



Concetti di valutazione delle prestazioni ed introduzione alla simulazione

Roberto Petroccia
petroccia@di.uniroma1.it



Riferimenti

- Valutazione delle prestazioni di un sistema e Introduzione alla simulazione
 - R. Jain, "*The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*", Wiley- Interscience, New York, NY, April 1991.
Capitoli 2, 3, 24



Outline

- Valutazione delle prestazioni di un sistema
- Introduzione alla simulazione
- Verifica ed analisi dei risultati simulativi



Valutazione delle prestazioni

- Offrire le migliori prestazioni al minor costo
- Confrontare delle soluzioni alternative e trovare la migliore
- Capire quale sistema si comporta meglio per uno specifico insieme di applicazione (scenario)
- Utile in qualsiasi fase del ciclo di vita di un sistema (progettazione, realizzazione, utilizzo, upgrade)



Approccio sistematico alla valutazione delle prestazioni (1/3)

- **Specificare l'obiettivo e definire il sistema**
 - Definire i limiti del sistema, in base all'obiettivo dello studio
- **Elencare i servizi e i risultati**
 - Esempi:
 - rete permette di spedire pacchetti verso specifiche destinazioni
 - una base di dati risponde a query
 - Ogni servizio presenta un numero di possibili esiti
 - La lista dei servizi e degli esiti è utile per selezionare le metriche e il carico di lavoro
- **Selezionare le metriche**
 - Selezionare i criteri (metriche) per confrontare le prestazioni (es. velocità, accuratezza, disponibilità)
- **Elencare i parametri che influenzano le prestazioni**
 - Parametri di sistema (dipendono dall'hardware e il software del sistema)
 - Parametri di workload (dipendono dalle richieste dell'utente)
- **Individuare i fattori da studiare**
 - Alcuni parametri variano durante la valutazione (fattori) e assumono diversi valori (livelli)



Approccio sistematico alla valutazione delle prestazioni (2/3)

- **Selezionare una tecnica di valutazione**
 - Modello analitico
 - tecniche di valutazione delle prestazioni per catturare il comportamento dinamico del sistema (vari strumenti: teoria delle code, reti di code, processi di markov,...)
 - Ottimizzazione di problemi per determinare limiti e soluzioni ottime per un sistema
 - Simulazione
 - Strumento software che consente di riprodurre il funzionamento della rete con un unico programma, e studiarne il comportamento, i vari eventi che accadono nel tempo (e.g. trasmissione e ricezione dei pacchetti), seguendo nel dettaglio il comportamento dello stack protocollare
 - Test-bed reale
 - Esperienza diretta con i dispositivi di rete



Approccio sistematico alla valutazione delle prestazioni (3/3)

- **Selezionare il carico di lavoro**
 - Insieme di richieste di servizio al sistema, che rappresenti l'utilizzo reale del sistema
 - Esempi:
 - Modello analitico: probabilità di varie richieste
 - Simulazione: tracce di sistemi reali, o generatori di traffico (network simulation)
 - Script utenti eseguiti sul sistema
- **Progettare gli esperimenti**
 - Decidere una sequenza di esperimenti, in cui si variano fattori e livelli, per determinarne l'effetto sulle prestazioni
- **Analizzare e interpretare i dati**
 - L'analisi produce risultati, che possono essere diversi per ogni esperimento, e devono essere interpretati per arrivare a una conclusione
- **Presentare i risultati**
 - Facili da comprendere (es. grafici)
 - Spesso si deve tornare al primo punto (la valutazione consiste di diversi cicli)



Criteri per selezionare una tecnica

Criterion	Analytical		
	Modeling	Simulation	Measurement
1. Stage	Any	Any	Postprototype
2. Time required	Small	Medium	Varies
3. Tools	Analysts	Computer languages	Instrumentation
4. Accuracy ^a	Low	Moderate	Varies
5. Trade-off evaluation	Easy	Moderate	Difficult
6. Cost	Small	Medium	High
7. Saleability	Low	Medium	High

^a In all cases, result may be misleading or wrong.



Tre regole di validazione

- Non fidarsi dei risultati simulativi finchè non sono stati validati attraverso modelli analitici o misure di sistema
- Non fidarsi dei modelli analitici finchè non sono stati validati attraverso simulazioni o misure di sistema
- Non fidarsi di misure di sistema finchè non sono stati validati attraverso simulazioni o modelli analitici
- Il miglior modo di validare un sistema è in genere una combinazione delle diverse tecniche



Diverse nozioni di prestazione

Bisogna sapere come valutare le prestazioni di un sistema



Diverse nozioni di prestazione

Bisogna sapere come valutare le prestazioni di un sistema

Velivolo	DC to Paris	Velocita'	Passeggeri	Throughput (pmp)
Boeing 747	6.5 hours	610 mph	470	286,700
Concorde	3 hours	1350 mph	132	178,200

Quale dei due velivoli ha le migliori prestazioni?

- **Tempo per terminare il task (Execution Time)**
 - execution time, response time, **latency**
- **Numero di Task per giorno, ora, sec, ns,... (Performance)**
 - **throughput** (# di task per unita' di tempo), bandwidth

Response time e throughput sono spesso in contrapposizione



Diverse nozioni di prestazione

- **Tempo di volo del Concorde rispetto al Boeing 747 ?**
 - **Il Concorde è 1350 mph / 610 mph = 2.2 volte più veloce**
= 6.5 ore / 3 ore

Il Concorde è 2.2 volte (“120%”) più veloce in termini di tempo di volo



Diverse nozioni di prestazione

- **Tempo di volo del Concorde rispetto al Boeing 747 ?**
 - **Il Concorde è $1350 \text{ mph} / 610 \text{ mph} = 2.2$ volte più veloce**
= $6.5 \text{ ore} / 3 \text{ ore}$

Il Concorde è 2.2 volte (“120%”) più veloce in termini di tempo di volo

- **Throughput = beneficio per passeggero = velocità per passeggero (pmp)**

Boeing 747 = 286.700 pmp

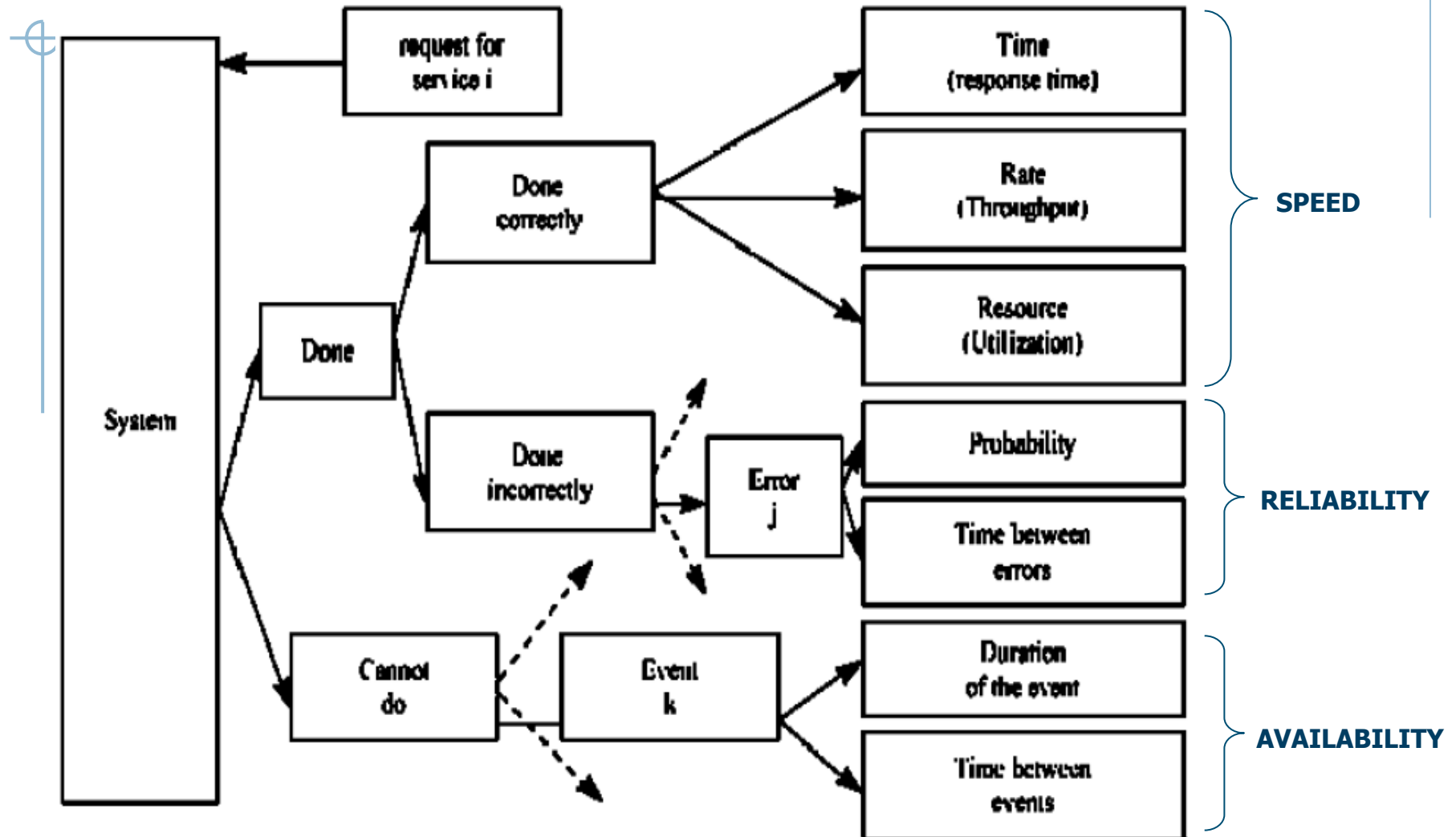
Concorde = 178.200 pmp

Il Boeing 747 produce $286.700/178.200 = 1.6$ volte “più benefici”

Il Boeing 747 produce 1.6 volte (“60%”) più benefici in termini di throughput



Selezionare le metriche (criteri di valutazione)



- Metriche globali: riflettono l'utilità del sistema
- Metriche individuali: riflettono l'utilità del singolo utente



Metriche globali e individuali

- Globali
 - Resource utilization, reliability, availability, throughput
- Individuali
 - Response time, throughput
- Ci sono casi in cui la decisione che ottimizza le metriche individuali è diversa da quella che ottimizza la metrica di sistema
 - Esempio: throughput totale e dei singoli nodi
 - Throughput costante per la rete, incrementando il throughput di un nodo, diminuisce quello di qualcun altro (impattando la fairness)
- Le metriche vengono selezionate in base a
 - Bassa variabilità (per ridurre il numero di ripetizioni)
 - Non ridondanza (metriche equivalenti sono inutili)
 - Completezza (tutti i possibili esiti devono essere catturati)

