periodici (sistema campionato) di piccola dimensione, mentre se si sta valutando una Computer network per office automation, il workload sarà costituito da pacchetti dati, magari di notevole dimensione (file di dati) generati in modo aperiodico.Infine, il tool è lo strumento di simulazione più adatto al problema (spesso vengono utilizzati dei tools general-purpose già pronti. In alternativa è possibile realizzare dei tools dedicati al modello e quindi molto più veloci ed efficienti). La scelta del tool è importante

perchè non esistono strumenti universali, adatti per tutte le occasioni e quindi, a seconda del tipo di valutazione che si vuole effettuare, può essere più adatto un tool invece che un altro.

Il primo passo nella realizzazione delle valutazioni di prestazioni è comunque la definizione del problema reale e la sua conversione (rappresentazione) in una forma in cui sia possibile usare le tecniche e i tools più adatti. E' questo un punto chiave nella valutazione del sistema, che richiede uno studio per mettere in risalto tutti i suoi aspetti significativi (trascurando quegli aspetti che pur essendo importanti nel sistema reale, sono irrilevanti per il tipo di valutazione che si vuole realizzare) e li traduce in forma modellabile attraverso il tool utilizzato.

2. Principali errori

Una volta effettuata la valutazione, i risultati ottenuti vanno esaminati con molta attenzione, ed in modo critico.

Bisogna assolutamente evitare di prendere per buoni dei dati affetti da errore e bisogna sempre mettere in atto strategie di debug che permettono di rivelare la presenza di errori e di validare i risultati ottenuti. La presentazione di risultati errati, oltre che a far perdere credibilità all'autore della valutazione ed alle tecniche utilizzate, può produrre effetti disastrosi sotto diversi aspetti: economici, di affidabilità, di qualità del servizio ottenibile, ecc.

I principali errori che vengono fatti nelle valutazioni delle prestazioni sono i seguenti:

- ***Assenza di obiettivi***: quando facciamo la valutazione di prestazioni, abbiamo chiaramente in mente cosa vogliamo ottenere da essa. Non è possibile valutare un sistema se non abbiamo chiaro quali informazioni vogliamo ottenere. Questo condiziona il modello che sviluppiamo quando vogliamo fare una simulazione: è importante capire il problema ed identificare il modello più adatto al problema che bisogna risolvere. Questo significa che il modello deve essere realizzato per mettere in risalto solo gli aspetti che ci interessano senza dovere rappresentare tutti i dettagli dell'intero sistema. Se invece vogliamo effettuare delle misure su un sistema fisico, dobbiamo valutare quali sono le variabili che ci interessano e predisporre gli opportuni strumenti HW/SW che ci permettono di catturare le informazioni. Questo è un aspetto molto importante, e prima di mettere mano alla realizzazione di un modello (analitico o simulativo) o predisporre un ambiente di misura è fondamentale fermarsi a riflettere sugli obiettivi che vogliamo raggiungere.
- ***Approccio non sistematico***: i parametri, le variabili da misurare e il workload non possono essere scelti in modo arbitrario, ma, avendo bene in mente i risultati a cui miriamo. Dobbiamo ricordarci che valutare un sistema significa ricavare le informazioni che descrivono il comportamento del sistema nelle più disparate condizioni operative. Ciò richiede di effettuare la valutazione cambiando continuamente il valore di alcuni parametri, ma ciò va fatto in modo sistematico e non casuale. In genere, si fissa il valore di alcuni parametri e si varia il valore di un solo altro parametro, in modo da comprendere l'effetto che questo parametro ha sul sistema. Cambiare il valore di due o più parametri contemporaneamente rende più difficile (se non impossibile) capire la relazione fra il valore di un parametro ed il comportamento di un sistema.
- ***Performance metrics inadatte***: le performance metrics (indici prestazionali) sono tutti i parametri che noi misuriamo in una valutazione di prestazioni (throughput, tempo di ritardo, affidabilità del sistema, ecc.) per descrivere il comportamento di un sistema.Una scelta inadatta di tali parametri fornisce una valutazione incompleta del sistema.
- ***Workload non rappresentativo delle condizioni reali***: Con riferimento alle reti, il workload rappresenta il carico di lavoro della rete inteso soprattutto come numero di pacchetti dati da trasferire nell'unità di tempo, ma anche la dimensione dei pacchetti e il numero di nodi presenti nella rete. E' essenziale fare in modo che il workload che utilizziamo per la valutazione del

sistema abbia una corrispondenza con il workload reale che il sistema troverà in condizioni operative, altrimenti la valutazione effettuata non avrà alcuna utilità. Le prestazioni che misuriamo debbono essere correlate ad uno scenario reale.

- ***Tecnica di valutazione inadatta***: le 3 tecniche utilizzabili sono: la simulazione, i modelli analitici, le misure. In base ai risultati che vogliamo ottenere è importante utilizzare la tecnica adatta: normalmente vengono utilizzate le simulazioni e i metodi analitici nelle fasi preliminari di valutazione di un sistema, mentre le misure si usano solo quando il sistema è fisicamente disponibile e quindi dopo la sua realizzazione.
- ***Trascurare parametri importanti***: se nel fare il modello non si tengono in considerazione alcuni parametri che possono invece avere una notevole importanza, allora si otterranno dei risultati che poi saranno notevolmente differenti da quelli veri. Pertanto è fondamentale individuare i parametri che influenzano in maniera sostanziale il comportamento di un sistema e trascurare, eventualmente, quelli il cui valore è irrilevante.
- ***Livello di dettaglio inappropriato***: occorre evitare formulazioni del problema troppo dettagliate o troppo generiche. Infatti nell'eccessivo dettaglio si rischia spesso di perdersi nei particolari e magari poi di non mettere in evidenza gli aspetti più importanti: il modello che si ottiene in questo caso è complicatissimo, con in genere un numero elevato di errori. Nel caso di un simulatore sarà un problema farlo girare. D'altro canto l'eccessiva genericità è anch'essa da evitare: non consente di mettere in risalto gli aspetti da valutare.
- ***Errata analisi dei risultati***: la valutazione fornisce dei risultati, ma bisogna poi capire ciò che essi rappresentano, cioè dai risultati bisogna estrapolare il comportamento del sistema. A parte i problemi legati agli errori o alle approssimazioni che noi facciamo, a volte c'è il problema di capire se qualche risultato è oppure no significativo, cioè se ha senso oppure no: in modo da evitare di prendere grosse cantonate. Ciò si collega ancora al problema della correttezza dei risultati ottenuti dalla simulazione, cioè un'attenta analisi dei risultati consente di capire se essi hanno senso e se quindi la valutazione è stata condotta correttamente. Va comunque sottolineato che, anche se la simulazione è stata condotta correttamente fornendo risultati esatti, l'analisi dei risultati potrebbe essere errata. Ciò può derivare da una scarsa conoscenza del sistema, per cui non ci si riesce a spiegare alcuni aspetti del suo comportamento, oppure da errata interpretazione delle cause che hanno determinato i valori ottenuti degli indici prestazionali.
- ***Assenza di analisi di sensitività***: l'analisi di sensitività permette di capire come la variazione di un parametro iinfluenzi il comportamento del sistema: ci sono parametri che influenzano poco il comportamento del sistema, e parametri particolarmente critici (una cui leggera variazione comporta una notevole variazione del comportamento del sistema). Ad es. nel calcolo del tempo di ritardo nelle reti con trasmissioni di messaggi a diversa priorità, il carico ad alta priorità influenza moltissimo il comportamento della rete, mentre il carico a bassa priorità lo influenza meno. Il comportamento della rete pertanto dipenderà fortemente dal carico ad alta priorità, a cui il sistema darà la precedenza rispetto a quello a bassa priorità. Se non si è coscienti della diversa sensibilità di un sistema rispetto a certi parametri, sarà difficile comprendere a fondo il suo comportamento.
- ***Trattamento inadatto dei valori singolari (outliers)***: nell'effettuare delle campagne di misure capita che ci sia ogni tanto qualche punto, detto valore singolare, il cui valore risulta completamente scorrelato dagli altri. E' necessario capire se si tratta di un evento anomalo, legato a errori nella misura o al concatenarsi di eventi particolari (ed in questo caso tale valore deve essere scartato in modo da non falsare i risultati) o se è il campanello di allarme di eventuali Bug nei modelli di simulazione o nel protocollo.
- ***Inadatta presentazione dei risultati***: i risultati ottenuti possono essere tantissimi e per meglio valorizzare il lavoro fatto è estremamente importante presentare i risultati in modo adatto. A tale

scopo è bene non limitarsi a delle semplici tabelle bensì è meglio graficare i risultati: in questo modo è possibile presentare chiaramente i risultati e sono anche possibili confronti con altre curve in condizioni di carico differente per lo stesso sistema. E' inoltre importante che le curve riportino negli assi le grandezze espresse nel modo più significativo, in modo da semplificarne la lettura. Per esempio, il workload può essere espresso in Kbytes/Sec. Oppure in percentuale della banda occupata (che spesso è più significativa del valore assoluto del workload).

- ***Omissione di assunzioni e limitazioni***: Un ultimo punto va preso in considerazione. Spesso quando si fanno valutazioni di prestazioni si omettono, volontariamente o involontariamente, alcune assunzioni e le limitazioni di base, che invece dovrebbero sempre essere ben chiare. Infatti, chi analizza i risultati delle valutazioni effettuate deve sempre conoscere le assunzioni e limitazioni in modo da avere una chiave per interpretare tali risultati. L'arte del *retail game* (gioco del venditore) è l'arte di interpretare i risultati della valutazione in modo da presentarli sempre in positivo, cioè in modo da dimostrare che in nostro sistema è migliore di quello degli altri, senza mentire sui risultati, ma semplicemente presentando bene tutti gli aspetti positivi del prodotto e minimizzando gli aspetti negativi. Tale atteggiamento è assolutamente da escludere nel nostro caso: non dobbiamo vendere un sistema ma dobbiamo solamente comprenderne il funzionamento.

3. Selezione della tecnica di valutazione

Vediamo quali sono i criteri in base ai quali è opportuno selezionare una delle 3 tecniche 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/Alta a seconda della accuratezza del modello	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 (credibilità)	Bassa	Media/Alta	Elevata

Quando si possono usare le varie tecniche di valutazione?

- Le **misure** sono possibili solo se esiste il sistema da valutare o qualcosa di simile al sistema da valutare. La misura è una forma di valutazione di prestazioni normalmente complessa e difficile che richiede la disponibilità di adeguati strumenti di misura ed una stretta correlazione con l'HW ed il SW del sistema. Ciò implica un'approfondita conoscenza del sistema (non facile da acquisire) . E' un approccio importante perchè i risultati ottenuti non si riferiscono ad un modello

del sistema (come nel caso delle simulazioni o delle analisi) ma al sistema reale e pertanto sono estremamente significativi. Presenta però diverse limitazioni:

- Innanzitutto occorre evitare che i sistemi HW/SW usati per la valutazione interferiscono col sistema sotto misura falsandone il comportamento e quindi i risultati ottenuti.
- Agendo su un sistema fisicamente reale è in genere difficile apportare modifiche alle strutture, per cui la valutazione fa riferimento ad un ben preciso contesto applicativo. Ad esempio, nel caso si stia valutando una rete, sarà abbastanza difficile operare delle modifiche sul numero di nodi, sulla lunghezza della rete, sul bit rate, etc.
- Non è facile valutare il comportamento in condizioni anomale (ad esempio guasto di qualche nodo) poichè tale condizione non è facile da realizzare(a meno di non decidere di sacrificare parte del sistema per la valutazione di prestazioni).

I modelli analitici e le simulazioni possono sostituire le misure in assenza di sistemi disponibili.

- L'**approccio analitico** consiste nella rappresentazione in forma matematica del sistema e nella sua valutazione attraverso opportune formule risolutive. Ampiamente usati sono i modelli basati sulla teoria delle code che permettono di utilizzare risultati analitici già collaudati.
  Il principale vantaggio dell'approccio analitico è insito nell'approccio stesso e consiste nella trasparenza del modello che essendo rappresentato in forma analitica può essere facilmente verificato da chiunque. Per tale ragione i modelli analitici sono preferiti quando si deve valutare un nuovo sistema ed occorre valutare i risultati. Questi possono essere facilmente verificati e pertanto sono molto credibili.
  Per contro il modello analitico si presta solo a rappresentare aspetti molto generali di un sistema (e questo potrebbe essere in qualche caso un aspetto positivo) che sono utili nella prima fase di valutazione di un sistema.
  L'approccio analitico limita inoltre fortemente il tipo di valutazione che è possibile realizzare, a causa della difficoltà a rappresentare analiticamente, in modo corretto e con il sufficiente livello di dettaglio, i vari indici prestazionali.

