

# SIMULACIÓN EN REDES DE COMUNICACIONES



## ¿Qué es una simulación?

- ✦ Es una técnica para el estudio cuantitativo de sistemas reales que consiste en:
  - ✦ Definir un modelo matemático del sistema
  - ✦ Representar numéricamente la evolución del modelo durante un cierto periodo de tiempo (mediante un programa de ordenador)
  - ✦ Medir el comportamiento del modelo y estimar las características de interés del sistema a partir de los datos recogidos



## ¿Para qué simular?

- ◆ Depende de lo que simulemos, pero típicamente para:
  - Estudiar un sistema con una configuración dada
  - Comparar varias configuraciones alternativas
  - Determinar la configuración de prestaciones óptimas



## ¿Qué otras técnicas hay?

- ◆ Experimentación sobre el sistema real
- ◆ Experimentación con modelos físicos (prototipos)
- ◆ Solución analítica de modelos matemáticos



## ¿Qué ventajas e inconvenientes tiene simular?

### ✦ Ventajas:

- ✦ Permite abordar modelos más complejos que los métodos analíticos
- ✦ Suple la experimentación con el sistema real cuando ésta es imposible o muy costosa

### ✦ Inconvenientes (limitaciones):

- ✦ Produce estimaciones, no soluciones "exactas"
- ✦ Normalmente requiere programas muy complejos y mucho tiempo de CPU
- ✦ Para entender porqué ocurren las cosas, es mejor el análisis



## Importante...

### ✦ La simulación debe complementarse con el análisis (y viceversa):

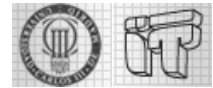
- ✦ Análisis de casos simples para validar simulador
- ✦ Simulación para validar análisis aproximados

### ✦ Simulación como "último recurso"



## Fases de la simulación

1. Estudio del problema
2. Recogida de datos sobre sistemas reales
3. Formulación de un modelo
4. Validación del modelo
5. Realización del modelo
6. Verificación del programa resultante
7. Diseño de los experimentos (plan de simulación)
8. Ejecución
9. Análisis de reultados



## Tipos de simulación

- ✦ Según la evolución temporal del modelo:
  - ✦ Continua
    - ✦ Conjunto de ecuaciones que se resuelven numéricamente variando el tiempo de forma (casi) continua (a intervalos fijos, pequeños, de tiempo)
  - ✦ Discreta (de eventos discretos)
    - ✦ Conjunto de relaciones lógicas que indican los posibles cambios de estado del sistema. Hay un número finito de cambios posibles y tienen lugar sólo en ciertos instantes de tiempo, no de forma continua.



## Modelo de simulación con eventos discretos

### ✦ Contiene:

- ✦ El estado del sistema
  - ✦ Elementos que lo componen
  - ✦ Atributos de cada elemento
  - ✦ Relación entre elementos
- ✦ Eventos (sucesos) que pueden alterar el estado del sistema a lo largo del tiempo



## Tipos de simulación de eventos discretos

### ✦ Orientada a eventos

- ✦ Enfoque de bajo nivel
- ✦ Un único elemento básico: el evento
- ✦ Conjunto mínimo de primitivas

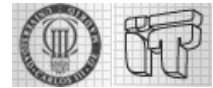
### ✦ Orientada a procesos

- ✦ Mayor nivel de abstracción
- ✦ Elementos básicos: procesos y recursos
- ✦ Conjunto de primitivas más rico



## Simulación orientada a eventos (I)

- ✦ **Cada evento** que pueda cambiar el estado del sistema **se considera por separado**
- ✦ A cada evento se le asocia un **procedimiento de tratamiento**, que agrupa todas las acciones a que dé lugar (incluyendo la programación de otros sucesos en el futuro)
- ✦ El programa principal mantiene una **lista de eventos pendientes** en orden cronológico y va llamando a los procedimientos de tratamiento cuando corresponda



## Simulación orientada a eventos (II)

- ✦ **Conceptos a distinguir:**
  - Tipos de eventos
  - Información asociada a cada evento concreto
  - Procedimiento de tratamiento de cada evento
- ✦ **Primitivas para manejar eventos:**
  - SIMTIME
  - SCHEDULE evento, tipo, info, tiempo
  - CANCEL evento



## Simulación orientada a eventos (III)

### Ejemplo cola

- ✦ Tipos de eventos:
  - llegada, salida
- ✦ Información asociada:
  - Cada llegada pide un cierto tiempo de servicio
- ✦ Procedimientos de tratamiento:
  - Llegada: si el servidor está libre, ocuparlo durante el tiempo de servicio asociado a la llegada. Si no, ponerla en la cola de entrada
  - Salida:...