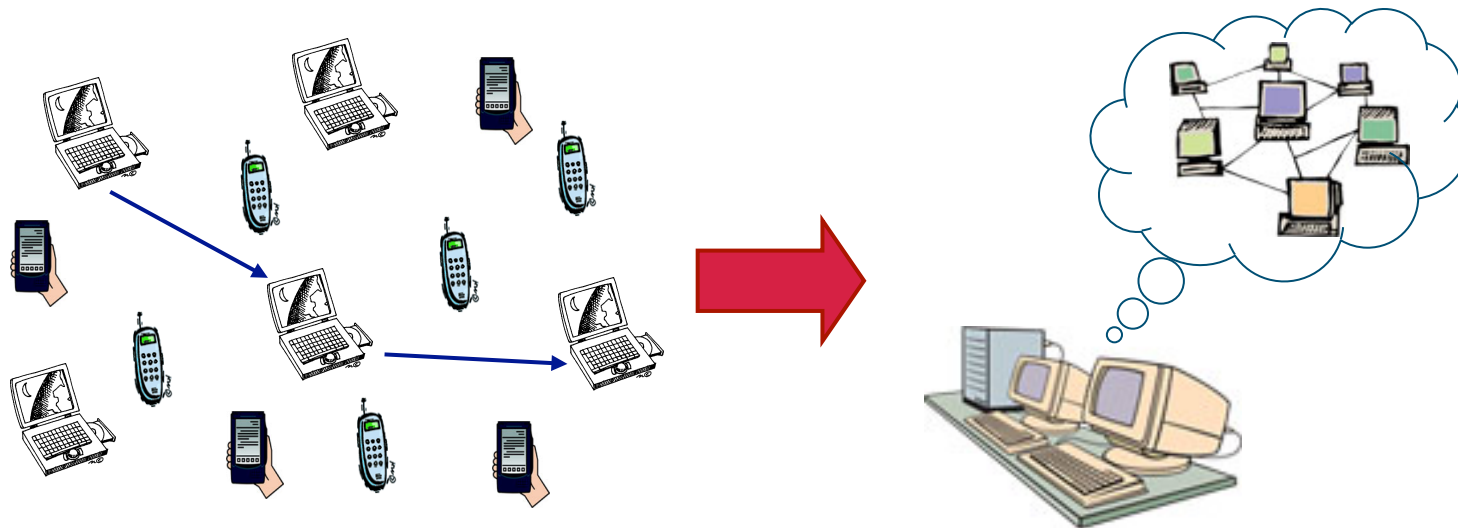


Outline

- Valutazione delle prestazioni di un sistema
- **Introduzione alla simulazione**
- Verifica ed analisi dei risultati simulativi



Introduzione alla simulazione



- Cosa è un simulatore di rete
 - Strumento software per la modellazione dei protocolli di rete (wired e wireless)
- Scopo:
 - Ricostruire un sistema che evolve come il sistema reale secondo alcuni aspetti, basandosi su un modello



Quando simulare

- Studio e sperimentazione delle interazioni interne di un sistema complesso (per es. TCP in sistemi wired e wireless)
- Valutazione delle prestazioni di un sistema prima della costruzione del prototipo
- Verifica di soluzioni analitiche
- Largamente diffuso in ambienti di ricerca
 - progettazione di nuovi protocolli
 - analisi del traffico
 - confronto tra protocolli
- Uso della simulazione per scopi didattici (comprendere meglio un sistema)



Perchè simulare

- Una sola workstation è necessaria per eseguire una simulazione
- Il simulatore permette di esaminare facilmente una ampia *varietà di scenari* in un tempo relativamente breve
- E' possibile simulare *topologie* di rete *complesse*, difficili e costose da realizzare in un test-bed
 - Ad-hoc o sensor networks di larga scala
 - Mobilità dei nodi
- Facile testare l'impatto di *modifiche* nei protocolli simulati



Pro e contro della simulazione

- Pro
 - Verifica del funzionamento di un nuovo sistema prima della costruzione del prototipo
 - Possibilità di eseguire un facile debugging del protocollo simulato
 - Possibilità di analizzare la scalabilità di un sistema
 - Identificazione delle vulnerabilità del sistema
 - Flessibilità nello studio del comportamento del sistema
- Contro
 - La creazione del modello e la sua validazione richiedono una comprensione dello strumento di simulazione
 - Non è possibile catturare svariati aspetti del sistema simulato (es. in NS2, prestazioni locali ai singoli nodi)



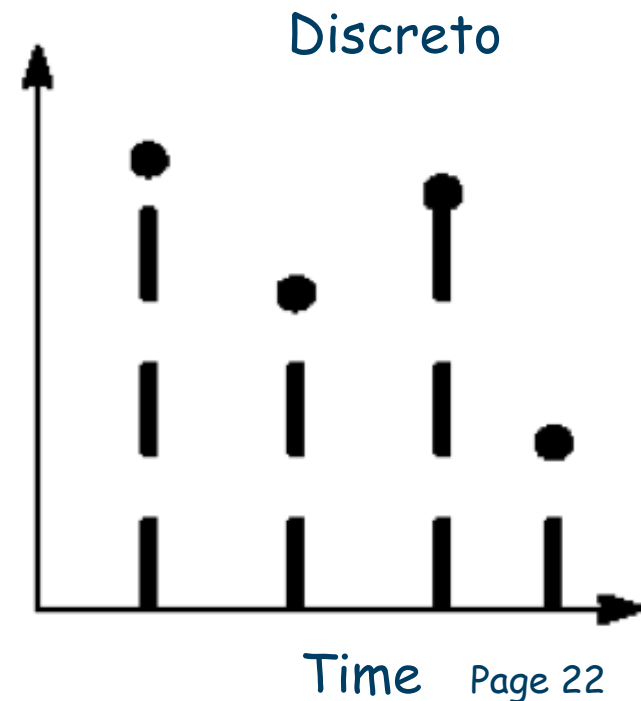
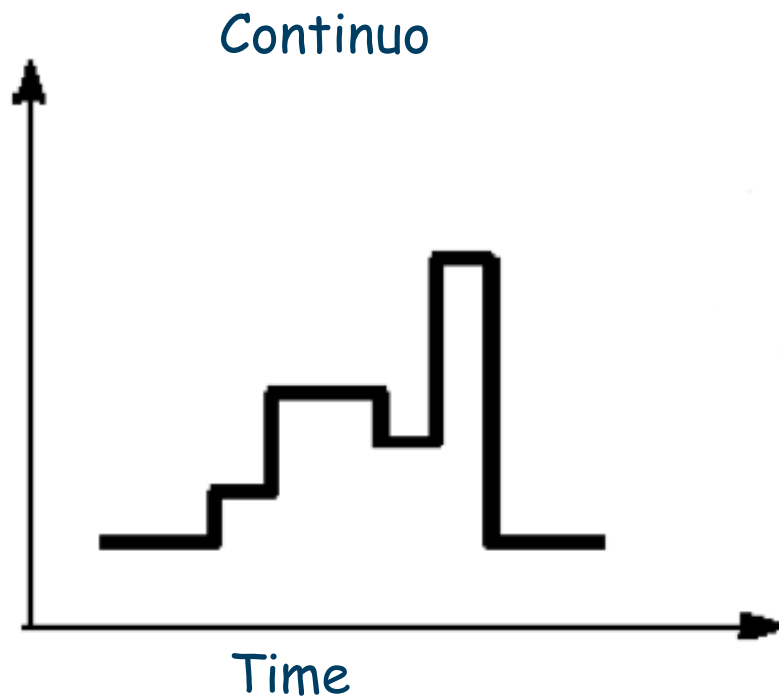
Terminologia della simulazione

- **Variabili di stato**
 - Variabili i cui valori definiscono lo stato del sistema
 - Network simulation: lista dei nodi, coda di trasmissione dei pacchetti, mac e routing utilizzati
- **Evento**
 - Un cambiamento nello stato del sistema
 - Network simulation: trasmissione di un pacchetto, ricezione di un pacchetto, introduzione di un nuovo nodo



Terminologia della simulazione

- **Modelli di *tempo* continuo e discreto**
 - **Continuo**: il sistema è definito per ogni istante di tempo
 - **Network simulation**: il numero di nodi, la comunicazione tra nodi è definita in ogni istante di tempo
 - **Discreto**: il sistema è definito solo in alcuni istanti di tempo
 - **Lezioni**: con incontri settimanali

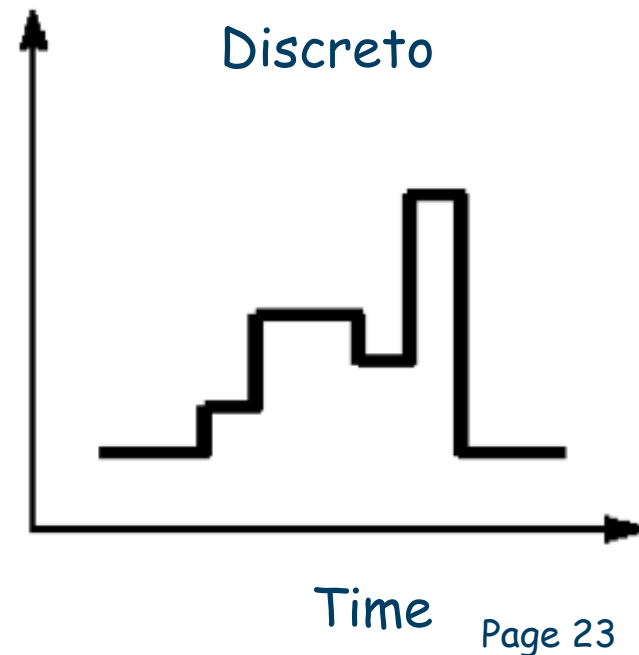
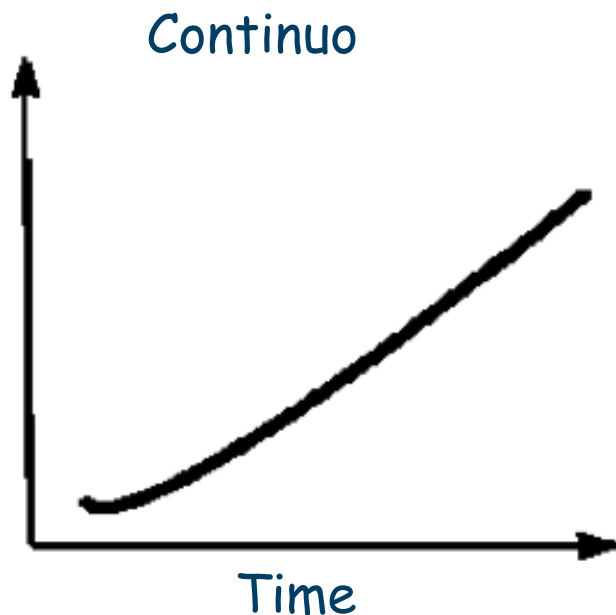