- L'**approccio simulativo** si basa sulla realizzazione di un modello del sistema e sulla valutazione attraverso un opportuno strumento di simulazione.
  Il modello, va realizzato con una tecnica adatta al tool di valutazione impiegato. Il principale vantaggio della simulazione risiede nel fatto che se il modello è ben realizzato e sufficientemente dettagliato è possibile ottenere dei risultati molto simili a quelli che sarebbero stati forniti dal sistema reale. Il modello può inoltre essere realizzato in modo da evidenziare specifici aspetti del sistema che si vogliono investigare. E' quindi possibile variare le condizioni a contorno e valutare come queste influiscono sul comportamento del sistema. Inoltre è possibile rappresentare anche sistemi molto estesi con un gran numero di nodi (cosa praticamente impossibile nel caso di misure su sistemi reali, a causa dell'eccessivo costo del sistema).
  Mediante la simulazione è possibile analizzare il comportamento del sistema sia in condizioni stazionarie (a regime), che in transitorio. Normalmente il sistema è valutato a regime, cioè in condizioni operative stabili, ed in tal caso occorre prestare attenzione a non includere i risultati relativi a fasi transitorie.

3.1 Precisione offerta da varie tecniche di valutazione

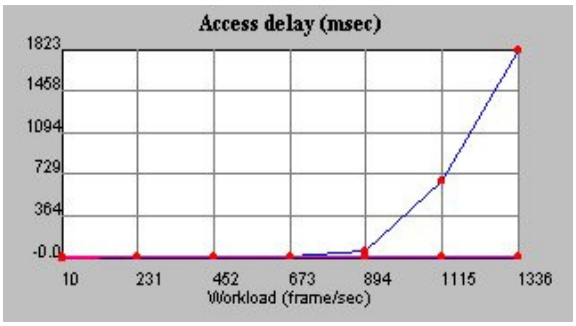
- I modelli analitici sono imprecisi perché quasi sempre, ad eccezione del caso di sistemi molto semplici, occorre fare un gran numero di semplificazioni per poter ottenere una formulazione matematica trattabile del sistema. Tuttavia è una tecnica importante perché fornisce dei risultati generali sugli aspetti fondamentali del sistema che poi possono essere dettagliati con la simulazione.
- Le simulazioni sono più precise ma possono richiedere molto tempo perché, mentre il modello analitico alla fine è una formula matematica risolvibile in forma chiusa, il modello di simulazione è un software, in genere abbastanza complicato, che richiede un tempo notevole per la preparazione e fornisce dei risultati affidabili solo dopo un lungo tempo di simulazione necessario per far andare a regime il sistema stesso.
- Le misure possono non fornire risultati accurati a causa della unicità (non ripetibilità) di alcuni parametri. Mentre nel caso di un modello analitico o di un modello simulativo si possono fare delle assunzioni di tipo teorico sul carico (spesso si stabiliscono alcuni tipi di carico standard che vengono utilizzati ripetutamente per valutare le prestazioni), nel caso delle misure il carico del sistema è quello reale di quel momento (unico e non ripetibile) ed il comportamento del sistema potrebbe non essere quello generale ma potrebbe dipendere da questo carico specifico. Per tale motivo, quando un sistema viene valutato attraverso misure è bene collezionare una notevole quantità di dati in modo che i valori siano statisticamente significativi. Inoltre è bene ripetere più volte la misura nelle medesime condizioni operative e verificare eventuali discrepanze nei risultati.

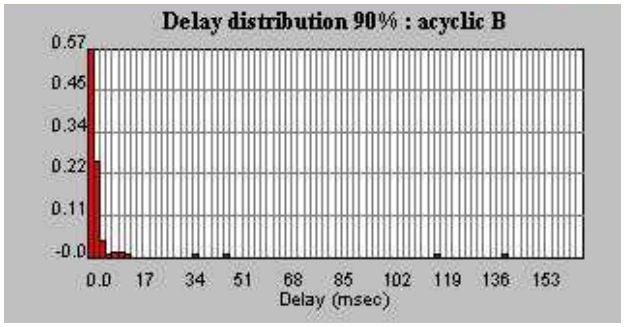
Un altro aspetto importante nella valutazione di un sistema è la correlazione tra i vari parametri sistema:

- I modelli analitici sono quelli più vantaggiosi da questo punto di vista perché nella formula matematica si vedono subito le relazioni tra i vari parametri, pertanto essi permettono di evidenziare l'effetto mutuo di più parametri in modo chiaro.
- Con le simulazioni, invece, a volte non è chiaro il trade-off fra i diversi parametri a meno che il progettista del software simulativo non lo metta espressamente in evidenza.
- Anche le misure rendono difficile interpretare il legame fra i vari parametri, non esistendo alcuna regola teorica in tal senso. Non è facile capire se una variazione nelle prestazioni dipende da un cambiamento casuale dell'ambiente o dal valore di qualche particolare parametro.

Una cosa molto interessante da osservare è che 2 o più tecniche possono essere usate in modo sequenziale: ad es. prima con un modello analitico si trova il range adatto dei parametri e poi con la simulazione si studiano le prestazioni del sistema in quel range. L'uso abbinato delle due tecniche risulta molto vantaggioso perchè un modello analitico semplificato (e quindi facile da realizzare) permette di determinare velocemente, le condizioni operative che mettono in risalto i comportamenti desiderati, che possono poi essere investigati in modo accurato attraverso la simulazione.

Per molte "metrics" il valore medio è quello importante nel senso che le prestazioni del sistema vengono di norma valutate in termini di comportamento medio a regime. Tuttavia non bisogna trascurare la variabilità tra i vari valori ottenuti che in alcuni casi può essere più pericolosa del valore stesso. Ad esempio, in una misura di tempo di ritardo fra i messaggi spediti in una rete per controllo di processo, il valore del tempo di ritardo medio è un'informazione insufficiente perchè se si ha un'elevata variabilità di valori, un basso valore medio non esclude che ci possano essere diversi valori singoli molto più elevati del valore medio stesso. Ciò può essere molto pericoloso in un ambiente operativo di questo tipo, dove i ritardi devono, in genere, essere limitati.





Questo concetto è chiarito nelle due figure mostrate sopra. Nella figura a sinistra è possibile vedere l'andamento del Delay medio di una rete (ethernet nell'esempio considerato) al variare del workload. Come si vede, per valori del workload fino a circa 700 frames al secondo, il ritardo è molto basso, quasi nullo. Tuttavia, la figura di destra che mostra la distribuzione percentuale di messaggi (ad un valore di workload pari a circa 700 frames al secondo) in funzione del loro ritardo, evidenzia come ci sia una percentuale (anche piccola) di messaggi con un ritardo molto maggiore degli altri. Questa informazione può essere estremamente utile nella valutazione dei sistemi per controllo di processo.

Quando la valutazione si riferisce ad un sistema distribuito, con diversi agenti, bisogna distinguere fra prestazioni individuali e collettive. Le prestazioni individuali descrivono il comportamento di un singolo agente e danno un'indicazione sul tipo di servizi che il singolo utente può aspettarsi dal sistema. Le prestazioni collettive descrivono invece il comportamento dell'intero sistema e danno indicazioni sulla tipologia di servizi globalmente offerti. Ad esempio, nello studiare una "Computer network" può essere utile analizzare separatamente il comportamento della rete da quello delle singole stazioni. Alcuni parametri (quale ad esempio la "fairness" sono strettamente legati a singole stazioni e dipendono dalla loro posizione, dal workload che offrono, ecc.)

Utilizzazione delle risorse, affidabilità e disponibilità sono "metrics" globali, mentre tempo di risposta e throughput possono essere visti come "metrics" sia globali che individuali.

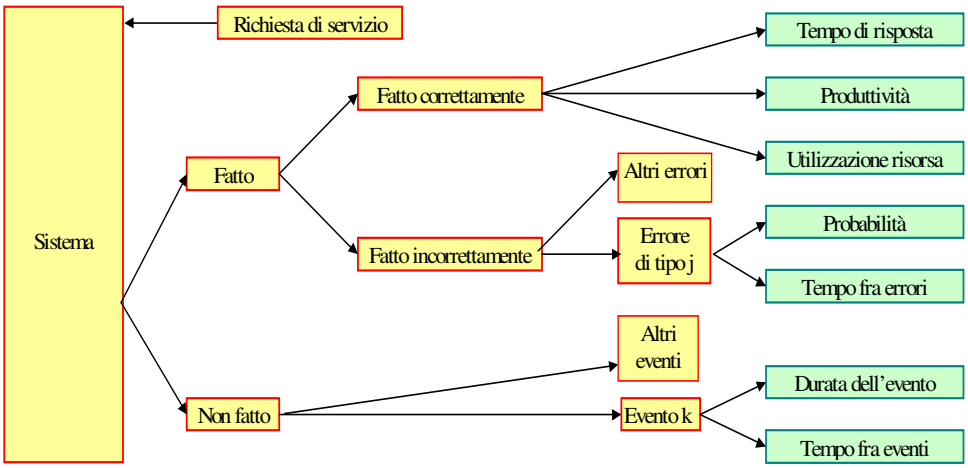
Nella scelta del tipo di "metrics" da usare per valutare un sistema è opportuno riferirsi ai seguenti criteri:

- "**Low variability**", cioè selezionare metrics che producono campioni caratterizzati da bassa variabilità. Ciò permette di ridurre il numero di misure che occorre fare per valutare il sistema, il che significa, nel caso di una simulazione, poter ridurre la durata della simulazione stessa.
- "**Non redundancy**" cioè selezionare metrics che contengono indicazioni diverse. Usare due metrics diverse per valutare lo stesso parametro serve solo a creare confusione.

"**Completeness**", cioè le metrics utilizzate devono essere sufficienti a definire il comportamento del sistema in modo completo. Ciò significa che di volta in volta può essere necessario inventare delle metrics specifiche, utili alla valutazione del particolare ambiente operativo.

3.2 Selezione degli indici prestazionali (performance metrics)

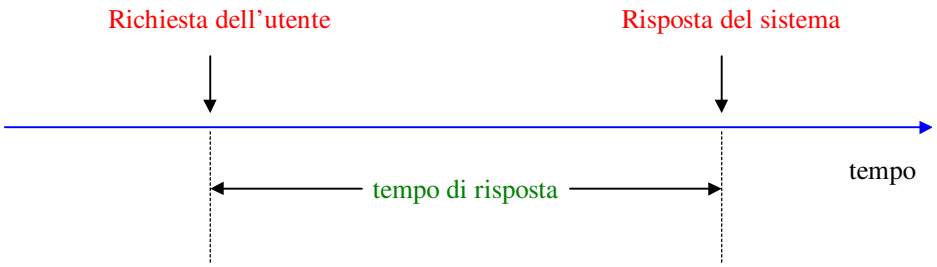
Gli indici prestazionali sono i valori di quei parametri che descrivono il comportamento del sistema. Un sistema può rispondere ad una richiesta di servizio in modo corretto, incorretto, o non rispondere affatto. Per ogni tipo di comportamento sono diversi gli indici prestazionali cui occorre fare riferimento. La figura sotto riportata mostra un sistema cui l'analista chiede di eseguire un servizio (rappresentato da un certo workload) e le diverse possibili risposte :



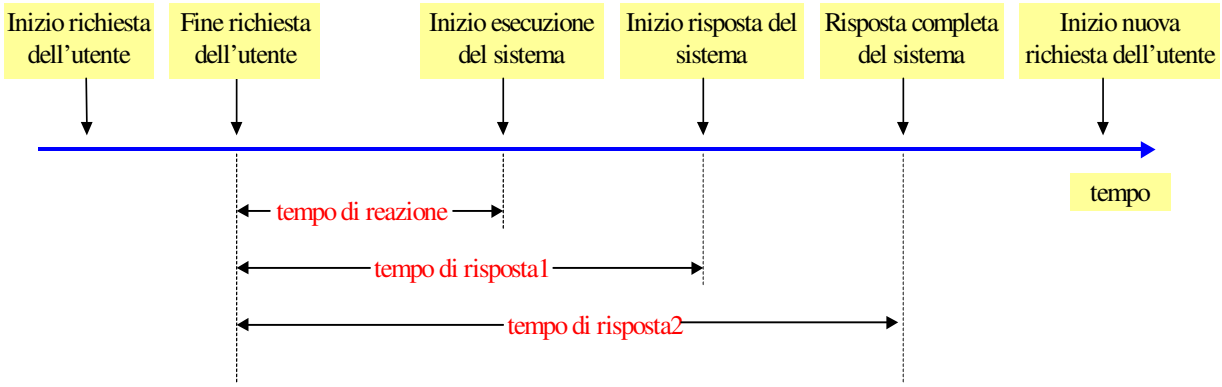
- se il sistema esegue il servizio correttamente allora le "metrics", cioè quei parametri che misuriamo come indici prestazionali del sistema, sono chiamate *responsiviness* (cioè tempo di risposta, tempo di attesa in coda, ecc.), *productivity* (cioè il throughput fornito), *utilization* (dà un'indicazione della percentuale di uso della risorsa, cioè misura l'efficienza nell'utilizzo delle risorse del sistema).Complessivamente definiscono la velocità (speed) del sistema.
- se il sistema non opera correttamente vuol dire che si è verificato un certo tipo d'errore per cui ciò che possiamo misurare e poi modellare è la probabilità che quel tipo d'errore si manifesti. Le metrics associate sono espresse in termini di affidabilità (reliability) del sistema.
- se il sistema non funziona (unavailable, cioè non disponibile) vuol dire che si è verificato un certo evento (ad es. un guasto) per cui si tratta di misurare la durata dell'evento e il tempo fra 2 eventi consecutivi della stessa classe, in questo modo possiamo classificare i *modelli di fallimento* (crash dell'intero sistema o solo di qualche suo nodo, oppure errori di omissione in trasmissione e/o in ricezione, oppure errori di falsi contatti sporadici) e determinare la probabilità di ciascuna classe d'evento. Il tempo medio tra 2 guasti è un indice molto importante per misurare l'affidabilità di un sistema.

