

Introducción a ns-2



Marco A. Alzate

Modelado y Simulación de redes

Diferentes opciones de simulación de redes de comunicaciones



- ⏏ Lenguajes de programación general (Java, Python, C++, matlab)
 - ⏏ Máxima flexibilidad
 - ⏏ Sólo modelos muy pequeños
- ⏏ Paquetes de propósito general (SimEvents)
 - ⏏ Intermedio entre flexibilidad y facilidad de uso
- ⏏ Lenguajes de simulación de propósito específico (ns-2, ns-3)
 - ⏏ Freeware
 - ⏏ Más orientado a la investigación
 - ⏏ Extensa librería de protocolos
- ⏏ Paquetes específicos (QualNet, Comnet, Opnet)
 - ⏏ Gran facilidad de uso
 - ⏏ Animación
 - ⏏ análisis estadístico
 - ⏏ Documentación
 - ⏏ Soporte al cliente

NS-2



⌘ Facilita el diseño y la evaluación de protocolos en Internet

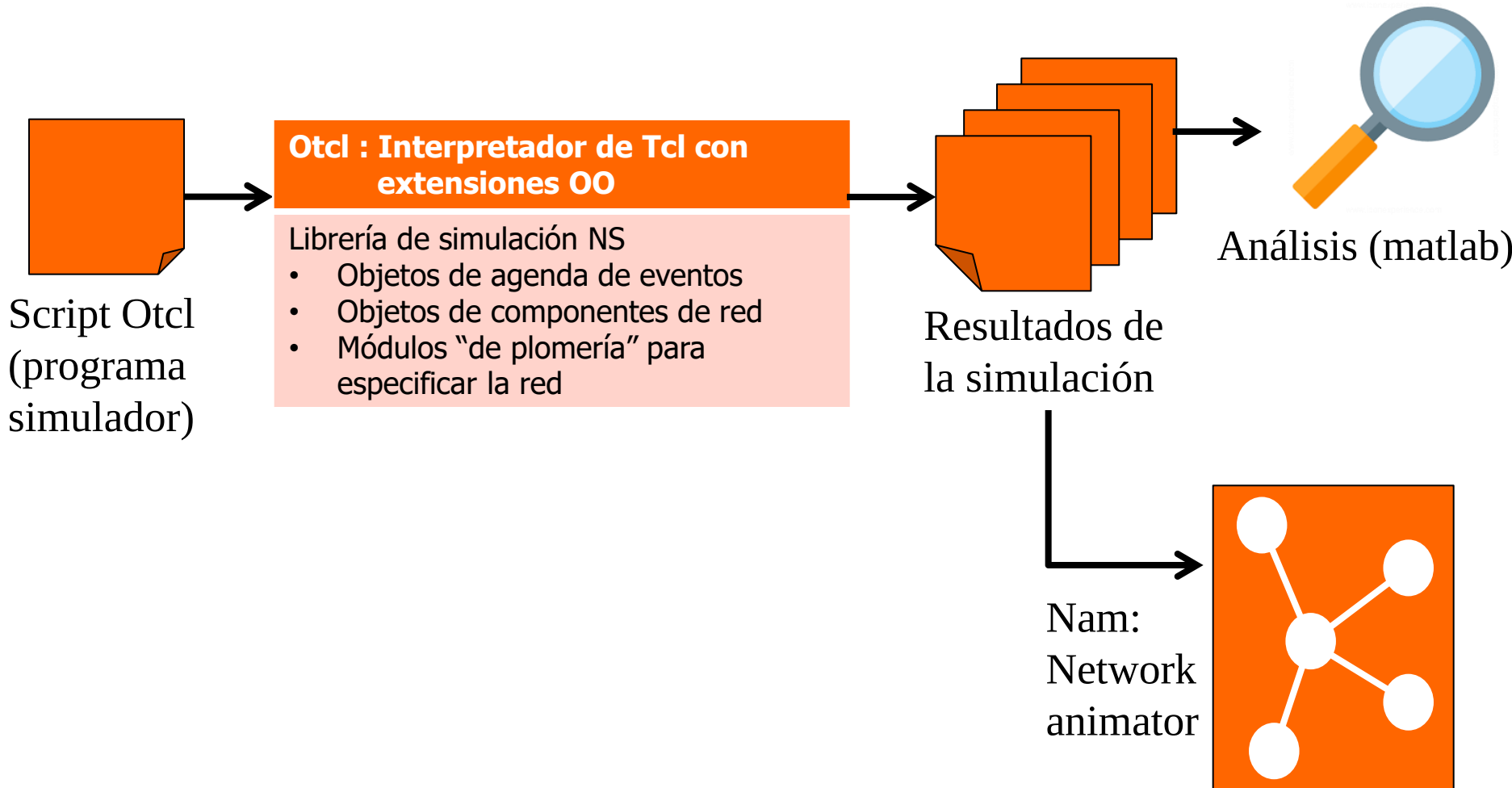
- ☑ Plataforma común, promueve la colaboración
- ☑ University of Southern California, Information Sciences Institute (1989)
- ☑ Proyecto VINT de DARPA (Virtual InterNetwork Testbed) incluyó a USC/ISI, Xerox PARC, LBNL, and UC Berkeley (1996)
- ☑ Yo lo conocí en 2000 y lo use bastante en 2001... ¡Y nunca más!