Terminologia della simulazione

- **Modelli di *stato* continuo e discreto**

- **Continuo:** le variabili di stato assumono valori continui
- **Lezioni:** tempo speso dagli studenti sul corso di reti di elaboratori 😊
- **Discreto:** le variabili di stato assumono valori discreti
- **Network simulation:** numero di nodi, o la lunghezza della coda di trasmissione dei pacchetti





Terminologia della simulazione

- **Modelli di *stato* discreto = Modello ad *eventi* discreti**
- **Modelli di *stato* continuo = Modello ad *eventi* continui**
- **La continuità del tempo non implica la continuità dello stato e viceversa!**
- **Quattro possibili combinazioni:**
 1. **Stato discreto / Tempo discreto**
 2. **Stato discreto / Tempo continuo**
 3. **Stato continuo / Tempo discreto**
 4. **Stato continuo / Tempo continuo**



Terminologia

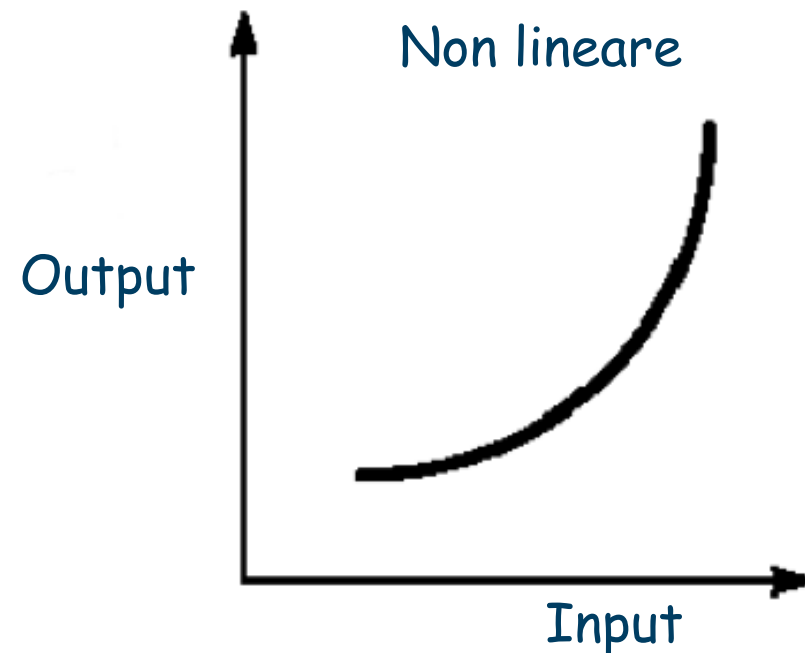
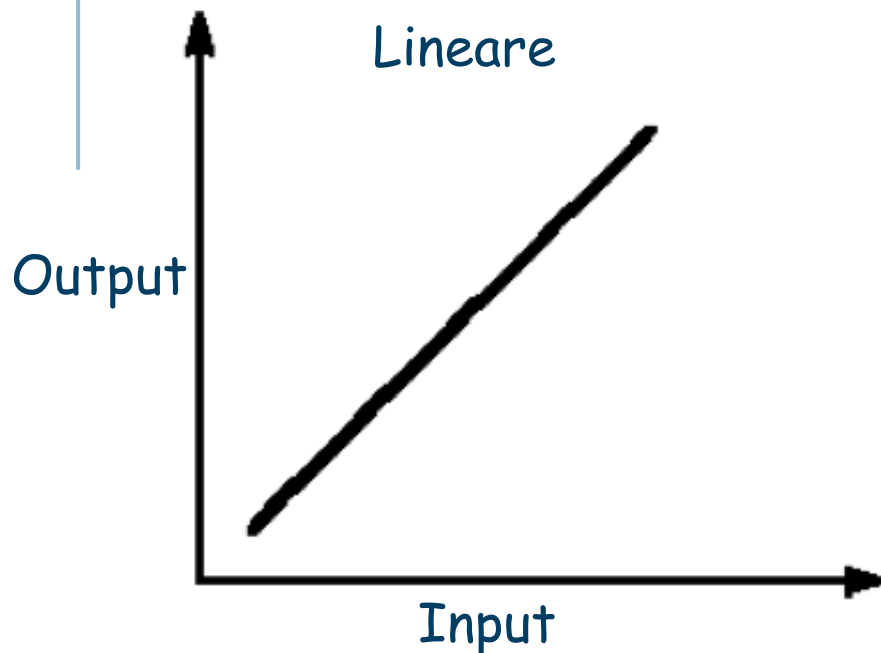
- **Modelli deterministici e probabilistici**
 - Deterministico: si può predire il risultato con certezza
 - Probabilistico: ogni ripetizione sullo stesso insieme di parametri di input produce un diverso risultato
- **Modelli statici e dinamici**
 - Statico: il tempo non è una variabile
 - Dinamico: lo stato del sistema cambia in funzione del tempo
- **Modelli aperti e chiusi**
 - Aperto: l'input è esterno al modello e indipendente da esso
 - Chiuso: non c'è input esterno

Computer system models: tempo continuo, stato discreto, probabilistico, dinamico e non lineare. Aperti o chiusi, stabili o instabili.



Terminologia

- **Modelli lineari e non lineari**
 - **Lineare:** i parametri di output sono una funzione lineare dei parametri di input
 - **Non lineare:** altrimenti



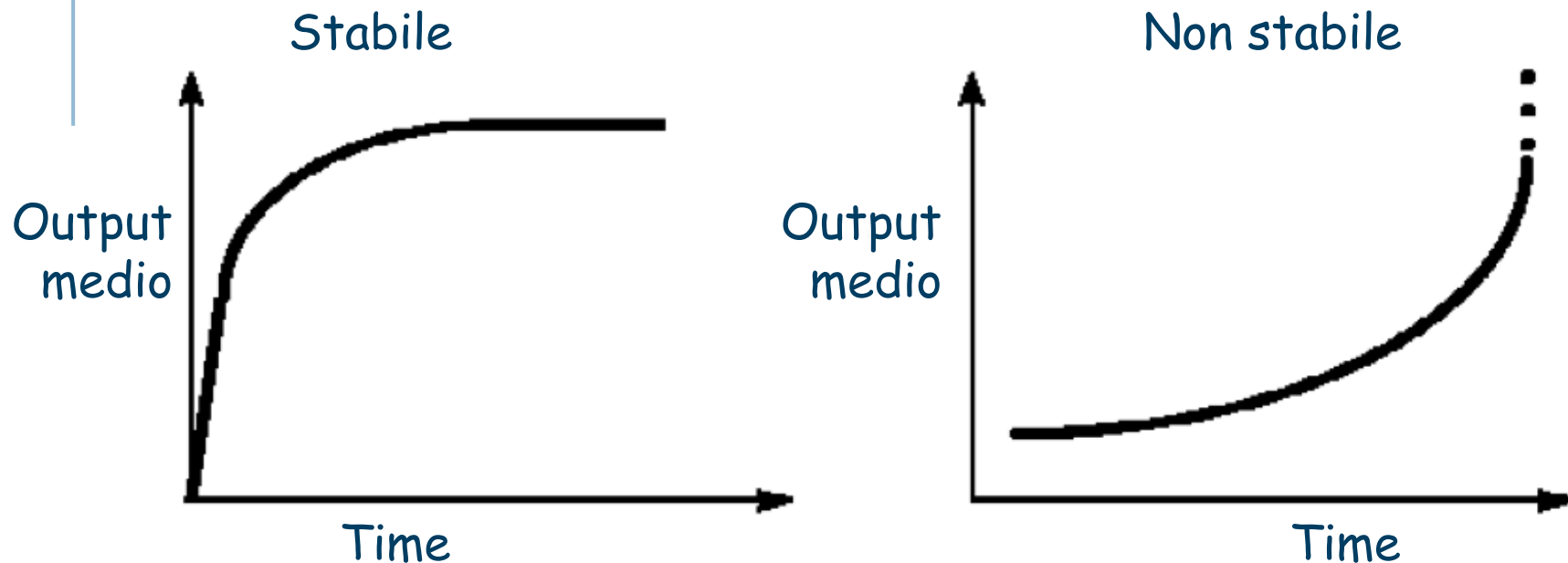
Computer system models: tempo continuo, stato discreto, probabilistico, dinamico e non lineare. Aperti o chiusi, stabili o instabili.



Terminologia

- **Modelli stabili e non stabili**

- **Stabile:** il comportamento del sistema è stabile, indipendentemente dal tempo
- **Instabile:** il comportamento del modello cambia continuamente



Computer system models: tempo continuo, stato discreto, probabilistico, dinamico e non lineare. Aperti o chiusi, stabili o instabili.