3.3 Performance metrics più usate

- **Response time** (tempo di risposta): è l'intervallo di tempo fra la richiesta dell'utente e la risposta del sistema. È costituito da diverse componenti che possono essere misurate individualmente o globalmente. E' importante mettere in evidenza quali sono le componenti del sistema che intervengono nella definizione di un tempo di risposta tra 2 eventi ben precisi. Nel caso dei sistemi di automazione industriale, il response time può avere una importanza cruciale nella valutazione della bontà di un protocollo di comunicazione. Infatti, nei sistemi campionati (che sono una larga fetta dei sistemi di automazione) le variabili in gioco, trasmesse attraverso la rete vanno consegnate entro un prefissato timeout. Risulta pertanto importante controllare se le variabili vengono consegnate in tempo e, in caso negativo, misurare la percentuale di variabili che sono consegnate in ritardo.

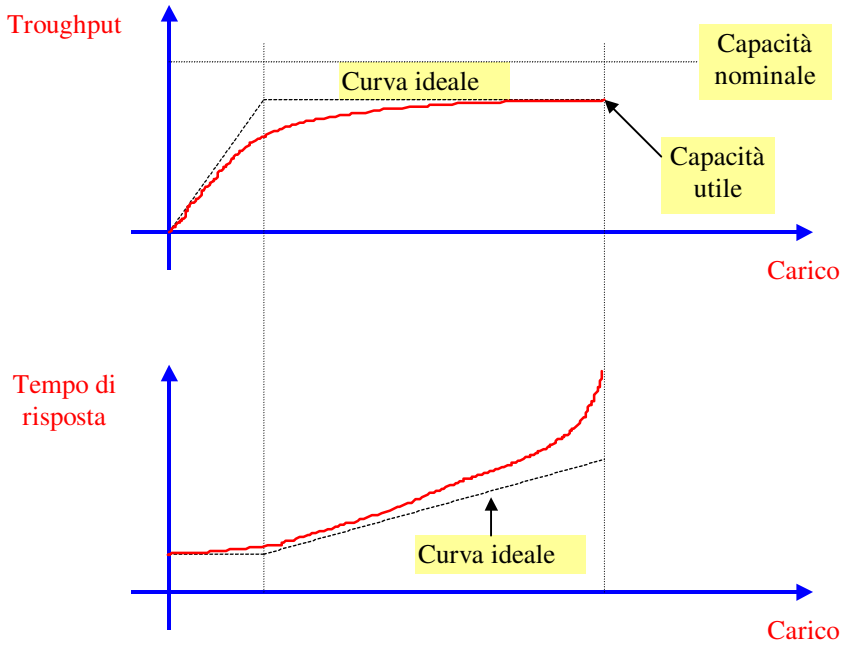


- **Reaction time** (tempo di reazione): è l'intervallo di tempo fra la sottomissione di una richiesta e l'inizio della sua esecuzione. E' una componente importante del tempo di risposta; può essere influenzata dalla presenza di un sistema operativo nel nodo o dal comportamento del particolare protocollo di comunicazione.



- **Stretch factor** (fattore di sforzo): è il rapporto fra il response time ad un certo carico e quello a minimo carico. Questo è un parametro molto importante perché permette di evidenziare il peggioramento della prestazioni del sistema rispetto al massimo che il sistema può fornire.
- **Throughput** (produttività):è la frequenza a cui le richieste possono essere servite dal sistema. Nel caso delle reti di calcolatori, le richieste sono rappresentate da messaggi (frames o pacchetti) mentre il servizio consiste nella loro trasmissione, per cui il throughput viene rappresentato in termini di messaggi trasmessi per unità di tempo. Se volessimo valutare il throughput di un sistema in termini di bit d'informazione trasmessi, allora tutta la parte header e trailer della frame deve essere scartata. Pertanto nella figura il workload sarà espresso in frames/sec (carico effettivo della rete) mentre il throughput sarà espresso in bytes/sec.
- **Nominal capacity** (capacità nominale): è il massimo throughput in condizioni ideali. Nel caso delle reti essa coincide con la "bandwidth", cioè con la larghezza di banda del canale trasmissivo. Il throughput non può quindi essere **mai** superiore alla larghezza di Banda del canale. Ad es. se una rete trasmette con un bit-rate di 1Mbps al massimo, nel caso ideale, essa potrà avere un throughput di 1Mbps; poi in realtà la capacità nominale del canale sarà inferiore ad 1Mbps perché occorrerà considerare : gli intervalli tra le varie frames, gli errori di trasmissione, i vari header e trailer per ogni frame che non costituiscono informazione utile, ecc. Ciò deve essere sempre ben chiaro in mente ed utilizzato come riferimento per non commettere errori di valutazione.
- **Usable capacity** (capacità utile): è il massimo throughput ottenibile senza superare un delay prefissato. In genere nelle reti la curva reale del throughput tende asintoticamente a quella ideale al crescere del carico, in quanto solo per un carico maggiore della massima capacità del canale

quest'ultimo viene sfruttato al massimo (se il protocollo di accesso è ben progettato). Però in corrispondenza i tempi di ritardo crescono anch'essi a dismisura; pertanto di norma per stabilire la capacità utile si fissa un tempo di ritardo massimo ammissibile ed in corrispondenza si legge il massimo throughput ottenibile.



- **Efficienza**: è il rapporto fra l'usable capacity e il nominal capacity, cioè è il rapporto tra il massimo throughput ottenibile e quello ideale. Questa efficienza si lega all'efficienza globale del sistema; supponiamo ad es. di avere 2 sistemi confrontabili costituiti da 2 protocolli simili: uno che utilizza frames con pochi bits di header e l'altro che invece utilizza frames con parecchi bytes di header, ovviamente questo secondo protocollo anche se fosse più affidabile nella consegna dei frames sarebbe ugualmente meno efficiente del primo perché la quantità di informazione che riuscirebbe a trasmettere sarebbe sempre molto minore del primo.
- **Utilizzazione**: è la frazione di tempo in cui una risorsa è impegnata per servire una richiesta. E' quindi misurabile come rapporto fra il tempo in cui la risorsa è occupata (busy time) ed il tempo totale considerato. Il tempo in cui una risorsa non viene utilizzata si chiama "idle time". Nella progettazione di un sistema è importante ottenere un bilanciamento del carico che permette di rendere massima l'utilizzazione delle risorse.
- **Reliability** (affidabilità): è misurabile in termini di probabilità d'errore o di tempo medio fra gli errori.
- **Availability** (disponibilità): è la frazione del tempo totale in cui un sistema è disponibile per le richieste dei vari utenti. Il tempo in cui il sistema non è disponibile è chiamato "down time". Il tempo in cui il sistema è disponibile è chiamato "uptime". Il valore medio dell' "uptime" anche chiamato "mean time to failure" è un ottimo indicatore della disponibilità e dell'affidabilità di un sistema.

4. Workload

4.1 Caratterizzazione del workload

Il workload consiste di richieste di servizi oppure di utilizzo di risorse da parte degli utenti di un sistema. Poiché l'ambiente vero dell'utente (real-user environment) non è in genere ripetibile, per realizzare il giusto workload è necessario studiare tale ambiente, osservarne le caratteristiche chiave e costruirne un modello da potere usare ripetutamente.
Ad esempio vogliamo valutare la capacità di una rete Ethernet di tipo shared, per trasmissione dati, per quanto riguarda il traffico vocale, cioè vogliamo vedere se è possibile una tale comunicazione telefonica su una rete ad accesso casuale (in questo caso infatti se la rete è congestionata si hanno molte collisioni e tempi di accesso possono diventare troppo lunghi per una conversazione vocale in cui le informazioni devono essere trasmesse con continuità per brevi intervalli di tempo): dopo aver realizzato la conversione analogico/digitale della conversazione sarà effettuato un campionamento a blocchi della voce per ottenere pacchetti contenenti più campioni, e saranno spediti in rete insieme agli altri pacchetti contenenti i dati. Per valutare il comportamento della rete Ethernet dobbiamo a questo punto caratterizzare il workload della rete: da una parte considereremo un modello di carico dati caratterizzato tipicamente da una distribuzione esponenziale, e dall'altro un modello di carico vocale caratterizzato da raffiche di pacchetti d'informazione successivi seguiti da lunghe pause.
I parametri che caratterizzano il workload (valori delle richieste d'utente, delle richieste d'uso di risorse, ecc.) devono dipendere dal workload e non dal sistema. Quelle caratteristiche che hanno un impatto significativo sulle prestazioni del sistema vanno incluse fra i parametri del workload; nel nostro es. mentre potrebbe essere ininfluente il fatto che la voce sia maschile o femminile, potrebbe essere importante il fatto che si parli in italiano o in inglese nella durata delle pause tra 2 burst. Ci sono 4 modi in genere per caratterizzare un workload:

- il modo più semplice è quello di utilizzare la *richiesta più frequente*, cioè poiché si fanno richieste di tipo differente al sistema noi possiamo pensare di usare un workload che è semplicemente costituito soltanto dalla richiesta più frequente. Nel caso di un sistema di automazione possiamo modellare il workload ipotizzando che tutte le variabili siano campionate alla stessa frequenza.
- oppure utilizzare un *miscuglio di richieste a frequenza diversa*, cioè possiamo rappresentare il traffico campionando le variabili a frequenze diverse, in accordo alla dinamica dei processi controllati. In questo caso bisogna generare un workload misto che tenga conto di un po' di tutto;
- oppure, è possibile utilizzare un *Trace di richieste di un sistema reale*, cioè utilizzare come workload una registrazione degli eventi reali che avvengono nel sistema in un certo intervallo di tempo arbitrariamente lungo. Il problema principale del trace è che la sequenza degli eventi registrati, per quanto lunga essa sia, è sempre troppo limitata rispetto alla grande quantità di memoria necessaria per contenerla.
- L'ultimo approccio che possiamo utilizzare è quello di fare *richieste mediate di servizi* con una prefissata probabilità di distribuzione nel tempo: tipicamente distribuzioni esponenziali o gaussiane.

Per *rappresentatività del workload* s'intende che il test workload deve essere rappresentativo del real workload. Questa rappresentatività deve essere verificata sotto 3 aspetti:

- *Arrival rate (frequenza d'arrivo) delle richieste*, cioè la frequenza di generazione delle richieste deve essere confrontabile con il real workload, ovvero devono avere la stessa distribuzione;

- *Resource demands* (richieste di risorsa) relativamente a ciascuna delle risorse chiave, cioè le richieste delle risorse chiave devono essere uguali sia nel syntetic workload che nel real workload;
- *Resource usage profile (profilo di uso delle risorse)*, cioè il nostro workload deve essere fatto in modo tale da sfruttare le risorse del sistema secondo un certo profilo di uso che deve essere uguale a quello del sistema reale (in un computer: stesso tempo d'uso della memoria centrale, del disco, ecc.).

Il workload permette di caricare il sistema con un certo numero di attività; il modo in cui l'attività viene assegnata al sistema permette di dedurre informazioni sul sistema stesso.

4.2 Tipi di workload

Il workload è quindi un elemento essenziale del nostro sistema ed occorre definirlo in modo che sia il più possibile rappresentativo di quello reale.

- Periodico- Nei sistemi campionati il traffico è soprattutto costituito da campioni generati in maniera periodica. Il workload più rappresentativo sarà quindi costituito da pacchetti (in genere di piccola dimensione perché i dati trasportati si riferiscono ad una sola variabile o al più a poche variabili) generati con una frequenza costante, pari a quella di campionamento. Il caso più semplice è quello in cui tutti i pacchetti hanno la stessa frequenza di generazione poiché l'intero processo controllato attraverso la rete ha una dinamica costante. In casi più complessi possono convivere processi con diverse dinamiche che richiedono frequenze di campionamento diverse. Pertanto il traffico sarà costituito da pacchetti che hanno diverse frequenze di generazione.
- Asincrono- Questo tipo di workload è legato a fenomeni non periodici (tipicamente allarmi) o eventi sporadici. In tal caso i pacchetti dati saranno normalmente di piccola dimensione, ma la frequenza di generazione non sarà periodica bensì sporadica. La generazione dl traffico sarà effettuata mediante una opportuna funzione random. Questo tipo di traffico di solito coesiste con quello periodico (ma ovviamente è molto minore) e va dimensionato in funzione delle condizioni operative che si vogliono rappresentare. Il traffico asincrono può anche essere legato alla generazione di messaggi di errore, ack, o ad operazioni legate ad interventi umani quali ad esempio l'aggiornamento di un DataBase. In questo ultimo caso il traffico potrà presentarsi in forma di Burst di pacchetti, anche di notevole dimensione (se occorre trasferire dei file). Esistono diverse distribuzioni con cui è possibile generare numeri casuali che si usano per modellare eventi asincroni. Le più note sono:
- **Distribuzione di Bernoulli**: una variabile può assumere solo i valori X=0, X=1 che determinano fallimento o successo.