## Simulación orientada a eventos (IV)

### Ejemplo cola

```
INICIAR
  cola vacía
  servidor libre
  cliente.Ts <- Tservicio()
  SCHEDULE e, llegada, cliente, Tllegada()

TRATAMIENTO DE LLEGADAS (cliente_i)
  IF servidor libre THEN
    servidor ocupado
    SCHEDULE e, salida, -, SIMTIME + cliente_i.Ts
  ELSE
    meter cliente_i en la cola
  END
  cliente.Ts <- Tservicio ()
  SCHEDULE e, llegada, cliente, SIMTIME + Tllegada ()
```



## Simulación orientada a eventos (V) Ejemplo cola

```
TRATAMIENTO DE SALIDAS
  IF cola vacía THEN
    servidor libre
  ELSE
    sacar cliente de la cola
    SCHEDULE e, salida, -, SIMTIME + cliente.Ts
  END
```



## Simulación orientada a procesos (I)

- ✦ Los eventos no se tratan individualmente sino que se consideran dentro del flujo de los procesos del sistema.
- ✦ Un proceso puede estar **ACTIVO**, **EJECUTANDO**, **PASIVO**, **ESPERANDO** o **TERMINADO**
- ✦ El programa principal coordina el avance de los diferentes procesos que intervienen en el sistema





## Simulación orientada a procesos (II)

- ✦ Conceptos a distinguir:
  - ✦ Tipos de procesos
  - ✦ Parámetros y datos de cada ejemplar de proceso
  - ✦ Procedimiento de tratamiento de cada proceso
- ✦ Primitivas para manejar eventos:
  - ✦ SIMTIME
  - ✦ NEWPROCESS proceso, tipo, parámetros
  - ✦ ACTIVATE proceso, retardo
  - ✦ PASSIVATE proceso
  - ✦ KILL proceso
  - ✦ TERMINATE
  - ✦ HOLD retardo



## Simulación orientada a procesos (III)

- ✦ Avance de la simulación:
  - ✦ En cada instante (simulado) se consideran todos los procesos activos
  - ✦ Se ejecuta uno de ellos hasta que pasa a pasivo (PASIVATE o HOLD)
  - ✦ Se van ejecutando sucesivamente
  - ✦ Cuando no quedan más procesos activos, se avanza el tiempo simulado hasta el próximo instante en que un proceso tenga algo que hacer



## Simulación orientada a procesos (IV)

### Ejemplo cola

#### ◆ Tipos de eventos:

- ★ Fuente, servidor

#### ◆ Parámetros:

- ★ Capacidad de servicio de cada servidor

#### ◆ Acciones:

- ★ Servidor: Ver si mi cola está vacía. Si no, sacar el primer cliente y...
- ★ Fuente:...



## Simulación orientada a eventos (V)

### Ejemplo cola

```
INICIAR
  cola vacía
  NEWPROCESS p, servidor
  NEWPROCESS q, fuente
  ACTIVATE p, 0.0
  ACTIVATE q, 0.0

FUENTE
  LOOP
    HOLD Tllegada ()
    cliente.Ts <- Tservicio ()
    meter cliente en la cola
    IF STATUS (p) = Passive THEN
      ACTIVATE p, 0.0
    END
  END
END
```



# Simulación orientada a eventos (VI)

## Ejemplo cola

```
SERVIDOR
  LOOP
    IF cola vacía THEN
      PASSIVATE CURRENT
    ELSE
      sacar cliente de la cola
      HOLD cliente.Ts
    END
  END
END
```



## Resumen

- Simulación orientada a eventos es de más bajo nivel
- La simulación orientada a procesos ofrece abstracciones de más alto nivel para representar los elementos reales del sistema a modelar
- Todas las interacciones entre procesos podrían reducirse a un modelo equivalente basado sólo en sucesos



# Diseño de experimentos

## ✦ Primero debemos:

- ✦ Identificar parámetros relevantes
- ✦ Identificar variables a medir

## ✦ Luego:

- ✦ Generar entrada
  - ✦ Trazas vs generadores (pseudo) aleatorios
- ✦ Medir salida (análisis de resultados)
  - ✦ Eliminar transitorio
  - ✦ Técnicas estadísticas para medir precisión (nivel e intervalo de confianza)



# Temas de investigación

## ✦ Algunos ámbitos de investigación en el terreno de la simulación

- ✦ Modelado de tráfico
- ✦ Optimización de simulaciones
  - ✦ Simulaciones paralelas / distribuidas
  - ✦ Muestreo de importancia



## Realización de simuladores: alternativas

- ✦ Paquetes de simulación: CONMET, OPNET
- ✦ Lenguajes de propósito general + bibliotecas: Sim++, PARSEC
- ✦ Lenguajes específicos de simulación: GPSS, SIMULA, SIMSCRIP, MODSIM
- ✦ A partir de técnicas de descripción formal
- ✦ Simulación de nivel físico (continuas): TOPSIM, MathCad



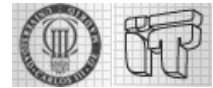
## ¿Qué me dan los lenguajes específicos / las bibliotecas de simulación?

- ✦ Control de la simulación
  - ✦ del paso del tiempo simulado (qué evento / proceso es el siguiente)
- ✦ Generación de números aleatorios
- ✦ Cálculo de estadísticos



## Network simulartor (ns-2)

- ✦ <http://www.isi.edu/nsnam/ns/>
- ✦ Es un simulador de eventos discretos orientado a facilitar la investigación en redes
- ✦ Implementa todos los protocolos estándar desde nivel físico hasta aplicación (TCP, routing, multicast, LAN, WLAN, satélite...)
- ✦ Estos bloques básicos (y otros que nos definamos) implementados en C++
- ✦ La simulación se hace en TCL
- ✦ El resultado se puede ver con nam



## Referencias

- ✦ Law, Averill M. and Law, W. David Kelton: Simulation modeling and analysis, 3rd ed. New York. McGraw-Hill, cop. 2000