Tipi di simulazione

- Monte Carlo
 - Simulazione statica, senza l'ascissa temporale
 - Modelli di fenomeni probabilistici che non cambiano caratteristiche al variare del tempo
- Trace-driven
 - Simulazione che prende in input una traccia di un sistema reale (lista di eventi ordinati nel tempo)
- A eventi discreti
 - Simulazione con modello di stato discreto
 - Network simulation: numero di pacchetti nella coda di trasmissione
 - Il modello del tempo può essere discreto o continuo



Simulatore a eventi discreti: componenti

- **Scheduler degli eventi**
 - Mantiene una lista di eventi che devono essere eseguiti
- **Simulazione del tempo**
 - Ogni simulazione ha una variabile globale che rappresenta il tempo simulato (diverso dal tempo reale del simulatore)
 - Lo scheduler è responsabile dell'avanzamento del tempo
 - Unità di tempo: incremento unitario, con controllo degli eventi che devono essere eseguiti
 - Event-driven: l'incremento del tempo dipende dal prossimo evento da eseguire
- Variabili di stato del sistema
- **Routine dell'evento**
 - Routine che viene eseguita quando deve essere eseguito l'evento
- Routine di input, inizializzazione, trace
- Programma principale



Outline

- Valutazione delle prestazioni di un sistema
- Introduzione alla simulazione
- **Verifica ed analisi dei risultati simulativi**



Errori comuni legati alla simulazione 1/2

1. Inappropriato livello di dettaglio
Più dettagli => Più tempo => Più bug => Più calcolo =>
Più parametri ≠ Più accurato
2. Progettazione impropria
Tropo generica => Più tempo e meno accurata
3. Modelli non verificati: Bug nel sistema
4. Modelli non validi: risultati lontani dalla realtà
5. Non accurata gestione delle condizioni iniziali del sistema
6. Simulazioni troppo corte: risultati poco significativi
7. Utilizzo di generatori random poco accurati
8. Impropria inizializzazione dei generatori random: tutte i flussi nella simulazione utilizzano gli stessi valori



Errori comuni legati alla simulazione 2/2

1. Non accurata stima dei tempi
2. Provare ad ottenere dei risultati non ottenibili
3. Carenza in una delle componenti essenziali:
 - a) Conoscenza del sistema
 - b) Modellazione del sistema
 - c) Implementazione del sistema
4. Inadeguata partecipazione dell'utente finale
5. Documentazione obsoleta o inesistente
6. Mancanza di capacità nello gestire software pre-esistenti di larghe dimensioni e necessità di sviluppo di nuovi
7. Interpretazione di risultati "misteriosi"



Cosa fare quando si vuol sviluppare un simulatore 1/2

1. Prima di iniziare lo sviluppo:
 - a) I risultati da raggiungere e verificare sono bene definiti?
 - b) Il modello è sufficientemente dettagliato da permettere di raggiungere quanto richiesto?
 - c) Il team di sviluppo possiede adeguate competenze in: modellazione, implementazione, conoscenza di sistemi?
 - d) Il tempo pianificato per lo sviluppo del simulatore è sufficiente?

2. Durante la fase di sviluppo:
 - a) Il generatore di numeri random è corretto? Uniformità ed indipendenza dei valori generati sono state verificate?
 - b) Il modello è stato controllato e rivisto regolarmente con l'utente finale?
 - c) Il modello è sufficientemente documentato?



Cosa fare quando si vuol sviluppare un simulatore 2/2

1. Quando si inizia la simulazione:
 - a) La lunghezza della simulazione è appropriata?
 - b) Le metriche di interesse sono adeguatamente collezionate ed investigate?
 - c) Il modello è stato verificato a sufficienza?
 - d) Il modello è stato validato prima di usare i risultati?
 - e) Se ci sono risultati inattesi, sono stati investigati e validati?



Verifica dei risultati e quindi del modello

1. **Debugging**

Effettuare semplici test e poi via via più complessi:

Es. La somma delle probabilità assunte fa 1.

Il numero di lavori da svolgere = lavori generati – lavori svolti

2. **Investigazione del sistema creato:**

Spiegare il sistema ad altri (funziona anche se dormono)

3. **Esecuzione di test semplici con valori costanti**

Assicurarsi che i risultati prodotti rispettino quanto atteso

4. **Consistenza dei risultati**

Eeguire simulazioni simili ma diverse: dovrebbero produrre stessi trend di risultati. Esempio, una rete con 4 nodi a 100Mbps o due nodi a 200Mbps

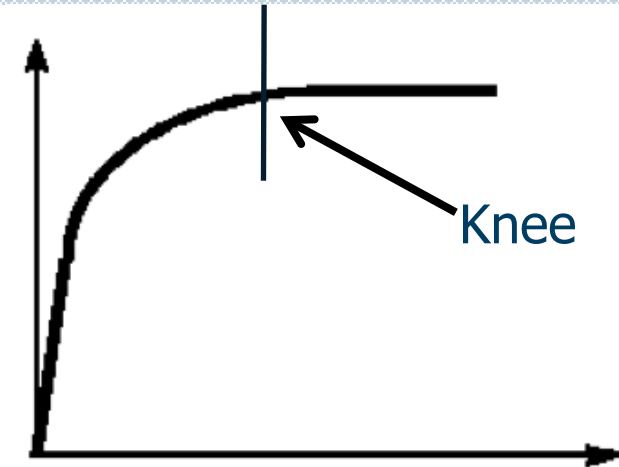
5. **Esecuzione di test degenerativi.**

Configurazioni o workload estremi: zero CPU, one CPU e zero disk



Analisi dei risultati

In genere si è interessati ai risultati sul sistema una volta che si stabilizza

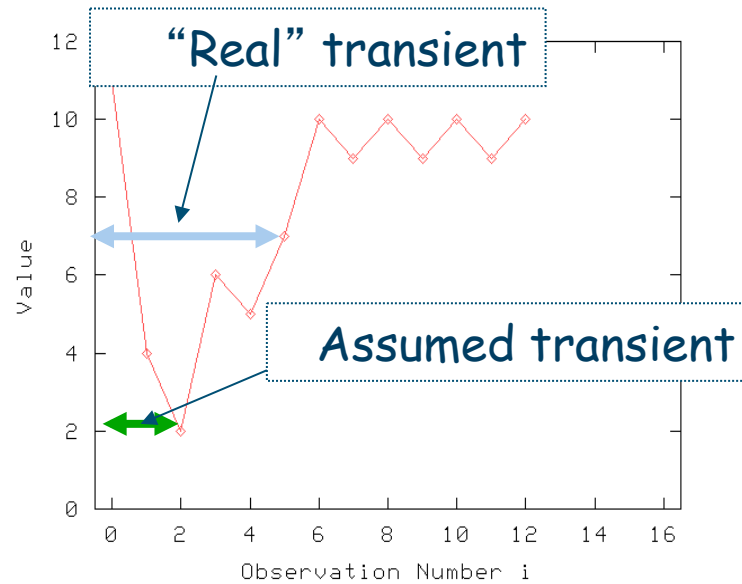
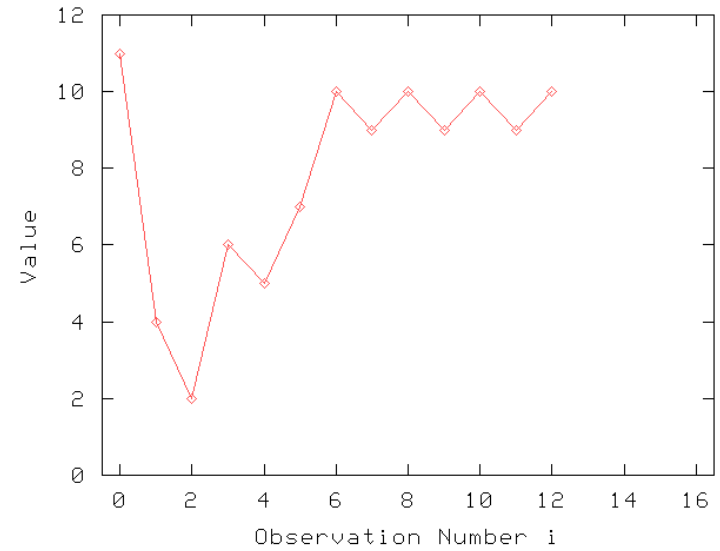
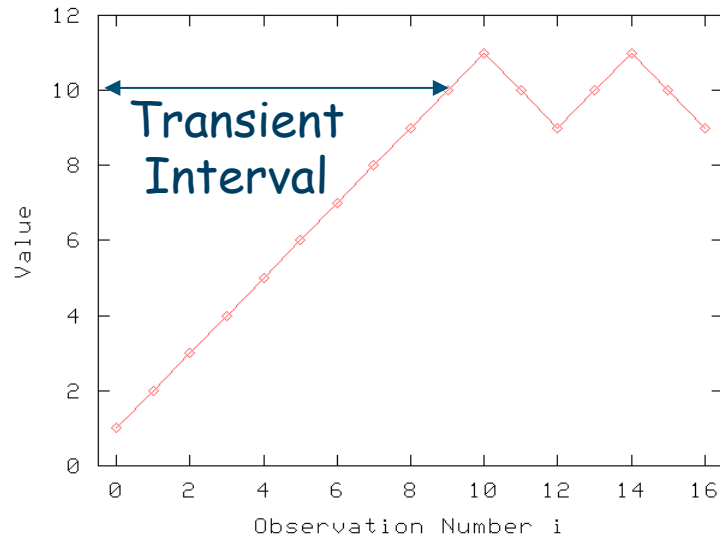


Rimuovere effetti della parte iniziale o transitorio

- Simulazioni molto lunghe
- Appropriate inizializzazione del sistema
- **Cancellazione dei dati iniziali (quanto cancello?)**
 - Truncation
 - Initial Data Deletion
 - Moving average of independent replications
 - Batch means



Truncation





Quando terminare la simulazione

Bisogna anche capire quando terminare la simulazioni

- Simulazioni troppo corte, poca confidenza nei dati
- Simulazioni troppo lunghe, spreco di risorse
- Evitare quando si termina una simulazione di perdere dati importanti o considerare valori non necessari che alterino la confidenza dei dati
- Per aumentare la confidenza, calcolare la media su più repliche indipendenti



Ricapitolazione

- Valutazione delle prestazioni di un sistema
 - Approccio sistematico alla valutazione
 - Diversi approcci: analitico, simulativo, misurazioni
 - Importanza della selezione delle metriche
- Introduzione alla simulazione
 - Vantaggi e svantaggi
 - Terminologia e modelli
 - Differenti tipologie di simulazione
- Verifica ed analisi dei risultati simulativi
 - Tipici errori simulativi
 - Verifica del sistema
 - Analisi dei risultati



Domande ?