  P = probabilità di successo, 1-P = probabilità di fallimento.

  La distribuzione di Bernoulli modella il verificarsi o meno di un evento. La distribuzione di Bernoulli viene utilizzata, ad esempio, per modellare la probabilità che un pacchetto una volta trasmesso venga rovinato dal rumore.
- **Distribuzione Binomiale**: Il numero di successi X in una sequenza di n tentativi di Bernoulli ha una distribuzione binomiale, adatta a modellare il numero di successi in una sequenza di n tentativi indipendenti. Ad esempio il numero di processori attivi in un sistema multiprocessore, il numero di bit disturbati in una frame, ecc.
- **Distribuzione esponenziale**: questa distribuzione è molto importante perché è quella che si utilizza per generare il carico in una generica rete dati. È la sola distribuzione continua

memoryless, questo significa che ogni campione che noi generiamo non ha memoria del campione precedente. Viene usata per modellare il tempo fra due eventi successivi indipendenti (tipicamente per la generazione del traffico in una rete di calcolatori).

5. Valutazione di prestazioni mediante Simulazione

La simulazione è utile sopratutto quando il sistema da valutare non è disponibile. Attraverso la simulazione è possibile confrontare alternative diverse con diversi workload ed ambienti. Una volta realizzato il modello del sistema da simulare, è possibile, cambiando il workload e quindi le condizioni al contorno, eseguire diverse valutazioni sullo stesso sistema. Ciò rappresenta un grosso vantaggio della simulazione, che la rende preferibile, nell'investigare le performances di un sistema, anche disponendo del sistema reale. Si pensi ad esempio ad una rete wireless in cui si vuole valutare il comportamento di un nodo, al variare del numero dei nodi, dell'entità di disturbi e della dimensione fisica della rete. Realizzare l'ambiente desiderato, in un sistema reale è molto complesso (a volte impossibile), costoso e richiede molto tempo. Con un simulatore, implementare le diverse condizioni al contorno richiede solo la configurazione di un certo numero di parametri.

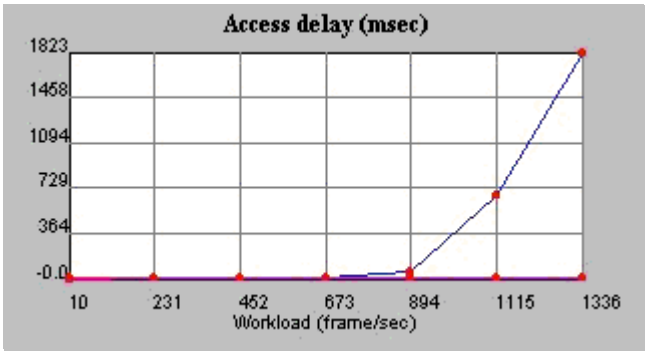
Va comunque ricordato quello che è il grosso rischio della simulazione: *l'errore è sempre in agguato*. Occorre sempre verificare con molta attenzione il modello realizzato e *non fidarsi mai ciecamente dei risultati ottenuti.*

Un altro dei vantaggi che abbiamo utilizzando una simulazione è che, alla fine della stessa, è possibile ottenere delle curve che forniscono una rappresentazione visiva dell'oggetto della simulazione (rappresentano ad esempio il tempo di ritardo di un sistema, il tempo di trasmissione, il tempo di attesa in coda, etc).

Una distinzione importante è relativa al tipo di valutazione di prestazioni che si intende effettuare: in Condizioni Stazionarie o in Transitorio. I due tipi di simulazione permettono di evidenziare aspetti diversi del sistema, che possono essere più o meno interessanti a seconda del tipo di applicazione. Va sottolineato come la stessa distinzione può essere fatta anche per le misure su un sistema reale, ma in questo caso la valutazione in Transitorio può essere difficile da realizzare per cui normalmente si opta per la valutazione in condizioni stazionarie.

5.1 Valutazione in condizioni stazionarie.

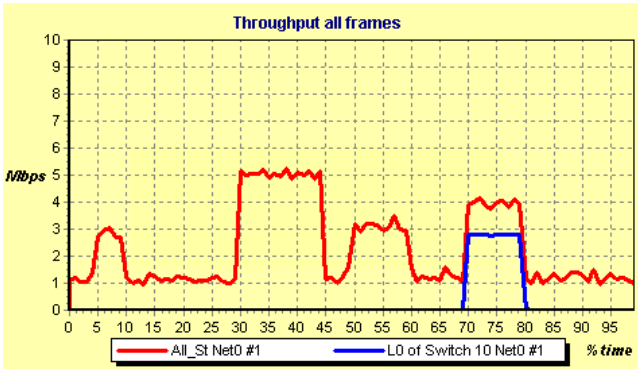
Questo tipo di valutazione va effettuata quando il sistema è a regime e permette di ricavare il valore stazionario di alcuni parametri di interesse.
Supponiamo di voler ottenere una curva che rappresenta il ritardo di trasmissione al variare del carico: a tal scopo è necessario eseguire, attraverso il simulatore, una serie di misure. Queste vengono effettuate variando il workload; per ogni workload otteniamo un certo ritardo (Delay). Interpolando i punti ottenuti in un grafico possiamo tracciare la curva. A titolo di esempio, la figura mostra un possibile risultato.

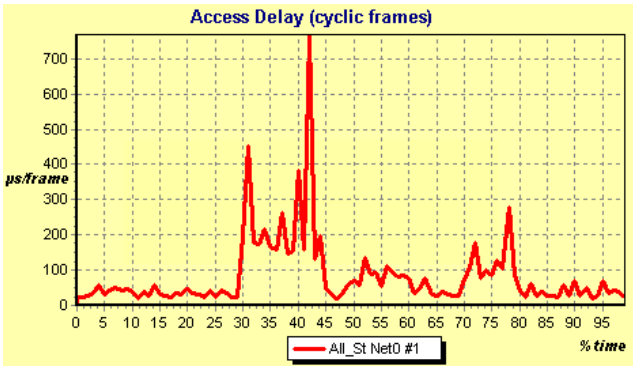


Va sottolineato come ogni singolo punto della curva non è, ovviamente, il risultato di una singola misura, bensì il risultato di una intera simulazione, è cioè una statistica. Ritornando all'esempio considerato, per ogni valore del workload il tempo di ritardo è misurato calcolando la media dei tempi di ritardo di un certo numero di pacchetti (10.000 - 100.000) spediti durante la simulazione.
E' importante avere già a priori un'idea del risultato che si vuole ottenere. Quando si esaminano i risultati ottenuti attraverso la simulazione, occorre sempre ricordare che i risultati ottenuti possono essere errati, per cui vanno sempre analizzati con spirito critico. E' pericoloso accettare per fede i risultati di una simulazione, specie se non riusciamo a giustificarli logicamente.
È facile commettere errori, (il simulatore può funzionare bene, ad esempio, per piccoli carichi e generare risultati scorretti quando il sistema va in saturazione, oppure può essere errato il modello di simulazione). Per evitare gli errori, o ridurli, occorre avere una certa preparazione per ciò che riguarda statistica e capacità nello sviluppo di software, oltre ad una approfondita conoscenza del problema da valutare.

5.2 Valutazione in Transitorio.

Questo tipo di valutazioni, a differenza della precedente, non si effettua quando il sistema è a regime, anzi al contrario, vogliamo mettere in evidenza il comportamento al transitorio.





Per comprendere bene la differenza fra i due tipi di simulazione osserviamo alcuni particolari delle figure.

- Nella valutazione a regime, la curva è ottenuta interpolando dei punti, ognuno dei quali rappresenta una statistica. Nella valutazione al Transitorio la curva è ottenuta collegando tutti i punti ottenuti, ciascuno dei quali rappresenta una singola misura (e non una statistica).
- Nella valutazione a regime l'asse delle ascisse rappresenta valori crescenti di un certo parametro (il workload nella figura di sopra). Nella valutazione al transitorio l'asse delle ascisse rappresenta lo scorrere del tempo, rappresentato come una percentuale della durata dell'intera simulazione.
- Nella valutazione a regime la curva evidenzia come varia un parametro al variare del valore di un altro parametro (il delay in funzione del workload nell'esempio considerato). Nella valutazione al transitorio viene evidenziato come varia nel tempo il valore di un parametro a causa di una brusca variazione di un altro parametro. Nell'esempio considerato si vede come varia il delay a causa di alcuni Burst di workload che hanno causato bruschi aumenti del Throughput.

5.3 Analisi degli errori nella simulazione.

Uno dei pericoli nella valutazione di prestazioni è la possibilità di commettere errori. Vediamo quali possono essere le principali cause:

- **Livello di dettaglio inappropriato**: quando si esegue una simulazione bisogna prima avere chiaro cosa si vuole ottenere da questa, e solo in un secondo tempo farne il modello. Un modello permette di eseguire alcuni tipi di misure, ma non tutte, quindi è necessario scegliere un modello opportuno di volta in volta. Nel caso delle reti, ad esempio, dopo aver studiato a fondo il protocollo di comunicazione, occorre prima decidere che tipi di problemi vogliamo analizzare, poi capire che tipo di misure dobbiamo fare, ed infine generiamo il modello adatto. Il modello può essere molto semplice se vogliamo ricavare delle valutazioni abbastanza generali, mentre se, ad esempio, vogliamo evidenziare il comportamento di una stazione che genera un certo tipo di traffico, rispetto ad una rete che ha un traffico di tipo diverso, è necessario creare un modello più complesso che tiene conto delle particolarità delle stazione rispetto alla rete (ad esempio una stazione che generare traffico di tipo vocale in una rete con traffico dati). Stabilire il livello di dettaglio della valutazione è fondamentale perchè influenza moltissimo il lavoro da svolgere.

- **Linguaggio inadatto**: possono essere usati linguaggi di tipo "general purpouse" (ad es. C, C++, Java) oppure linguaggi specifici (negli anni ne sono stati sviluppati diversi) o ambienti di simulazione quali OmNet++ o NS2-3. I linguaggi general purpouse permettono di rappresentare dettagliatamente tutti gli aspetti del problema in esame, però non hanno al loro interno delle strutture già predisposte per la simulazione. Ciò significa che il programma di simulazione dovrà essere costruito per intero, partendo dalle istruzioni offerte dal linguaggio prescelto. I linguaggi di simulazione invece hanno delle strutture predisposte per la simulazione, però rispetto ai linguaggi general purpouse sono meno flessibili. Pertanto, se spesso semplificano il lavoro, quando si devono simulare sistemi con caratteristiche particolari, possono presentare limitazioni che richiedono la messa a punto di modelli complicati e poco efficienti . L'uso di ambienti di simulazione predefiniti è oggi la strada più usata. Mettono a disposizione dei moduli che, opportunamente assemblati, permettono di realizzare velocemente il modello. Fra l'altro esistono diverse librerie che modellano i protocolli più usati e semplificano molto il lavoro.

- **Modello non valido**: se il modello non descrive correttamente il funzionamento del sistema è chiaro che la valutazione darà risultati inesatti. È molto importante allora una fase preliminare di studio. Se, ad esempio, vogliamo eseguire una simulazione su un protocollo di tipo CSMA/CA, è necessario innanzi tutto studiare il protocollo di accesso al mezzo fisico; non tutto il protocollo, ma almeno la parte relativa ai punti che stiamo investigando attraverso la simulazione.
- **Condizioni iniziali errate**: quando si manda in esecuzione una simulazione occorre inserire delle condizioni iniziali per definire lo stato del sistema. Un eventuale errore comporterebbe dei risultati errati pur partendo da un modello corretto.
- **Tempo di simulazione troppo breve**: se vogliamo simulare il comportamento di un sistema a regime, e le condizioni operative sono variabili, come normalmente avviene, i singoli valori che si ottengono non sono rappresentativi del sistema. E' necessario che la simulazione duri un tempo sufficientemente lungo per acquisire un numero di campioni sufficientemente elevato, tale da permettere di effettuare una statistica significativa (diciamo almeno qualche migliaio) . Nel caso di una rete di calcolatori, per esempio, alcuni secondi possono costituire un buon valore di tempo di simulazione, se in tale intervallo di tempo girano migliaia o centinaia di migliaia di messaggi. Per la simulazione di sistemi lenti, invece il tempo di simulazione può dover essere anche molto più lungo.

6.Tipi di simulazione

Sono tre i tipi di simulazione che vengono utilizzati:

6.1. Montecarlo Simulation

La simulazione Montecarlo è una simulazione statica, manca infatti l'asse dei tempi. Viene usata per analizzare dei fenomeni di tipo probabilistico e tempo invarianti. È utile nel caso si debbano analizzare sistemi che sono descrivibili analiticamente.

6.2 Trace Driven Simulation

In questo tipo di simulazione usiamo come ingresso una *trace* di eventi di un sistema reale per ottenere una *trace* di uscita. Una *trace* è un file che descrive una serie di dati.