Instalación



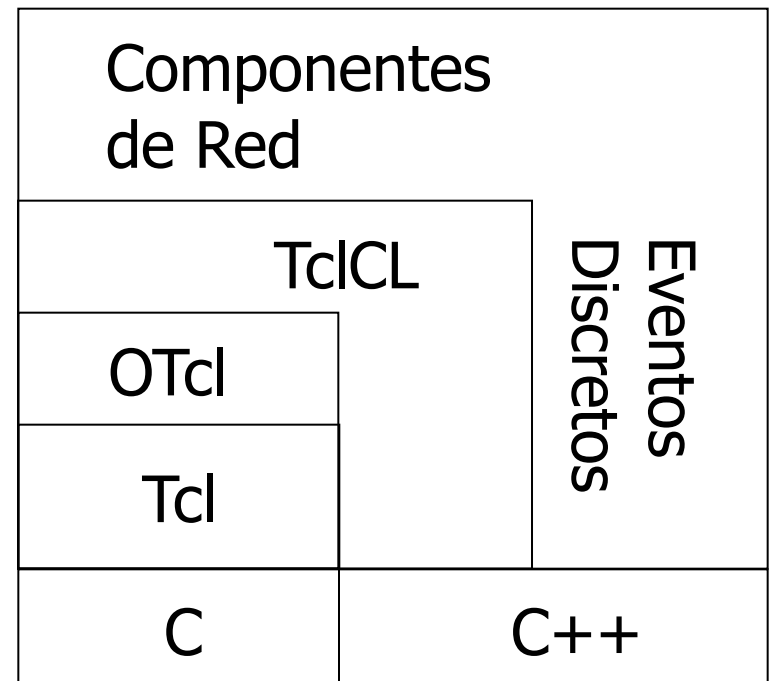
⌘ <http://www.isi.edu/nsnam/ns/ns-build.html>

Arquitectura Básica de NS2



NS es una extensión del interpretador Tcl

- ⌘ Tcl: Lenguaje Script
- ⌘ OTcl: Tcl Orientado a objetos
- ⌘ TclCL: Encadena C++ y OTcl
- ⌘ Control de sucesión de eventos discretos
- ⌘ Componentes de Redes IP (enlace, acceso al medio, enrutamiento, transporte y aplicación)



ns-2

Tcl básico

Variables:

```
set x 10
set x
puts "x is $x"
```

Functions and expressions:

```
set y [pow x 2]
set y [expr x*x]
```

Control flow:

```
if {$x > 0} { return $x } else {
    return [expr -$x] }
while { $x > 0 } {
    puts $x
    incr x -1
}
for {set i 0} {$i<10} {incr i} {
    puts "$\i == $i"
}
```

Procedures:

```
proc pow {x n} {
    if {$n == 1} { return $x }
    set part [pow x [expr $n-1]]
    return [expr $x*$part]
}
```

Files:

```
set file [open "nstrace.txt" w]
set line [gets $file]
puts -nonewline $file "hello!"
close $file
```