Simulatore di rete NS2

Roberto Petroccia
petroccia@di.uniroma1.it



Riferimenti

- Architettura e utilizzo del Network Simulator NS2
 - <http://www.isi.edu/nsnam/ns/>
- Tutorials:
 - Marc Greis's tutorial
<http://www.isi.edu/nsnam/ns/tutorial/index.html>
 - "NS for Beginners" di Altma e Jimenez
 - Ns Manual
 - "NS by Example" di J. Chung e M.Claypool
<http://nile.wpi.edu/>



Outline

- Architettura del Network Simulator NS2
- Utilizzo di NS2
- Esempi di utilizzo del simulatore (protocolli UDP e TCP)



Network Simulator NS2

- Simulatore ad eventi discreti (sviluppato all'UC Berkley)
- Simulazione a livello di pacchetto
- Modellazione dal livello DataLink a livello Applicazione
 - Protocolli livello MAC
 - Algoritmi di routing: Dijkstra, etc...
 - Meccanismi di gestione delle code dei router: DropTail, Random Early Detection/Drop (RED) e (Class-Based Queueing) CBQ
 - Protocolli di rete: TCP, UDP (over IP e IPv6)
 - Sorgenti di traffico: FTP, Telnet, Web, CBR e VBR
- Codice sorgente di **pubblico dominio (open source)**
- In continua evoluzione, aggiornato e modificato da ricercatori e studenti di tutto il mondo
- Sito ufficiale: <http://www.isi.edu/nsnam/ns/>



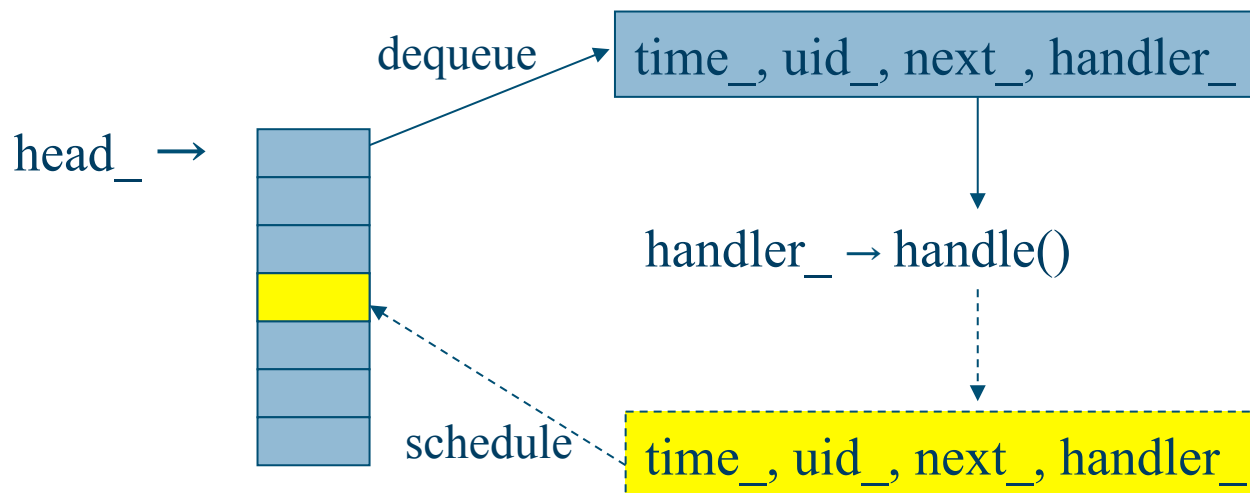
Network Simulator NS2

- Approccio **modulare**
- Implementato in **tcl** (Tool Command Language) e **C++**
- Il linguaggio di scripting *tcl* è usato per eseguire i comandi dell'utente, ovvero per descrivere lo scenario simulativo
 - Configurare topologia, nodi, canale, e schedulare gli eventi
- Il linguaggio C++ è usato per implementare il simulatore
 - Implementazione dei protocolli di rete (mac, network, transport, application)



Implementazione di NS2

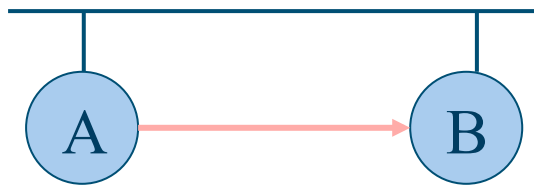
- Simulatore di rete
 - a eventi discreti (le variabili di stato assumono valori discreti)
 - con modello del tempo continuo (le variabili di stato sono sempre definite)
 - avanzamento del tempo event-driven
- Modellazione ad eventi
 - Lo scheduler
 - Mantiene la lista di eventi che devono essere eseguiti
 - Estrae il primo evento dalla coda e lo esegue invocando l'handler associato
 - Ogni evento è eseguito in un istante di tempo (simulato) virtuale, ma impiega una durata arbitraria di tempo reale
- NS usa un singolo thread di controllo





Esempio

Consideriamo due nodi A e B, con A che spedisce un pacchetto a B



modello
CSMA/CD

t=1.0:

- A invia il pacchetto alla NIC
- NIC di A inizia il carrier sense

t=1.005:

- NIC di A conclude il cs, e inizia la trasmissione

t=1.006:

- NIC di B inizia a ricevere il pacchetto

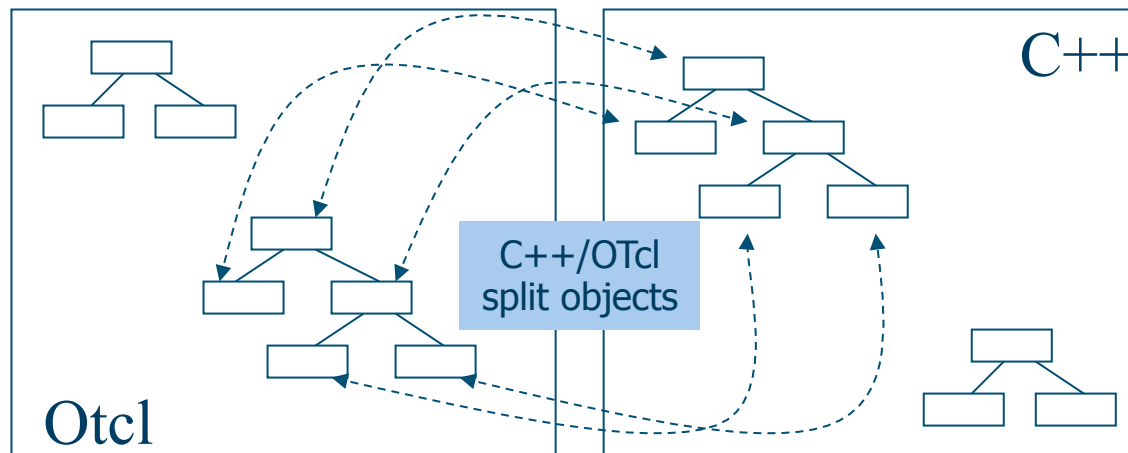
t=1.01:

- NIC di B conclude ricezione
- NIC di B passa il pacchetto all'applicazione



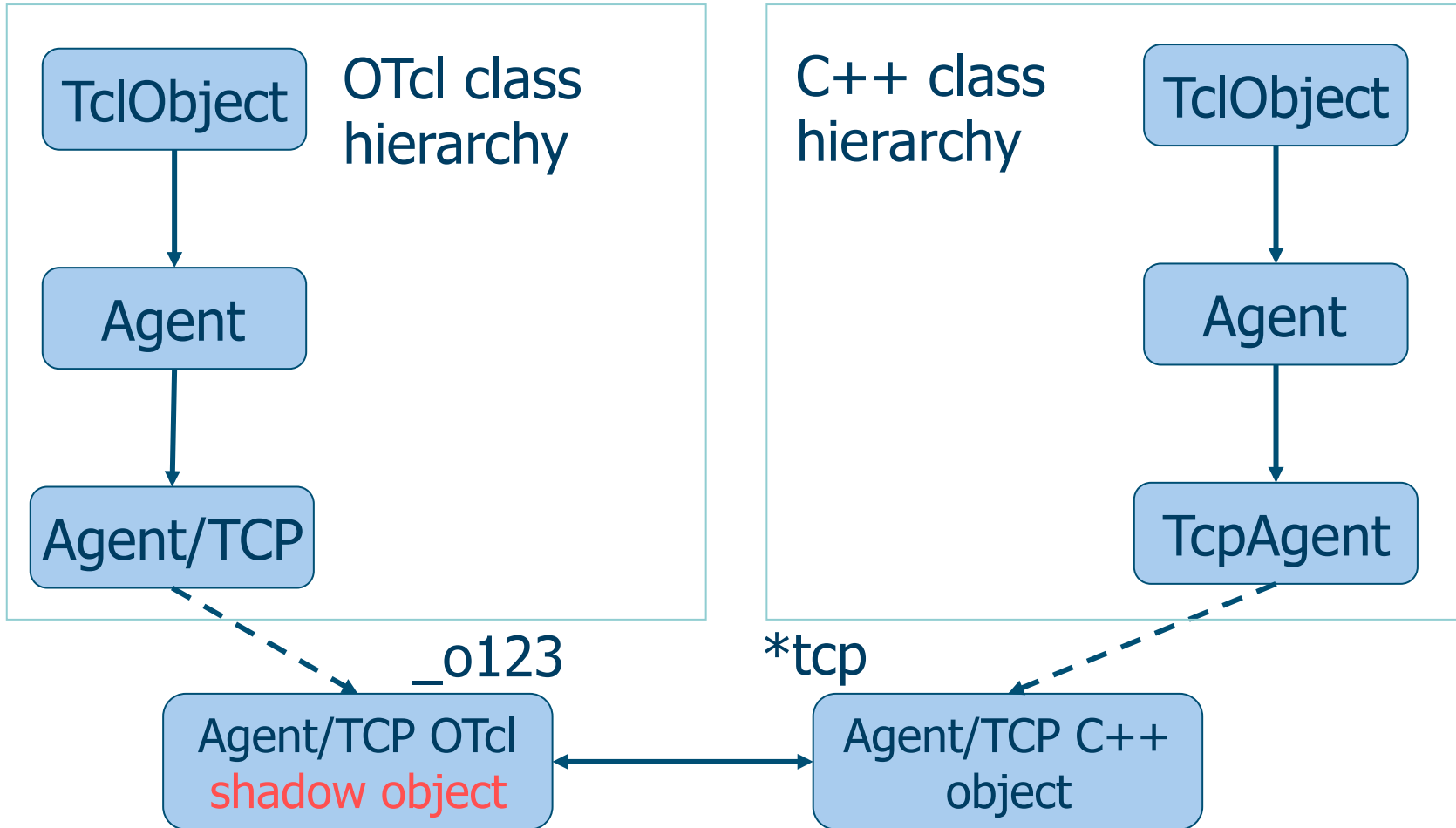
NS2: struttura orientata agli oggetti

- Il simulatore supporta una gerarchia di classi in C++ (*gerarchia compilata*), e una simile gerarchia di classi all'interno dell'interprete OTcl (*gerarchia interpretata*).
- Le due gerarchie sono strettamente legate l'una con l'altra; c'è una corrispondenza one-to-one tra una classe nella gerarchia interpretata e una nella gerarchia compilata.