- Vantaggi: credibilità, workload accurato, facilità di confronto, somiglianza con implementazioni reali. Il workload che viene fornito in ingresso è molto accurato perché è descritto campione per campione; ciò permette di eseguire delle simulazioni diverse utilizzando sempre lo stesso trace e quindi di confrontare sistemi diversi nelle identiche condizioni operative.
- Svantaggi: finitezza del campione, eccessivo dettaglio (sia del campione che del modello).

6.3 Discrete Event Simulation

Si implementa un modello a stati discreti del sistema, cioè un modello in cui lo stato del sistema evolve a scatti anziché in modo continuo. In questo modo possiamo far avanzare lo stato del sistema in funzione degli eventi che ci interessa mettere in risalto. (Se vogliamo, ad esempio, analizzare una rete di calcolatori; ogni pacchetto che viene generato rappresenta un evento che influenza il sistema, il quale evolve attraverso un trigger continuo da parte di questi eventi, che sono tempo discreti).
Il tempo cui riferiamo la simulazione può essere sia un tempo continuo che un tempo discreto. Normalmente viene utilizzato un tempo discreto, perché utilizzare un tempo continuo richiederebbe un livello di granularità nelle operazioni che modellano l'evoluzione del sistema troppo fine, che consuma tempo di calcolo inutilmente. Descritto il sistema, possiamo, ad esempio, fare un analisi in cui la risoluzione del tempo sia di un microsecondo: si fa avanzare il tempo di un microsecondo per volta e si osserva ciò che succede nel sistema. In questo modo di operare il tempo viene considerato continuo con risoluzione di un microsecondo; ciò significa che ogni microsecondo occorre verificare lo stato di tutte le variabili sotto osservazione (e ciò richiede tempo) anche se queste variabili non sono cambiate rispetto al microsecondo precedente. Se invece si fa avanzare il tempo a scatti, incrementandolo della differenza di tempo fra due eventi successivi, allora la simulazione viene eseguita più velocemente e la misura che si ottiene è ugualmente rigorosa perché non si perdono eventi e non si approssimano i tempi. In questo caso, il tempo viene considerato discreto. Nel primo caso per ogni avanzamento del tempo si controlla se si è verificato un qualche evento, nel secondo ogni volta che si verifica un evento si incrementa il tempo di una certa quantità.
La Discrete Event Simulation è quella più usata nella pratica e viene adottata sia in OMnet++ che in NS2-3

6.4 OMnet++

Omnet++ è un ambiente object-oriented per simulazioni ad eventi discreti quali possono essere:

- Reti di comunicazione wired e/o wireless;
- Modellazione di protocolli;
- Modellazione di reti di code;

ARCHITETTURA

Omnet++ fornisce l'infrastruttura e tools per realizzare simulatori. Uno degli ingredienti fondamentali di questa infrastruttura è l'architettura a componenti per i modelli di simulazione.

I modelli di simulazione vengono realizzati tramite l'impiego di componenti riusabili chiamati moduli. Questi moduli possono essere combinati come dei blocchi LEGO.

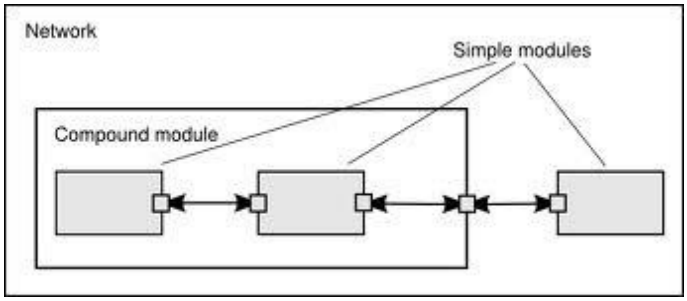
I moduli possono essere connessi tra di loro attraverso i gates e combinati insieme per formare dei moduli composti (compound modules). Non c'è un limite sulla profondità di nesting dei moduli.

La comunicazione tra moduli normalmente avviene tramite message passing e i messaggi possono contenere delle strutture dati arbitrarie (a parte informazioni predefinite tipo i timestamp). Questi messaggi possono viaggiare attraverso percorsi predefiniti di gates e connections oppure essere inviati direttamente alla loro destinazione (quest'ultima scelta è molto utile nel caso dell comunicazioni wireless). Inoltre un modulo può essere corredato di parameters per personalizzarne il comportamento e per renderlo parametrico.

Omnet++ permette di eseguire simulazioni attraverso interfaccia grafica (per dimostrazioni e debug) oppure attraverso interfaccia command-line (migliore per esecuzione batch). I simple modules possono essere raggruppati in compound modules.

ELEMENTI PRINCIPALI

Un modello di simulazione (simulation model) omnet++ consiste in moduli che comunicano attraverso message passing. I moduli che racchiudono la vera e propria logica si chiamano simple modules, essi sono scritti in C++ attraverso delle librerie di supporto messe a disposizione da omnet++.I simple modules possono essere raggruppati in compound modules senza alcun limite in livelli di gerarchia. L'intero modello, chiamato network, è esso stesso un compound module. Nella figura sotto, i rettangoli grigi sono i simple modules mentre quelli bianchi sono compound modules, si può anche notare che vi sono dei quadratini che rappresentano i gates (d'ingresso e/o d'uscita) e le frecce che sono le connections



Come detto in precedenza i moduli comunicano attraverso messaggi, i quali possono attraversare vari gates oppure essere inviati direttamente al modulo destinazione. I gates quindi sono le interfacce di input e di output dei moduli: i messaggi sono spediti attraverso un output gates e arrivano agli input gates. Un input gate e un output gate possono essere collegati da una connection (o link); ogni connection è creata all'interno di un singolo livello della gerarchia di un modulo: all'interno di un compound module si possono connettere i gates di 2 simple module o un gate di un simple module e uno del compound module stesso.

Connection che interessano diversi livelli di gerarchia non sono possibili perché ostacolerebbero la riutilizzabilità. A causa della natura gerarchica del modello, i messaggi tipicamente viaggiano attraverso una catena di connections partendo da e arrivando a simple modules. Alle connections possono essere assegnati dei parametri come:

- Propagation delay;
- Data rate;

- Bit error rate.

È anche possibile definire delle connections con proprietà specifiche. I moduli possono avere dei parameters, usati principalmente per passare dati di configurazione ai simple modules e per aiutare a definire la topologia del modello di simulazione. I parameters possono essere stringhe, valori numerici o booleani. Oltre all'essere usati come costanti, i parameters possono essere anche sorgenti di numeri random secondo una data distribuzione. La libreria C++ di omnet++ mette a disposizione delle classi per modellare:

- Moduli, gates, connections, parameters;
- Messagges, packets (i packets sono un'estensione dei messages usati per le reti di comunicazione);
- Containers (array, code);
- Data collection classes;
- Statistiche e distribuzioni (istogrammi, etc...);

THE NED LANGUAGE

Il linguaggio NED (Network Description) è usato per realizzare una descrizione modulare di una network, di un simple module o compound module.Una file NED può contenere I seguenti componenti in quantità arbitraria:

- Direttive di import (meccanismo Java-like)
- Definizione di channels (usati per realizzare le connection)
- Definizioni di simple and compound module
- Definizione di interfacce (sono descrizioni astratte di moduli, quelli concreti "estenderanno" l'interfaccia)
- Definizioni di network Per quanto riguarda gli identificatori (usati per dare un nome a qualsiasi componente) sono ammesse lettere (a-z,A-Z), numeri (0-9) e il carattere underscore; un identificatore non può iniziare con un numero,

7. Analisi dei risultati

Durante lo sviluppo di un modello di simulazione occorre garantire che esso sia rappresentativo del sistema reale (validazione del modello) e che esso sia implementato in modo corretto (verifica del modello). La validazione del modello è legata alla correttezza delle assunzioni fatte sul comportamento del sistema (cioè al fatto che abbiamo modellato il sistema in modo corretto o il nostro modello contiene qualche errore). La verifica (debugging sul modello implementato) è relativa alla correttezza della sua implementazione, cioè al SW.

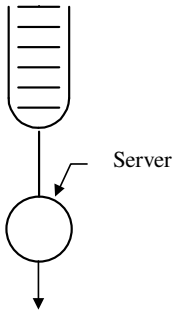
7.1. Tecniche di validazione del modello

La validazione intende assicurare che le assunzioni usate per il modello siano ragionevoli e che se correttamente implementate forniscano risultati simili a quelli del sistema reale. Tre aspetti chiave vanno validati:

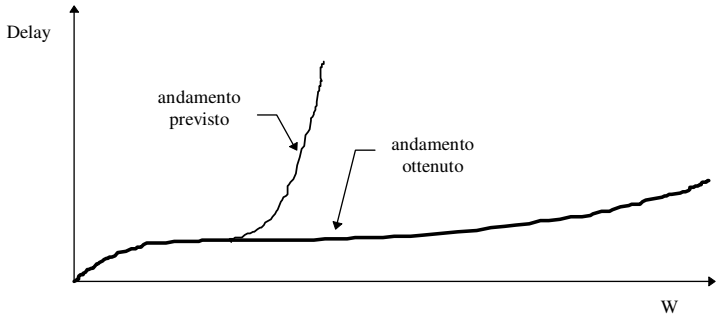
- **Assunzioni**: dato un sistema occorre capire come funziona il sistema, cioè capire quali sono le caratteristiche del sistema che devono essere implementate, trascurando quelle caratteristiche che sono irrilevanti ai fini di ciò che si vuole simulare (modello semplificato del protocollo).

- **Valori dei parametri di ingresso e distribuzioni**: dobbiamo cioè valutare le caratteristiche del sistema che si sta simulando con determinati valori del workload. Occorre, in altre parole, fornire al sistema, un workload significativo in riferimento alle reali applicazioni del sistema stesso. Per far questo può essere utile effettuare inizialmente dei calcoli grossolani che consentono di stabilire il range adatto del workload. La distribuzione del carico deve essere descrittiva del tipo di sistema che si sta valutando (a seconda se stiamo simulando un sistema per il controllo di processo un sistema per trasmissione di dati, etc). Se, ad esempio, la rete è costituita da dieci nodi e lavora a 1 Mbit/s, nell'ipotesi che i messaggi siano di lunghezza fissa 1000 bit il workload che manda in saturazione il sistema può essere calcolato approssimativamente in 1000 messaggi al secondo. Poichè i nodi sono dieci, per mandare in saturazione il canale ogni nodo deve spedire 100 messaggi (stiamo supponendo un carico distribuito in maniera uniforme). Quindi 100 messaggi al secondo, in media, portano verso la saturazione del canale, valori molto più bassi implicano che il canale è scarico o debolmente caricato, valori più alti significano canale oltre la saturazione.
- **Valori d'uscita e conclusioni**. Alla fine della simulazione, occorre analizzare i valori di uscita e trarre da questi le giuste conclusioni. Nell'analizzare i risultati ottenuti, occorre ricordare che se stiamo effettuando una valutazione di prestazioni, in condizione stazionarie, occorre essere certi di trovarsi in tali condizioni prima di giustificare i risultati ottenuti. Un punto critico è il comportamento del sistema quando si è in condizioni di saturazione. In tale condizione, il sistema non riesce ad andare a regime, ed i risultati ottenuti non sono significativi o vanno analizzati con molta attenzione.

Approfondiamo questo punto che è particolarmente importante. Nel fare un modello di simulazione ogni nodo viene tipicamente schematizzato come una coda (o eventualmente più code se abbiamo diversi tipi di traffico o diverse priorità). Se il canale non è saturo normalmente il server ha un tempo di servizio che è più breve del tempo con cui arrivano i messaggi per cui, in tali condizioni, la coda è vuota o parzialmente piena. In queste condizioni è possibile calcolare correttamente i tempi di ritardo della rete.



Quando però forniamo al nodo un carico talmente elevato che il sistema va in saturazione, il tempo di servizio del server risulta più grande del tempo di arrivo dei messaggi. In questa condizione la coda si riempie. Questa è una condizione critica per la valutazione del sistema sia che si usi un simulatore, ma anche se si effettuano misure su un sistema reale. Infatti tutti i messaggi che vengono generati, essendo la coda già piena, vengono scartati (a meno che la coda non abbia lunghezza infinita). Questo fa si che i risultati ottenuti siano falsati. In tali condizioni si possono ottenere curve di ritardo con l'andamento mostrato in figura:



Come si vede, all'aumentare del workload, dapprima il delay si mantiene costante, poi quando si raggiunge la saturazione del canale trasmissivo ci si aspetta che il delay cresca rapidamente (andamento previsto). La curva che invece si ottiene, vede crescere il delay molto lentamente (andamento ottenuto). Questo comportamento è legato al modo in cui è calcolato il ritardo, facendo la differenza fra l'istante di generazione dei messaggi e quello di consegna a destinazione.

Poichè in condizioni di saturazione vengono trasmessi solo pochi messaggi (la maggior parte resta in coda), la statistica viene effettuata su pochi campioni, molti dei quali sono relativi al transitorio iniziale, quando la rete non era ancora congestionata. In tali condizioni il tempo medio di attesa in coda non tiene conto del fatto che ci sono code piene di messaggi che non si riescono a trasmettere.

Per mettere in evidenza quest'ultimo aspetto possiamo aumentare il tempo di simulazione lasciando invariato il carico, così facendo la statistica viene influenzata meno dai primi messaggi che in effetti fanno parte del transitorio iniziale. In condizioni di saturazione, poichè il calcolo del tempo di ritardo non è molto affidabile, conviene fare riferimento ad altri tipi di performance metrics, quali ad esempio il numero di elementi in coda alla fine della simulazione o il calcolo della pendenza con cui le code crescono nel tempo (sebbene anche questi parametri siano influenzati dalla durata della simulazione). Comunque, poiché vogliamo effettuare delle misure a regime mentre in saturazione il sistema si trova in regime transitorio (senza fine) i dati ottenuti vanno considerati solo qualitativi.

Per eseguire un ulteriore test sulla validità dei risultati ottenuti si può eseguire un confronto con:

- **Intuizione di esperti**: magari attraverso brainstorming meeting. Riunirsi attorno a un tavolo fra più colleghi e discutere i risultati può essere molto utile per evidenziare errori o aspetti particolari di un sistema.
- **Misura sui sistemi reali:** Il confronto fra i risultati ottenuti mediante simulazione e quelli forniti da un sistema reale nelle stesse condizioni, costituisce un efficace strumento per validare la simulazione stessa.
- **Risultati teorici:** Anche il confronto con un modello analitico, pur se semplificato, può costituire un utile strumento di validazione dei risultati ottenuti.

7.2 Tecniche di verifica del modello

Il secondo problema è relativo alla corretta implementazione del modello di simulazione. Due importanti tecniche per sviluppare, fare il debug, e mantenere programmi di simulazione sono:

- **Modular design**: richiede che il modello sia strutturato in moduli che comunicano fra loro attraverso opportune interfacce. Questo permette di modificare parte del modello senza intervenire su tutto.
- **Top-down design**: consiste nello sviluppare una struttura gerarchica del modello in modo che il problema sia ricorsivamente diviso in un set di sottoproblemi.

Altre tecniche di verifica utili a individuare malfunzionamenti nel modello sono:

- **Antibugging**: consiste nell'includere checks ed outputs addizionali nel simulatore, che evidenziano gli errori. Stampare periodicamente la lunghezza di una coda di trasmissione e di ricezione permette ad esempio di capire se il sistema sta evolvendo nel modo previsto. Oppure leggere i tempi di generazione e di consegna di un messaggio può essere utile per trovare il punto dove il messaggio resta in attesa per un tempo troppo lungo.
- **Modelli deterministici**: usando parametri deterministici è facile fare un debug del modello che andrà poi eseguito con i parametri corretti. Se generiamo un carico fisso deterministico possiamo calcolare in maniera deterministica tutte le grandezze che ci interessano, confrontando tali grandezze con quelle ottenute attraverso la simulazione verifichiamo la correttezza del modello.
- **Eseguire dei casi semplificati:** E' il primo tentativo da effettuare per capire se il sistema funziona correttamente. Se ad esempio consideriamo l'ambiente senza disturbi e con un solo nodo, ci aspettiamo che il trasmettitore possa trasmettere liberamente alla massima velocità , non ci siano collisioni e le code trasmissive siano vuote. Se così non è occorre capire cosa sta succedendo. Il secondo passo è considerare una rete con due soli nodi: un sender ed un receiver: anche in questo caso ci aspettiamo che la comunicazione avvenga senza problemi, e così via. Se si trovano problemi in casi molto semplici è facile trovare l'errore. In uno scenario complesso le cause di errore possono essere tante e non è facile capire dove si è sbagliato.
- **Continuity test**: forti variazioni dell'uscita per piccole variazioni degli ingressi, normalmente sono sospette e vanno osservate con molta attenzione.

8. Misure sui sistemi reali

Effettuare delle misure su sistemi reali è ovviamente più difficile che valutare un sistema mediante la simulazione. E' vero che operando su un sistema fisico non c'è bisogno di realizzarne il modello, (con i relativi possibili errori) ma è anche vero che nascono problemi che in un sistema simulato non sono presenti. Elenchiamone alcuni e vediamo come risolverli:

- Misura dei tempi
- Misura del Throughput
- Definizione dell'ambiente operativo
- Valutazione del Physical layer e del mezzo fisico.

8.1 Misura dei tempi

In un sistema simulato si fa riferimento ad un unico tempo (il clock del simulatore) ed è facile confrontare i tempi in cui si verificano gli eventi in un sistema distribuito. Ad esempio la misura del ritardo di comunicazione di un pacchetto richiede la misura dell'istante di generazione del pacchetto, della messa in coda, della trasmissione e della ricezione nel nodo remoto. In un simulatore, poiché abbiamo un tempo unico di riferimento il calcolo è effettuato senza problemi. Fra l'altro, essendo un sistema virtuale, la lettura delle grandezze di interesse non introduce tempi di ritardo come invece avviene nei sistemi reali.

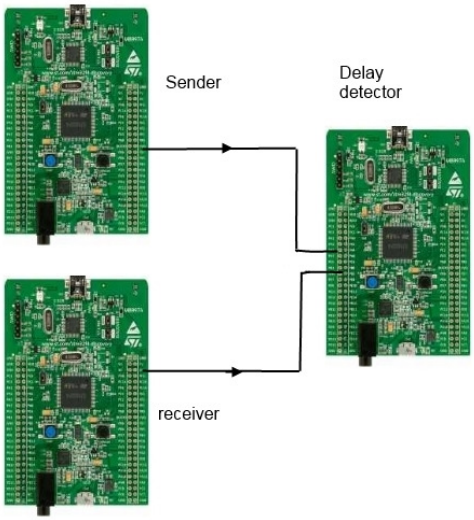
In un sistema reale, poiché i clock dei vari nodi sono fra di loro scorrelati non possiamo confrontare le letture effettuate nei vari nodi. Una possibile soluzione è sincronizzare i clock nei

vari nodi ma questa operazione è pesante (consuma banda e può interferire con la misura) e non sempre garantisce la precisione richiesta.

Inoltre, anche la lettura dei tempi è più complessa: mentre in un simulatore il tempo è una variabile di simulazione, disponibile per la lettura, in un sistema reale i tempi sono contenuti in opportuni timer o contatori e la loro lettura richiede tempo e ciò può introdurre degli errori. Le due soluzioni più semplici sono:

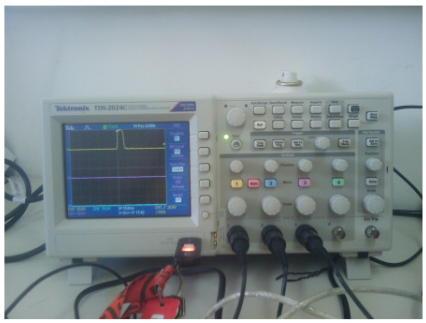
- Misurare il tempo di andata e ritorno (esempio invio di un pacchetto e ricezione dell'Ack). Poiché la lettura avviene su un unico nodo non c'è problema con il clock. Il limite di questo approccio è che valuta una intera transazione e non le singole trasmissioni per cui non è possibile capire quanto tempo impiega il pacchetto e quanto l'ack di conferma: la misura è globale. Inoltre, questo approccio non riesce a valutare il ritardo introdotto dal software nei nodi fra la ricezione del pacchetto e la trasmissione del relativo ack.

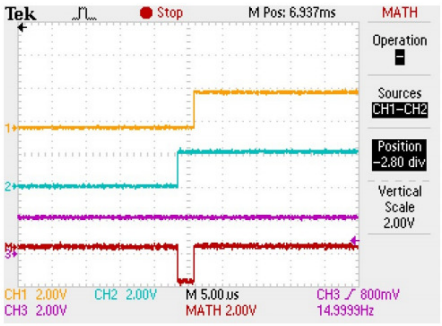
- Misurare il tempo impiegato da una singola trasmissione, mediante l'ausilio di un terzo dispositivo. Questo approccio è adatto alle applicazioni con microcontrollori o in ogni caso con quei dispositivi in cui si può avere una interazione diretta con l'Hardware.



Consideriamo la configurazione a lato in cui un Sender invia un pacchetto ad un Receiver (abbiamo omesso di rappresentare i due transceiver per la connessione) ed un Delay detector misura il ritardo tra gli eventi. Nel momento in cui nel Sender inizia l'evento di cui si vuole misurare la durata (generazione di un pacchetto, suo inserimento in coda, ecc.) viene messo alto (o Basso) un piedino della sua porta GPIO collegato all'ingresso di un altro microcontrollore (Delay Detector, DD). Questo evento genera nel DD un interrupt che fa partire un contatore Hw che misura la durata dell'evento.

Appena il Receiver riceve il pacchetto, mette alto (o basso a seconda dei casi) un piedino, che genera nel DD un interrupt che blocca il contatore. A questo punto il contatore contiene la misura esatta del ritardo fra i due eventi.





Una variante a questo approccio è basata sull'uso, al posto del Delay detector, di un oscilloscopio digitale mediante il quale è possibile misurare il ritardo fra gli eventi legati al set della porta GPIO selezionata nel sender e nel receiver. Nell'esempio presentato, la linea azzurra mostra il set della porta nel sender (inizio trasmissione) mentre la linea gialla mostra il set della porta nel receiver. La linea rossa misura il ritardo fra le due. Con un oscilloscopio di qualità è possibile effettuare misure molto precise, fino a pochi nanosecondi.

C'è da considerare comunque che l'operazione di SET del piedino della GPIO e la gestione del conseguente interrupt nel DD introduce un ritardo che teoricamente dovrebbe produrre un errore

nella misura. Va però sottolineato che poiché lo stesso ritardo viene introdotto sia all'avviamento del contatore che al suo arresto, questi due ritardi dovrebbero teoricamente compensarsi.

8.2 Misura del Throughput

La misura del Throughput è in genere abbastanza semplice. Basta realizzare nel sender un modulo software che generi traffico secondo le modalità previste dal sistema (dimensione dei pacchetti, frequenza di generazione, ecc.) e contare nel receiver il numero di pacchetti ricevuti correttamente. Il Throughput può essere calcolato contando il numero di pacchetti moltiplicati per la loro dimensione considerando l'intero pacchetto o solo il payload a seconda dei casi.

Se il protocollo di comunicazione prevede l'uso automatico dei retry, può essere utile contarli (se l'HW lo consente) in modo da avere una statistica sul numero medio di ritrasmissioni effettuate.

Una misura seria del Throughput deve tenere conto di uno scenario realistico (in cui sono presenti diversi sender) in modo da tenere conto di possibili collisioni o della necessità di effettuare una schedulazione delle trasmissioni da parte di un Master. Uno scenario formato solo da un sender ed un receiver è adatto a calcolare il Max throughput che il sistema può fornire in condizioni ideali.

8.3 Definizione dell'ambiente operativo.

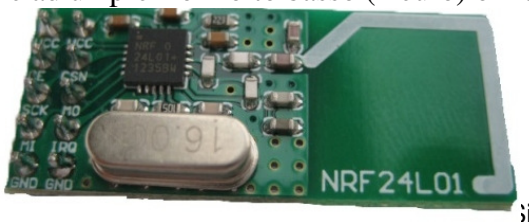
L'ambiente operativo può essere estremamente variabile a seconda dell'applicazione considerata. Le differenze possono essere considerate da diversi punti di vista e per ogni tipo di applicazione occorre definire chiaramente le caratteristiche peculiari ed i parametri di interesse. Le caratteristiche più interessanti possono riguardare:

- Tipologie di traffico: percentuale di traffico periodico ed asincrono.
- Densità dei nodi e valori del workload
- Rumore ambientale
- Mobilità dei nodi (ad es.applicazioni di vehicle to Vehicle V2V communication)
- Presenza di ostacoli metallici (tipicamente in ambienti industriali)

8.4. Valutazione del Physical layer e del mezzo fisico

La struttura del physical layer e le caratteristiche del mezzo fisico influenzano molto le prestazioni del sistema. Ricordiamo che, in accordo al modello a livelli, le prestazioni dell'applicazione sono influenzate da quelle di tutti i livelli sottostanti. Il physical layer è il livello più basso e mette a disposizione un canale digitale con un certo bit rate. Più elevato è il bit rate offerto dal livello Phy maggiore è la banda disponibile per i livelli superiori. E' pertanto essenziale che il PHY layer metta a disposizione un canale affidabile con la maggior banda possibile dato che la larghezza di banda disponibile all'applicazione sarà solo una frazione di quella offerta dal PHY layer.

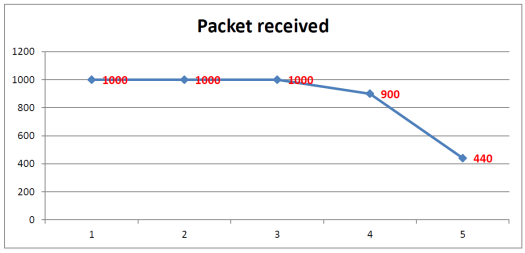
Nel seguito faremo riferimento solo ai sistemi wireless dato che i sistemi wired in genere introducono pochi problemi rispetto alle prestazioni generali (hanno banda elevata e sono parecchio affidabili) e forniremo alcuni suggerimenti sulle misure che è utile effettuare. Consideriamo un generico transceiver wireless (in particolare faremo riferimento al transceiver basato sul NRF24L01 che ad un prezzo molto basso (1 euro) offre prestazioni molto interessanti.)