Esempio: split object



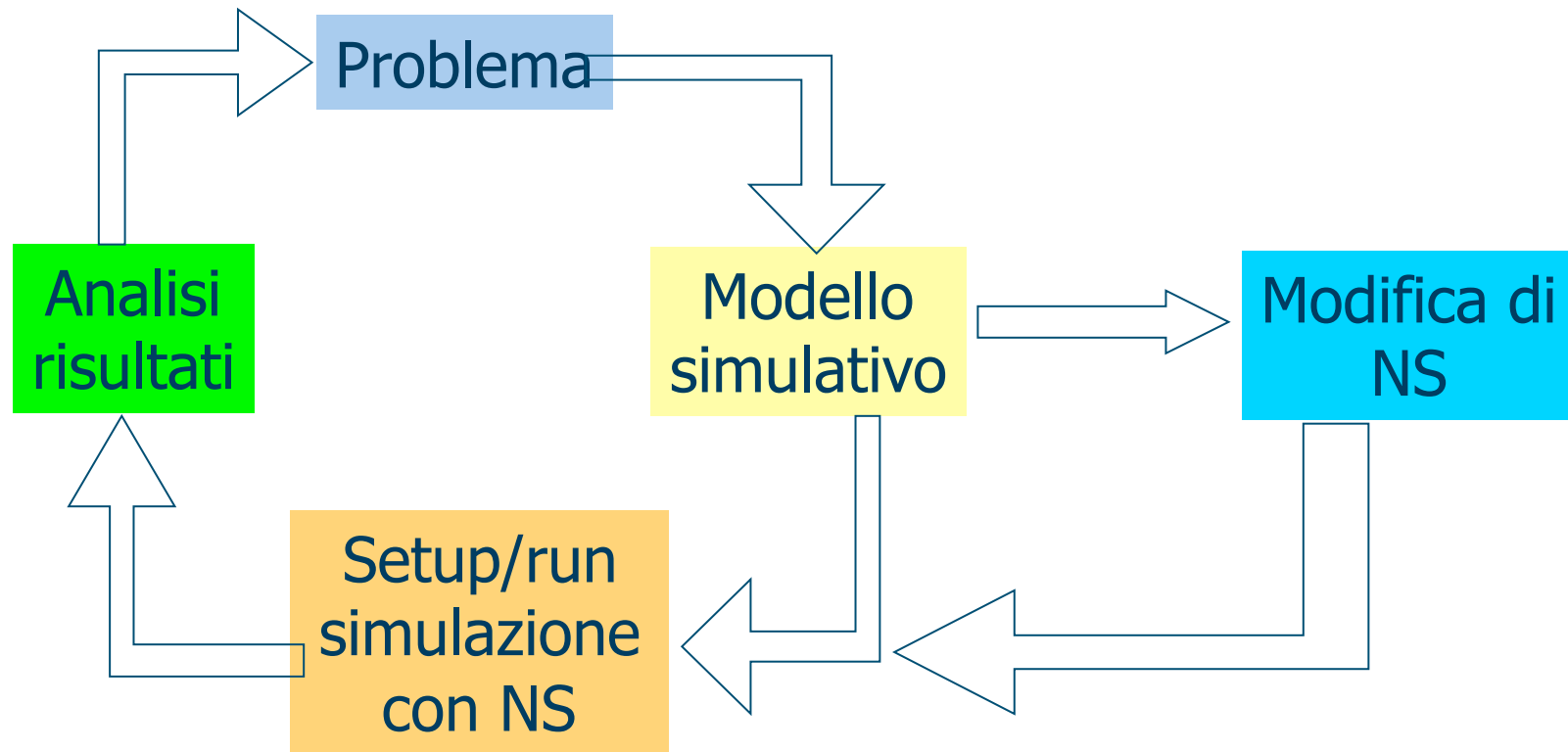


Outline

- Architettura del Network Simulator NS2
- **Utilizzo di NS2**
- Esempi di utilizzo del simulatore (protocolli UDP e TCP)

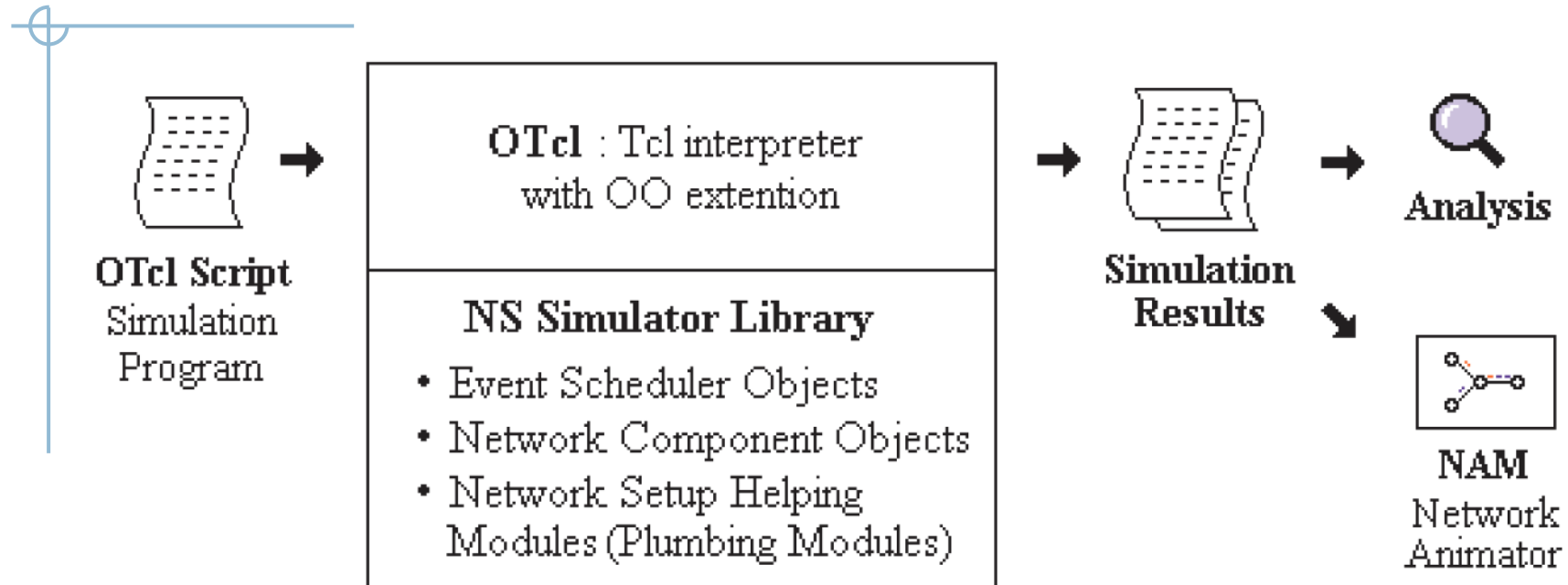


Usare NS2: fasi





Architettura di NS2



- Dove operare:
 - Tcl: script per costruire il modello di rete che si vuole simulare
 - C++: per implementare nuovi protocolli è necessario creare o modificare classi C++
- Due tipi di output
 - out.tr -> trace file per successiva elaborazione
 - out.nam -> file per visualizzazione grafica



Passi per eseguire la simulazione

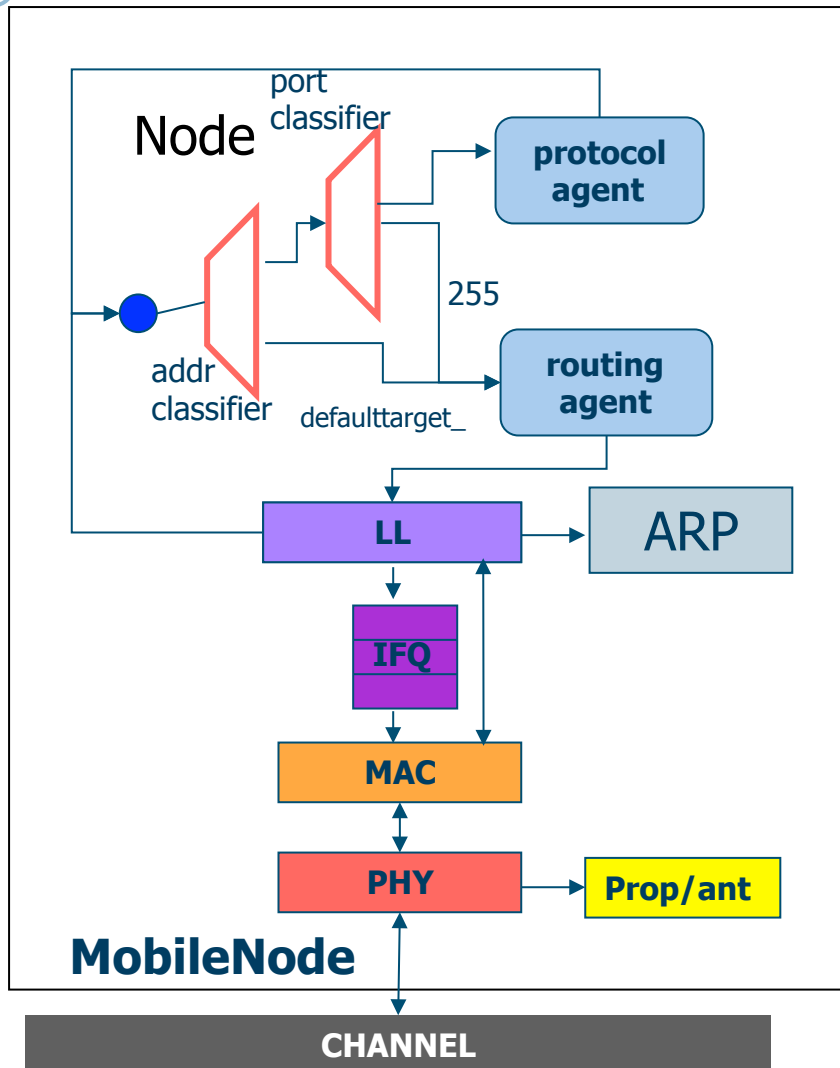
1. Descrivere lo scenario simulativo in uno script tcl
 - Gestione del simulatore (inizializzazione e terminazione)
 - Definizione topologia (nodi, link)
 - Definizione degli agenti (TCP, UDP)
 - Definizione delle applicazioni (FTP, CBR)
 - Schedulazione degli eventi
 - Generazione file di trace

2. Eseguire la simulazione
 - NS interpreta lo script Otcl

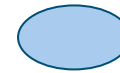
3. Visualizzare e analizzare i risultati
 - Visualizzazione tramite "nam"
 - Analisi dei risultati (file di trace)