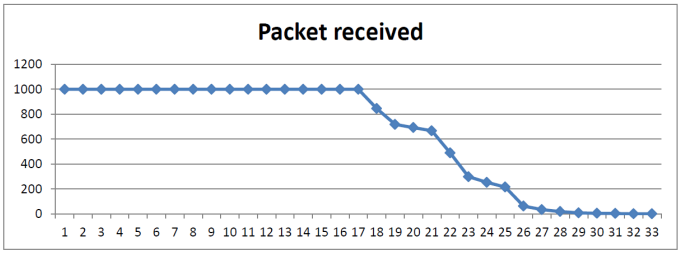
L' nRF24L01+ è un ricetrasmettitore wireless a chip singolo a 2,4 GHz con embedded baseband protocol engine (Enhanced ShockBurst ™), quest' ultimo è

basato sulla comunicazione a pacchetti e fornisce varie modalità di configurazione con operazioni di gestione del protocollo avanzate. In effetti il transceiver implementa funzionalità che non sono solo del Physical layer ma implementa anche un semplice MAC che rappresenta la base per progettare protocollo più sofisticati. Il front-end radio utilizza la modulazione GFSK e i suoi parametri sono configurabili dall'utente (canale di frequenza, potenza di uscita e la velocità di trasmissione dati), ad esempio il data rate può essere impostato a 250kbps, 1Mbps o 2Mbps. Inoltre la possibilità di effettuare il frequency Hopping su oltre 100 canali, permette di implementare sofisticati protocolli per la gestione del mezzo.
La principale valutazione da effettuare è quella relativa alla distanza coperta sia indoor che in open air. E' una misura molto importante perché influenza la topologia della rete nel senso che in base alla distanza coperta si può optare per una rete single-hop o multi-hop. Inoltre ha un impatto anche sulla potenza consumata che, nelle WSN, è un parametro molto importante.
A tale scopo, si prepara un file che genera un certo numero di pacchetti (1000 è sufficiente) e si effettuano diversi test trasmissivi, a distanze crescenti, sia indoor che in open air. I risultati ottenuti vengono rappresentati in opportune tabelle come quella di figura.

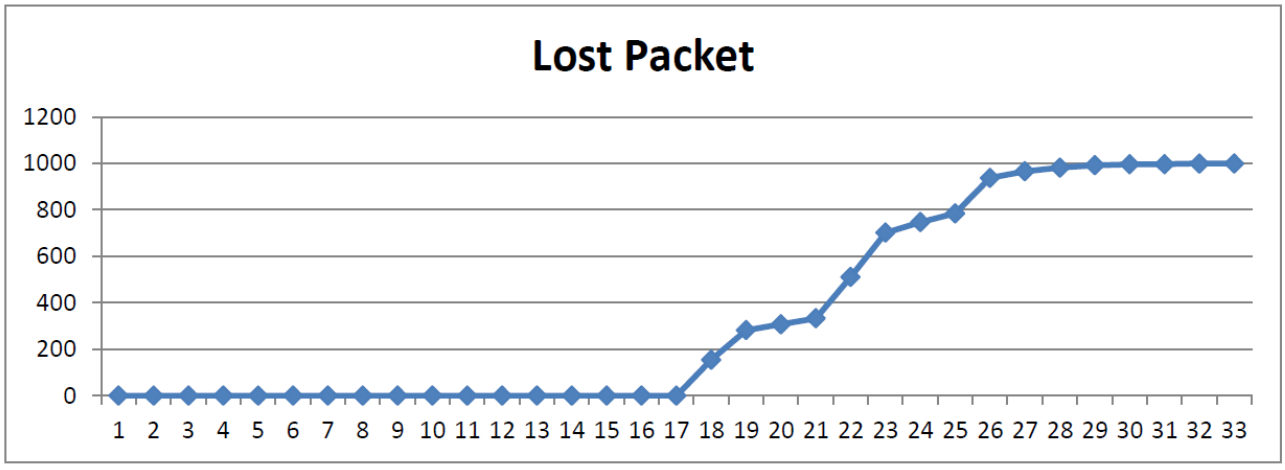
Test interno, ambiente domestico



Test esterno, extraurbano senza ostacoli



Come è da aspettarsi in ambiente indoor la distanza coperta è molto inferiore rispetto all'ambiente outdoor, senza ostacoli (sull'asse delle ascisse è riportata la distanza coperta in metri).



Oltre al throughput può essere interessante valutare il numero di pacchetti persi al variare della distanza. Nella figura sopra è mostrato il caso outdoor,con un bit rate di 2 MBPS, in ambiente senza disturbi. E' importante sottolineare che, nelle curve ottenute, il sistema è stato configurato senza possibilità di retry in caso di errore, quindi i valori ottenuti si riferiscono a singole trasmissioni. E' anche possibile configurare il sistema in modo da ripetere la trasmissione N volte (il valore N è configurato dall'utente) in modo da valutare il max Throughput o il packet error rate in tali condizioni.

Le misure possono essere ripetute per diversi valori del bit rate (per il dispositivo considerato varia da 250Kbps a 2 Mbps) poiché a bit rate più bassi vengono in genere coperte distanze maggiori a parità di potenza consumata e quindi è un parametro da valutare accuratamente in funzione delle esigenze temporali del traffico trasportato.

Le misure possono essere ripetute a diverse frequenze (visto che il dispositivo permette il frequency hopping) per vedere se alcune frequenze sono più favorevoli di altre, ed in presenza di stazioni interferenti, su canali vicini. A questo proposito, può essere utile valutare l'interferenza prodotta da una stazione al variare della distanza del canale e/o della distanza fisica rispetto alla trasmittente. Si tratta di misure lunghe, a volte noiose, che però permettono di caratterizzare in modo molto accurato il comportamento del transceiver.

Completiamo questa sezione con una nota molto interessante. Abbiamo visto come la simulazione (ad esempio mediante OmNet++) permetta di valutare sistemi anche di grandi dimensione (cosa difficile da fare con le misure) andando a sfruculiare anche parametri interni difficili da misurare (il delay è uno di questi). Il punto debole di un simulatore è di solito il modello del Physical layer poiché il suo comportamento è influenzato da molti parametri alcuni dei quali sono difficili da tenere in conto (ad es. le caratteristiche fisiche dell'ambiente o la presenza di elementi di disturbo od ostacoli). Per gli standard wireless più noti e consolidati esistono già dei modelli che sono stati sviluppati per il Physical layer e vengono usati normalmente nella simulazione, ma sono comunque modelli teorici, approssimati, che in condizioni operative particolari possono fallire. Per tale motivo può essere utile un approccio integrato fra simulazione e misura. E' possibile generare dei trace di traffico predisponendo un certo numero di nodi nelle condizioni simili a quelle dell'ambiente che si vuole valutare e poi usare tali trace al posto del modello del physical layer nel simulatore. In pratica, ogni volta che viene effettuata una trasmissione, il simulatore preleva un elemento del trace ed acquisisce il suo stato (trasmissione con successo/fallimento) passando il risultato al DLL. In tal modo, il comportamento del Physical layer non dipende da un modello matematico teorico, ma è il risultato di misure reali effettuate nelle stesse condizioni previste dal modello.

9. Due ulteriori problemi: il transitorio iniziale e la durata della valutazione.

Dopo lo sviluppo del modello occorre decidere quante osservazioni iniziali devono essere scaricate prima che il modello sia in uno stato stazionario, e quanto tempo debba durare una simulazione. Questi due aspetti vanno considerati solo per le valutazioni a regime. Invece, nelle valutazioni in condizioni transitorie, poiché si visualizza il comportamento di un sistema nel tempo, le osservazioni iniziali vengono visualizzate chiaramente nei risultati, per cui l'utente può eliminarle senza problemi. Lo stesso si dica per la durata della simulazione in quanto non facciamo delle statistiche ma osserviamo il comportamento di un sistema al transitorio: la durata della simulazione (o misura) dipenderà da quanto è lungo il transitorio che vogliamo analizzare.
Questi due punti sono invece cruciali nell'esecuzione di una simulazione o di una misura in condizioni stazionarie: infatti, durante il transitorio iniziale, prima che il sistema si porti a regime, si ottengono dati che in genere non sono significativi poichè le condizioni operative non sono stabili. E' pertanto indispensabile capire quando il transitorio è terminato. E' inoltre importante comprendere quanto tempo debba durare la simulazione, in modo da collezionare una quantità adeguata di dati, da analizzare.
Ovvero, bisogna affrontare i seguenti due problemi:

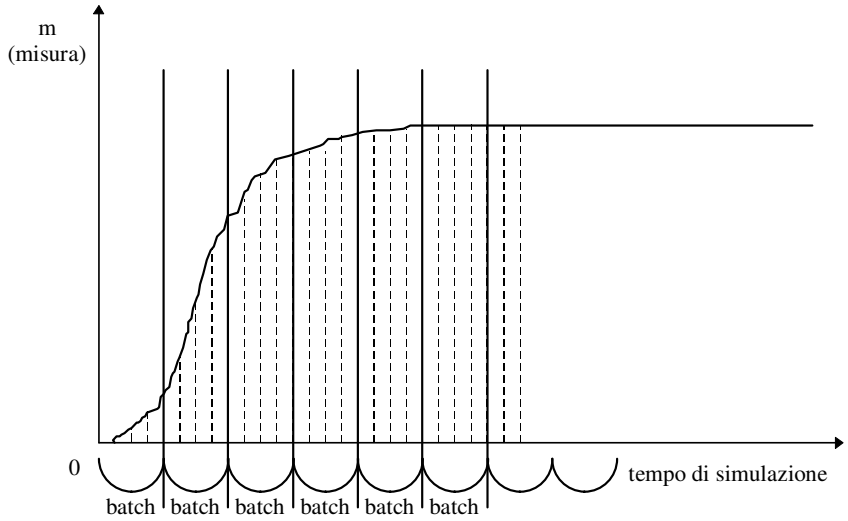
- **Transient removal**: rimozione del transitorio iniziale
- **Stopping criterion**: scegliere i criteri per bloccare la simulazione.

9.1 Eliminazione del transitorio iniziale

Nella maggior parte dei casi in una simulazione interessa solo la *Steady-state performance* (Performance nello stato stazionario). Pertanto i risultati della parte iniziale della simulazione (transitorio iniziale) vanno rimossi. Le strategie che possono essere utilizzate sono le seguenti:

- **Long runs**: la simulazione dura talmente a lungo che le condizioni iniziali non influenzano i risultati. È la tecnica più semplice, però presenta due svantaggi: 1) spreco di risorse nel senso che, per mettersi in condizioni di sicurezza spesso si fa durare la simulazione più a lungo del tempo strettamente necessario; 2) non è sicuro che il run sia abbastanza lungo (caso in cui il workload è basso per cui il numero di pacchetti spediti in tutta la simulazione è basso), per cui può capitare che i valori collezionati nel transitorio iniziale influenzino pesantemente i risultati.
- **Proper initialization**: consiste nel fare iniziare la simulazione in uno stato vicino a quello stazionario. Ciò naturalmente riduce il transitorio. Il problema è determinare lo stato vicino a quello stazionario, da cui fare iniziare la simulazione. Difficile da usare.
- **Initial data deletion**: durante la fase stazionaria, la media dei valori ottenuti non cambia molto, anche eliminando alcuni campioni. Si eliminano valori iniziali fin quando la loro eliminazione provoca variazioni sensibili nella media.

- **Batch means**: una lunga simulazione è divisa in parti (batch) di uguali durata. Occorre studiare la varianza di questi batch in funzione della loro dimensione. Ciò permette di eliminare i batch relativi al transitorio iniziale, ed utilizzare nelle statistiche solo i dati relativi ai batch in condizioni stazionarie, mediando fra i vari batch.



Spesso la tecnica usata è la Long runs. Anche se come detto spreca risorse, è però facile da usare.

9.2 Arresto della simulazione

Per determinare il tempo di simulazione, cioè il tempo necessario affinché il sistema sia a regime, ed abbia prodotto un numero sufficienti di campioni per le statistiche non esistono regole precise. In genere basta eseguire due simulazioni alle stesse condizioni ma con tempi diversi; se i risultati ottenuti sono simili allora vuol dire che siamo a regime. A questo punto occorre contare il numero di campioni della grandezza di interesse (pacchetti spediti, numero di collisioni, ecc.) . Se il numero è sufficiente per una statistica significativa (almeno qualche centinaia/migliaia di campioni) si sceglie il tempo di simulazione inferiore fra i due.

10. Analisi dei sistemi da campioni

A questo punto, dopo aver diligentemente effettuato la nostra valutazione di prestazioni, tenendo conto di tutti i suggerimenti e le indicazioni viste nei precedenti capitoli, riteniamo di aver concluso il nostro lavoro. Abbiamo definito un modello, lo abbiamo simulato (o abbiamo effettuato delle misure su un sistema reale) abbiamo ottenuto dei risultati e li abbiamo commentati. Abbiamo finito no?