Struttura di un nodo



Classifier: Forwarding



Agent: Protocol Entity



Node Entry



LL: Link layer object



IFQ: Interface queue



MAC: Mac object



PHY: Net interface



Radio propagation/
antenna models



Configurazione e inizializzazione di una rete wireless

```
#-----  
#  
#          --- Parametri del sistema di trasmissione ---  
#-----  
set val(chan)          Channel/WirelessChannel          ;# Tipo di canale\<\  
set val(prop)          Propagation/TwoRayGround         ;# Mod. di propagazione\<\  
set val(netif)         Phy/WirelessPhy                 ;# Tipo di interfaccia\<\  
set val(mac)           Mac/802_11                     ;# Tipo di MAC \<\  
set val(ifq)           Queue/DropTail/PriQueue         ;# Tipologia di coda \<\  
set val(ll)            LL                              ;# Link Layer\<\  
set val(ant)           Antenna/OmniAntenna             ;# Modello di antenna\<\  
set val(ifqlen)        50                             ;# Num. di pacch. in IFQ\<\  
set val(nn)            200                             ;# Numero di nodi\<\  
set val(adhocRouting)  AODV                           ;# Protocollo di routing\<\  
#-----  
#          --- Configurazione dei nodi ---  
#-----  
$ns node-config  
  -adhocRouting        $val(adhocRouting) \<\  
  -llType              $val(ll) \<\  
  -macType             $val(mac) \<\  
  -ifqType            $val(ifq) \<\  
  -ifqLen             $val(ifqlen) \<\  
  -antType            $val(ant) \<\  
  -propType           $val(prop) \<\  
  -phyType            $val(netif) \<\  
  -channel            $chan \<\  
  -topoInstance       $topo \<\  
  -agentTrace         OFF \<\  
  -routerTrace        OFF \<\  
  -macTrace           ON \<\  
  -movementTrace      OFF  
#-----
```



Definizione della topologia

- Creazione degli oggetti di base

- Simulatore

```
set ns [new Simulator]
```

(creazione scheduler)

(riferito come \$ns)

- Nodi

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

(node è metodo di Simulator)

(riferite come \$n0, \$n1)

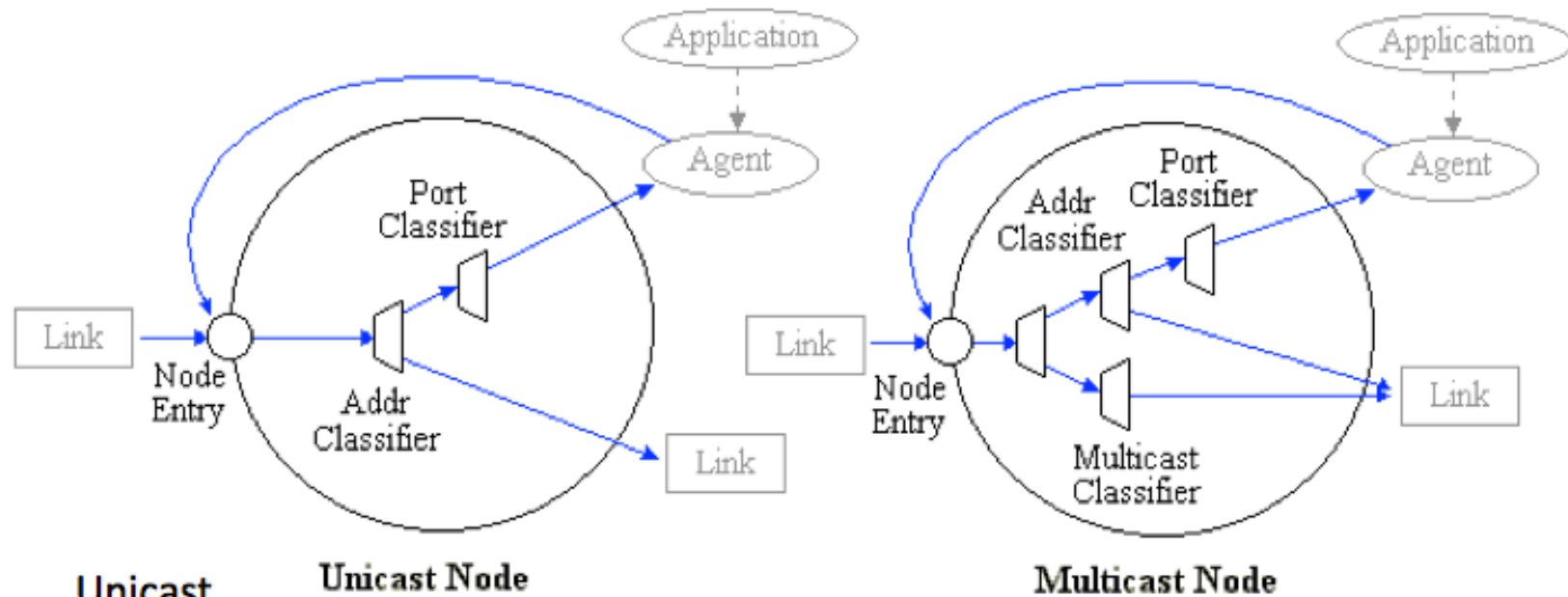
n0

n1

- Tipo di canale e quindi di link fa parte della definizione del nodo



Nodi e routing



Unicast

Unicast Node

- `$ns rproto type`
(type: Static, Session, Distance Vector, cost, multi-path)

Multicast

- `$ns multicast` (right after set `$ns [new Scheduler]`)
- `$ns mrtproto type`
(type: CtrMcast, DM, ST, BST)



Schedulazione degli eventi

- Lo scenario simulativo definito (topologia, agenti, e applicazioni) deve essere "animato"
- Stabilire *quando* eseguire gli eventi
- La maggior parte degli eventi sono nascosti all'utente, poiché generati da altri eventi
- Gli eventi vengono schedulati usando il comando

```
$ns at <time> <event>
```

- Lo scheduler viene avviato tramite il comando

```
$ns run
```



Esempio di schedulazione degli eventi

- Schedulazione dell'avvio e terminazione di una applicazione CBR

```
$ns at 0.5 "$cbr0 start"
```

```
$ns at 5.5 "$cbr0 stop"
```

- Schedulazione di una procedura "finish" definita dall'utente

```
$ns at 150.0 "finish"
```



Eseguire la simulazione

L'esecuzione della simulazione avviene facendo interpretare lo script Otcl a NS

```
ns mio_script.tcl
```



Domande?



Estensione MIRACLE di NS2

Roberto Petroccia + Loreto Pescosolido

petroccia@di.uniroma1.it

loreto.pescosolido@uniroma1.it



Multi InteRfAce Cross Layer Extension

- ns MIRACLE: Estensione di ns2
- Sito ufficiale telecom.dei.unipd.it/pages/read/58/

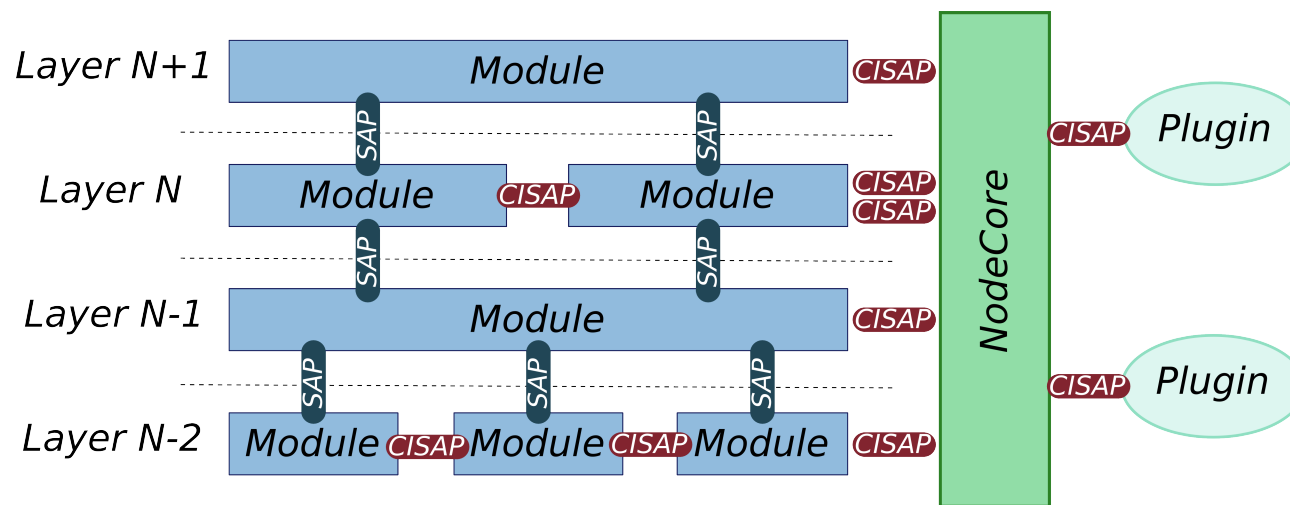
Limiti di ns2:

1. Non supporta la possibilità di avere più interfacce radio all'interno dello stesso nodo
2. Non presenta un supporto per lo scambio di messaggi *cross-layer*
3. Modello di canale semplicistico
4. Mancanza di interfacce standard (codice difficilmente riutilizzabile)



Multi InteRfAce Cross Layer Extension

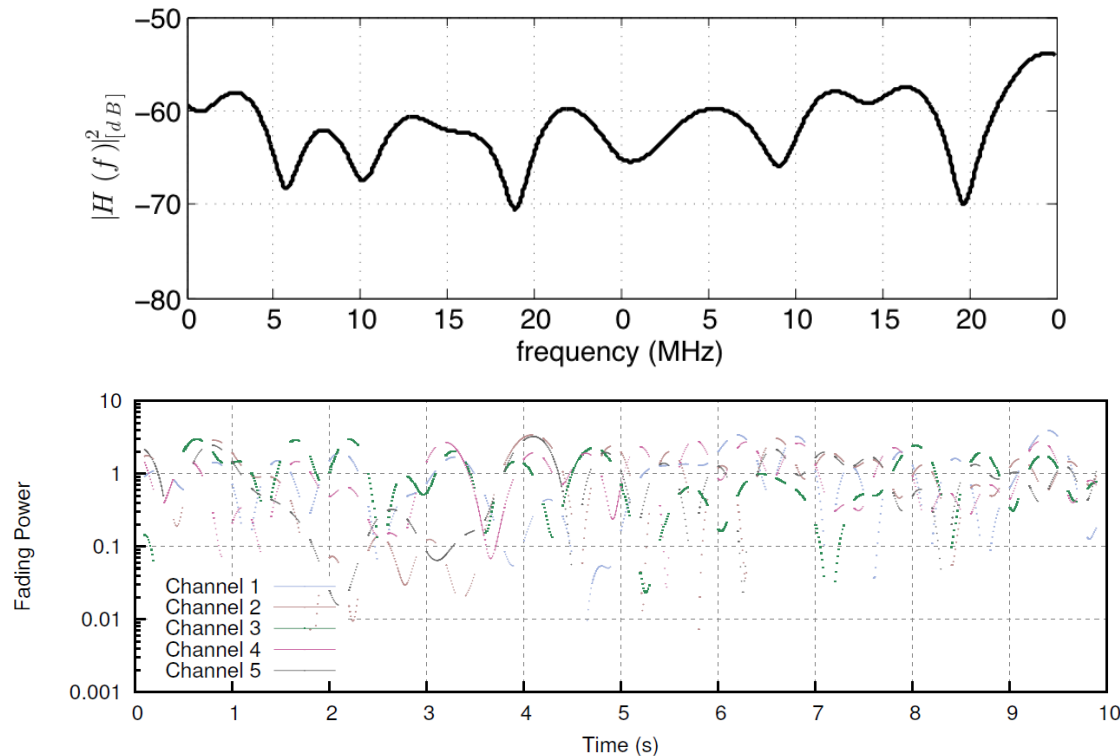
- Semplice interconnessione tra i diversi moduli nello stack
- Molteplici moduli per ogni livello della pila e supporto nativo per i messaggi *cross-layer*
- Migliore caratterizzazione dello strato fisico
- Spectrum awareness
- Interfacce standard per la creazione di nuove soluzioni ad ogni livello dello stack





Selezione adattiva dei sottocanali nei sistemi multi-carrier a banda larga

Esempio di fading selettivo in tempo e frequenza



Il canale viene stimato dai ricevitori per ogni Allocation Time Interval (ATI):

- 1. Uplink:** Stimato dalla BS alla ricezione
- 2. Downlink:** Stimato alla ricezione delle MS e comunicate alla BS tramite un canale dedicato



Selezione adattiva dei sottocanali nei sistemi multi-carrier a banda larga

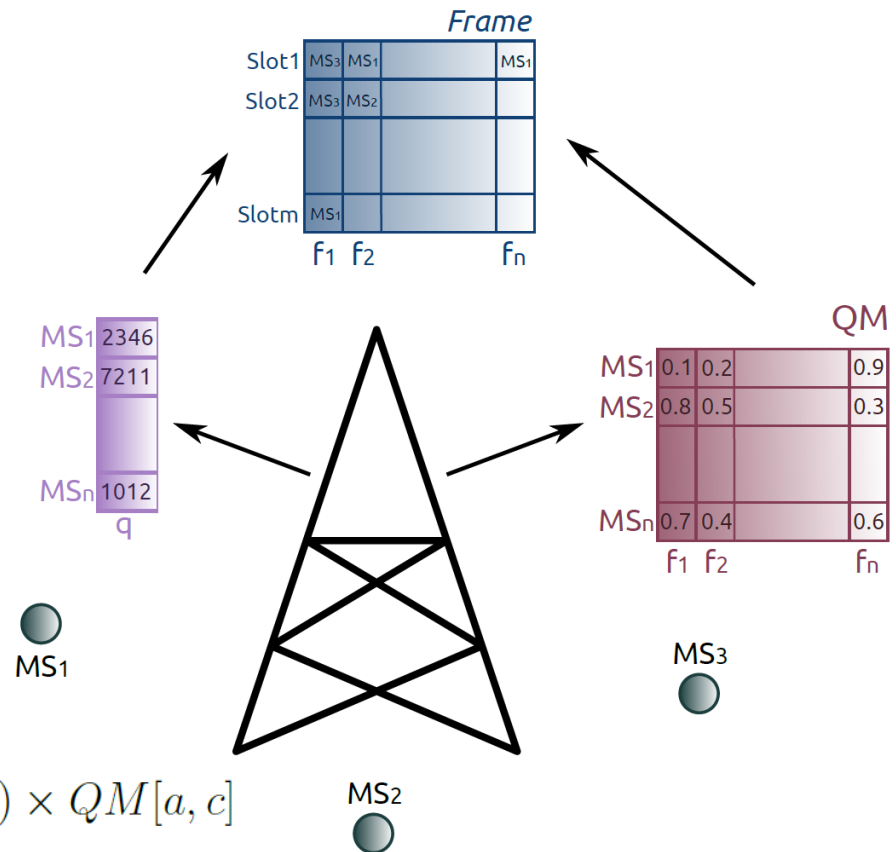
Grazie alla conoscenza dei canali, per ogni ATI la BS esegue un algoritmo di allocazione in modo da trovare il compromesso migliore tra throughput globale e fairness tra tutti gli utenti attivi.

Il nostro algoritmo iterativo considera:

- Qualità del canale
- Dimensione della coda

$$Quality(MS a, Channel c) = \alpha \times q(a) + (1 - \alpha) \times QM[a, c]$$

Ad ogni iterazione viene selezionata la migliore coppia utente-canale. A tale utente vengono allocati N slot nel canale scelto.





Selezione adattiva dei sottocanali nei sistemi multi-carrier a banda larga

- Grazie a questo tipo di resource allocation adattiva si riesce a ottimizzare le performance del sistema in termini di data rate per gli utenti, QoS e fairness.
- Rispetto a sistemi di generazioni precedenti si cerca di far utilizzare a ciascun utente la migliore risorsa per lui. Ne consegue che in situazioni di poco traffico ci saranno canali che non vengono utilizzati per lunghi periodi (quelli "cattivi" per gli utenti attivi). Mentre nei sistemi tradizionali si cercava comunque di far esplorare a ciascun utente la maggior quantità di frequenze possibile, nella speranza di fargli utilizzare anche quelle buone.



Aspetti di simulazione

- Simulatori di rete vs simulatori di link
- Astrazione dello strato fisico
- Un simulatore di rete “vuole” in input un valore di probabilità di errore nella ricezione di un pacchetto
- Problematiche relative all’inserimento di aspetti di variabilità del canale (e dei diversi canali) nei simulatori di rete



Aspetti di simulazione

Studio congiunto di:

- modelli adeguati per lo strato fisico
- moduli da inserire nel simulatore di rete che siano in grado di riprodurre le caratteristiche salienti prendendo in input un insieme di parametri che rappresentano la situazione attuale del canale e dell'interferenza



Nuovi paradigmi di comunicazione

La disponibilità di

1. Tecnologie per l'acquisizione dello stato dei canali in tempi rapidi (sia in termini di attenuazione che di interferenza)

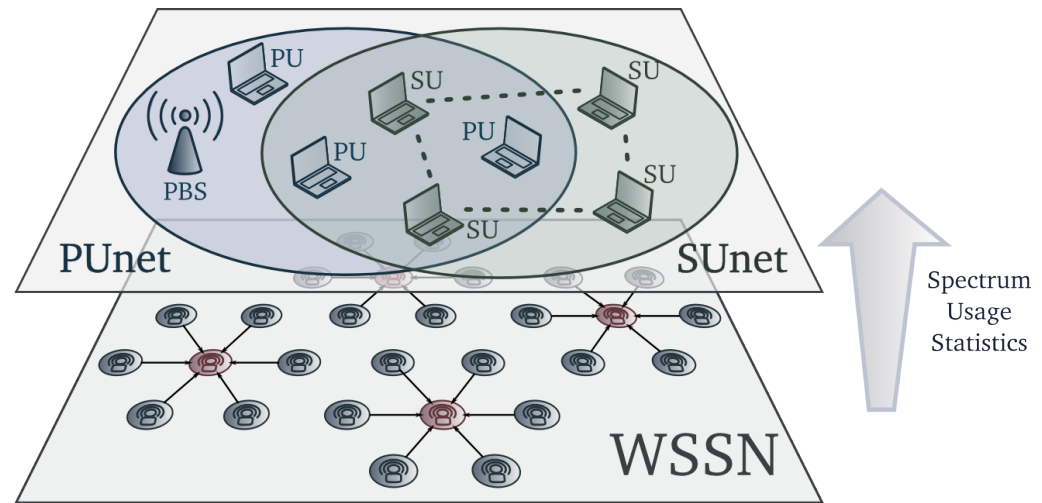
2. Tecnologie di trasmissione in grado di adattarsi velocemente a situazioni mutevoli

ha reso possibile la nascita di sistemi coesistenti tra loro. Ad esempio, le **cognitive radio**.

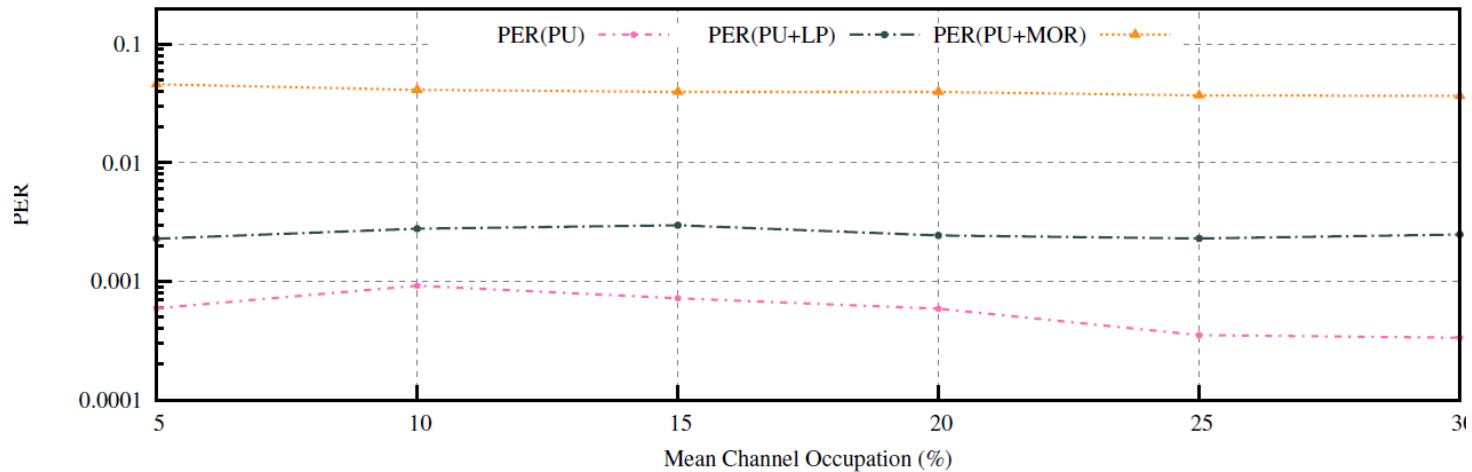


Cognitive Networks

Soluzione implementata in ns MIRACLE



Primary Network - Packet Error Rate





Domande?