E invece manca ancora un ultimo lavoro da effettuare. Quest'ultima parte della dispensa sulle valutazioni di prestazioni affronta un tema non meno importante dei precedenti (forse il più importante) relativo alla bontà della valutazione effettuata nel caso di regimi stazionari. Per quanto riguarda lo studio dei sistemi in transitorio, i risultati ottenuti hanno un significato più qualitativo che quantitativo, per cui più che la precisione della valutazione interessa piuttosto il comportamento generale che viene messo in evidenza.

Concentriamoci quindi sulle valutazioni in sistemi a regime. Quando eseguiamo delle misure sui sistemi otteniamo delle misure puntiformi, nel senso che le valutazioni di prestazioni che noi eseguiamo non sono relative al comportamento del sistema nella sua globalità, ma a specifici punti di lavoro del sistema. Definite le condizioni al contorno, otteniamo una serie di valori che descrivono il comportamento del sistema in uno specifico punto di lavoro. Quindi in teoria abbiamo valutato completamente il sistema in questo punto di lavoro. Però ricordiamo che noi effettuiamo la

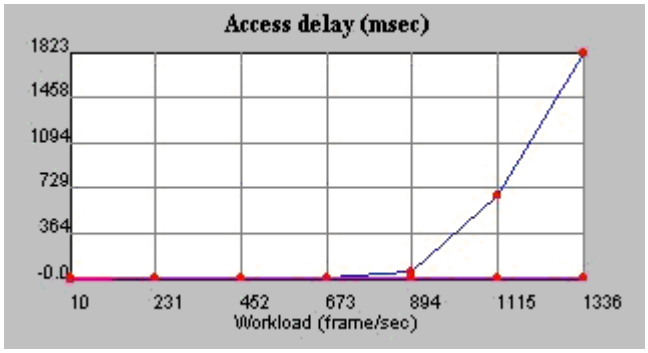
valutazione di prestazioni per un tempo limitato (che è solo una piccolissima frazione del tempo di vita del sistema) e otteniamo solo dei campioni che occorre quindi analizzare per comprendere quale significato attribuire ai valori trovati. Il problema che emerge con evidenza, è che risulta impossibile ricavare le caratteristiche esatte di un sistema basandosi solo sull'osservazione di alcuni campioni. Se, ad esempio, vogliamo valutare le caratteristiche di una rete, quello che facciamo è:

- Definire la struttura della rete
- Fornire un workload sotto forma di pacchetti dati;
- eseguire delle misure su questi pacchetti;

da queste misure cerchiamo di estrapolare il comportamento della rete (es. Throughput e delay).

Appaiono subito evidenti 2 considerazioni:

•Perchè la valutazione che stiamo effettuando sia significativa (nel senso che i dati siano in qualche modo interpetrabili) occorre definire delle condizioni al contorno, ben delimitate. Occorre cioè scegliere un workload ben identificabile (definito come frequenza media di arrivo dei messaggi, loro lunghezza media, probabilità di errore, ecc.), fissare un numero di nodi, la dimensione della rete, ecc. Ciò consente di ottenere una valutazione del sistema *in un ben preciso punto di lavoro*. Ribadiamo quindi che il risultato ottenuto non è una valutazione del sistema nella sua globalità, ma solo in un punto di lavoro. Si intuisce chiaramente come una valutazione esaustiva (o quantomeno, sufficientemente completa) di un sistema sia un'operazione lunga, che richiede diverse valutazioni puntiformi e consuma tempo. Consideriamo la figura a lato. L'Access delay è stato ricavato per diversi valori di workload. Per ogni valore di workload si è ottenuto un punto (cioè il valore che descrive il comportamento del sistema in uno specifico punto di lavoro). Raccordando vari punti si ottiene la curva che descrive il comportamento del sistema in un ampio range di condizioni operative.



•Ma, anche la valutazione esatta *di un singolo punto di lavoro* è un problema di non facile soluzione. Infatti, per determinare il comportamento esatto della rete occorrerebbero un numero infinito di campioni, il che ovviamente non è possibile visto che, considerata la finitezza della vita umana, siamo costretti a limitare la durata della nostra valutazione.

Quella che otteniamo è pertanto una rappresentazione, necessariamente approssimata, della realtà. Quindi, caduta l'illusione di poter valutare in modo esatto e completo le prestazioni del sistema considerato, accontentiamoci di determinare con quale probabilità le caratteristiche del sistema sono contenute in un certo range, cioè di capire, quando facciamo una valutazione di prestazioni, in quale intervallo hanno significato i risultati che abbiamo trovato. Più piccolo è tale range, migliore è la valutazione.

Riprendiamo l'esempio di prima, supponiamo di voler misurare il tempo medio che impiega un pacchetto per giungere a destinazione, qualunque sia l'approccio che stiamo usando, analisi, simulazione, etc, quello che facciamo è generare dei pacchetti e misurare il tempo di ritardo. Alla fine abbiamo un campione (X1,X2,...Xn), cioè **n** valori ricavati; indichiamo con **Xm** la media

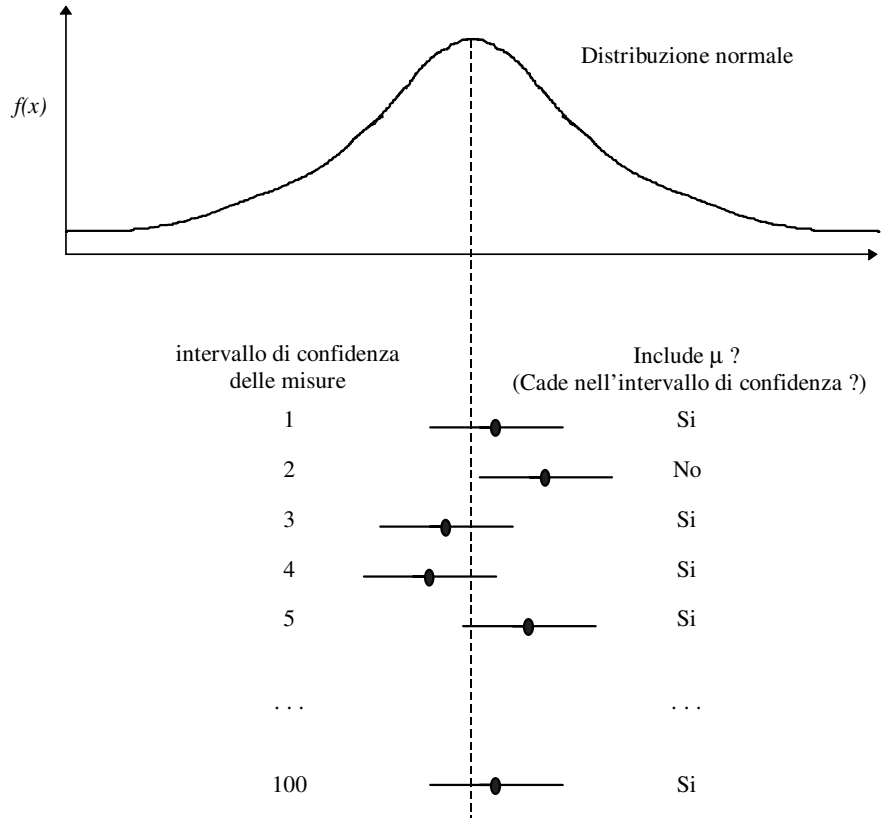
relativa al campione. Tale media è diversa da quella μ della popolazione di dati misurati (il valore μ si ottiene considerando un numero infinito di valori). Valgono le seguenti definizioni:

- **Xm** è il valore misurabile, definito *Statistica*
- μ è il valore cercato, definito *Parametro*

Xm è il valore che otteniamo su un certo numero, finito, di dati; se invece analizzassimo un numero infinito di dati otterremmo μ. Per un dato sistema, fissate le condizioni al contorno, i parametri sono fissi poichè dipendono dalle caratteristiche del sistema, mentre le statistiche sono variabili in modo casuale essendo influenzate da variabili aleatorie presenti in tutti i sistemi (ad esempio i disturbi o gli errori accidentali). Quella che possiamo ricavare noi è la media di un campione che rappresenta una stima della media della popolazione. Non è possibile ottenere una stima perfetta della media della popolazione partendo da campioni di dimensione finita. Quando studiamo un sistema (sia attraverso misure che attraverso simulazioni) otteniamo set di valori in numero finito. Il comportamento del sistema, che estrapoliamo attraverso l'analisi dei valori osservati, si avvicinerà sempre più al comportamento reale del sistema all'aumentare del numero di valori osservati (ecco il motivo per cui occorre fare durare la simulazione/misura un tempo sufficientemente lungo per acquisire un numero notevole di campioni).

La stima di un parametro di una popolazione, data da un solo numero (per esempio, il valore Xm) è definita **Stima Puntuale** del parametro stesso. Se la stima è data da due numeri che si possono considerare, uno maggiore ed uno minore del parametro reale, allora si parla di **Stima per Intervallo.** Esempio**:** se diciamo che la misura di un tempo di ritardo è 23 mSec stiamo fornendo una stima puntuale. Se invece diciamo che il ritardo è compreso fra 23 +/- 3 mSec cioè fra 20 mSec e 26mSec, allora stiamo fornendo una stima per intervallo. La stima per intervallo è l'approccio normalmente più corretto poiché a meno di casi particolari (ad esempio un sistema time-slotted centralizzato in cui i tempi di ritardo sono esattamente uguali) i valori che noi otteniamo sono sempre distribuiti in un certo intervallo. Più piccolo è l'intervallo, meglio è.

Un parametro molto importante è quello che viene chiamato *Intervallo di confidenza*. Questo è un intervallo (C1,C2) che con una elevata probabilità $\mathbf{P = 1- \alpha}$ contiene la media μ della popolazione (cioè il valore esatto). Ad esempio se per un tempo di ritardo l'intervallo di confidenza è compreso tra 3ms e 3.5ms con una probabilità del 90%, significa che su 100 diverse valutazioni, il 90% dei campioni che misuriamo cadono nell'intervallo (3ms, 3.5ms). *L'intervallo di confidenza garantisce quindi la qualità della valutazione che stiamo eseguendo.*



Il grafico fa riferimento alla distribuzione normale, quella che normalmente si ottiene quando si eseguono misure sui sistemi reali, in cui si osserva il valore medio μ (relativo al campione infinito) e gli intervalli di confidenza relativi a certi campioni. Come si vede alcuni di questi intervalli contengono il parametro μ, altri invece non lo contengono. Supposto di avere eseguito cento misure, il numero di quelli che cadono dentro sarà $\geq 100(1-\alpha)$, mentre quelli che cadono fuori saranno $\leq 100(\alpha)$. Quindi l'intervallo di confidenza fornisce un intervallo entro il quale è possibile dare significato alle misure effettuate.

- **$100(1-\alpha)$** che rappresenta la percentuale di confidenza è anche chiamata *livello di confidenza*
- **$(1-\alpha)$** si chiama *coefficiente di confidenza*

La qualità della misura è quindi legata al valore di α che indica la probabilità che i valori che noi misuriamo cadono dentro l'intervallo di confidenza. Per calcolare l'intervallo di confidenza non è necessario eseguire un gran numero di sequenze di misura, ma è possibile fare riferimento ad un singolo campione (X1,X2, ... Xn) purché le singole osservazioni siano indipendenti e provengano dalla stessa popolazione. Il fatto che le misure siano indipendenti è molto importante, ed è legato al modo con cui viene generato il workload . In particolare si deve considerare un workload che sia rappresentativo del carico reale e che sia costituito da campioni tra di loro scorrelati, cioè indipendenti, in modo da ottenere una distribuzione di tipo normale; se c'è una qualche correlazione tra i valori del workload in ingresso sicuramente ci sarà una correlazione tra i valori in uscita e la distribuzione che otteniamo non è una distribuzione normale. Sotto questa ipotesi, è possibile utilizzare il "teorema del limite centrale" il quale stabilisce che se le osservazioni in un campione (X1,X2, ... Xn) sono indipendenti e provengono dalla stessa popolazione che ha una media μ ed una deviazione standard σ, allora il valor medio di un campione sufficientemente ampio è distribuito in modo normale, con valore medio μ e deviazione standard $\sigma/\sqrt{n}$.

Segnalo un sito:

https://www.mccallum-layton.co.uk/tools/statistic-calculators/confidence-interval-for-mean-calculator/

che mette a disposizione un tool per il calcolo dell'intervallo di confidenza una volta nota la dimensione del campione, il valor medio, la deviazione standard ed il livello di confidenza desiderato.

3.1 Calculator

Enter Sample Size ? 12500

Enter Sample Observed Mean ? 33

Enter Sample Observed Standard Deviation ? 2

Select Desired Confidence Level (%) ? 95

Reset Calculate

3.1.1 Results

3.1.1.1 ***Confidence Interval: ±0.04***

3.1.1.2 ***Range for the true population mean: 32.96 to 33.04***

Un'altro tool simile è disponibile all'indirizzo: http://www.surveysystem.com/sscalc.htm