Otcl básico

```
Class Person
```

```
# constructor:
```

```
Person instproc init {age} {  
    $self instvar age_  
    set age_ $age  
}
```

```
# method:
```

```
Person instproc greet {} {  
    $self instvar age_  
    puts "age $age_: Hello!"  
}
```

```
# subclass:
```

```
Class Child -superclass Person  
Child instproc greet {} {  
    $self instvar age_  
    puts "age $age_ kid: Wassup!"  
}
```

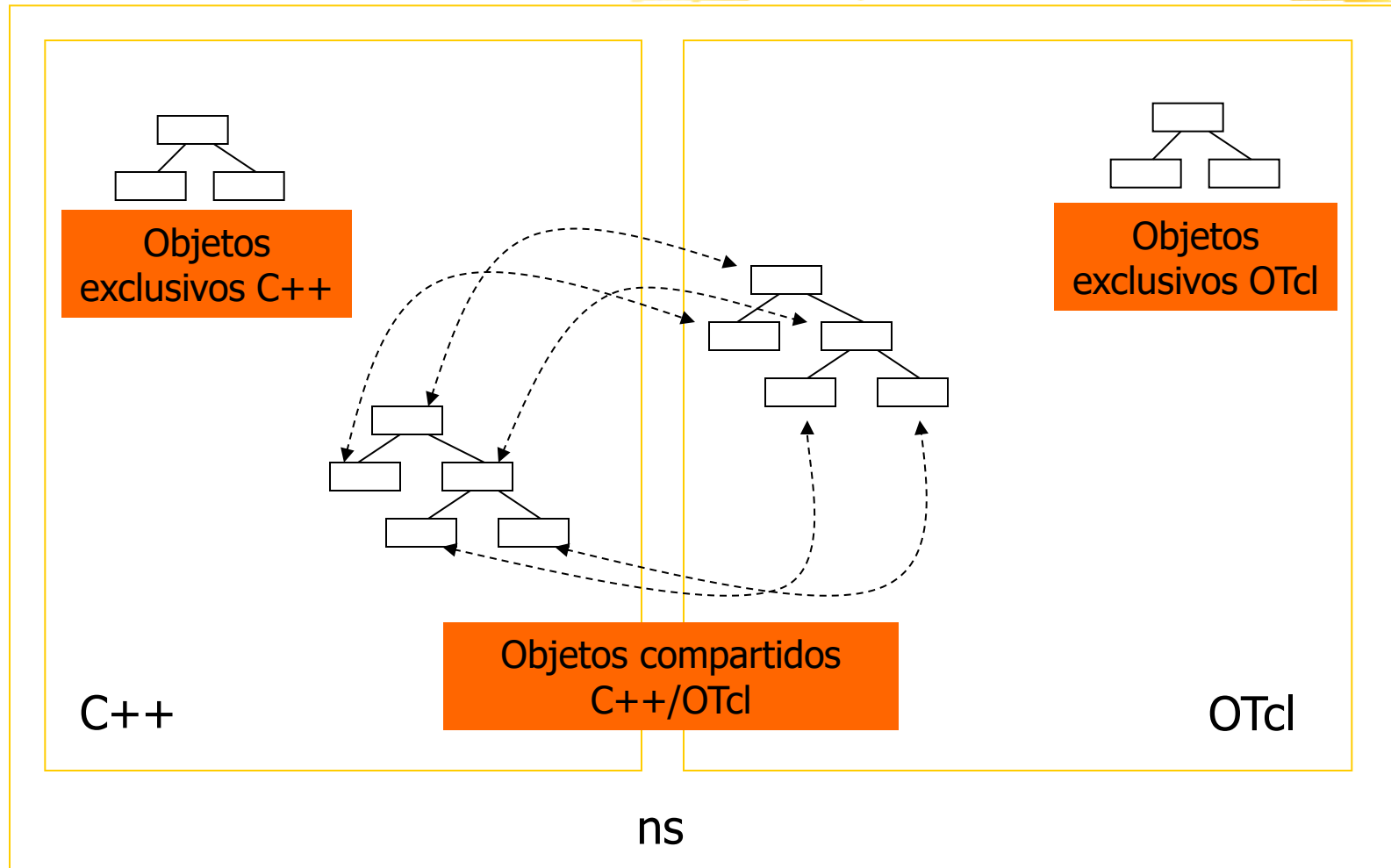
```
set a [new Person 45]
```

```
set b [new Child 15]
```

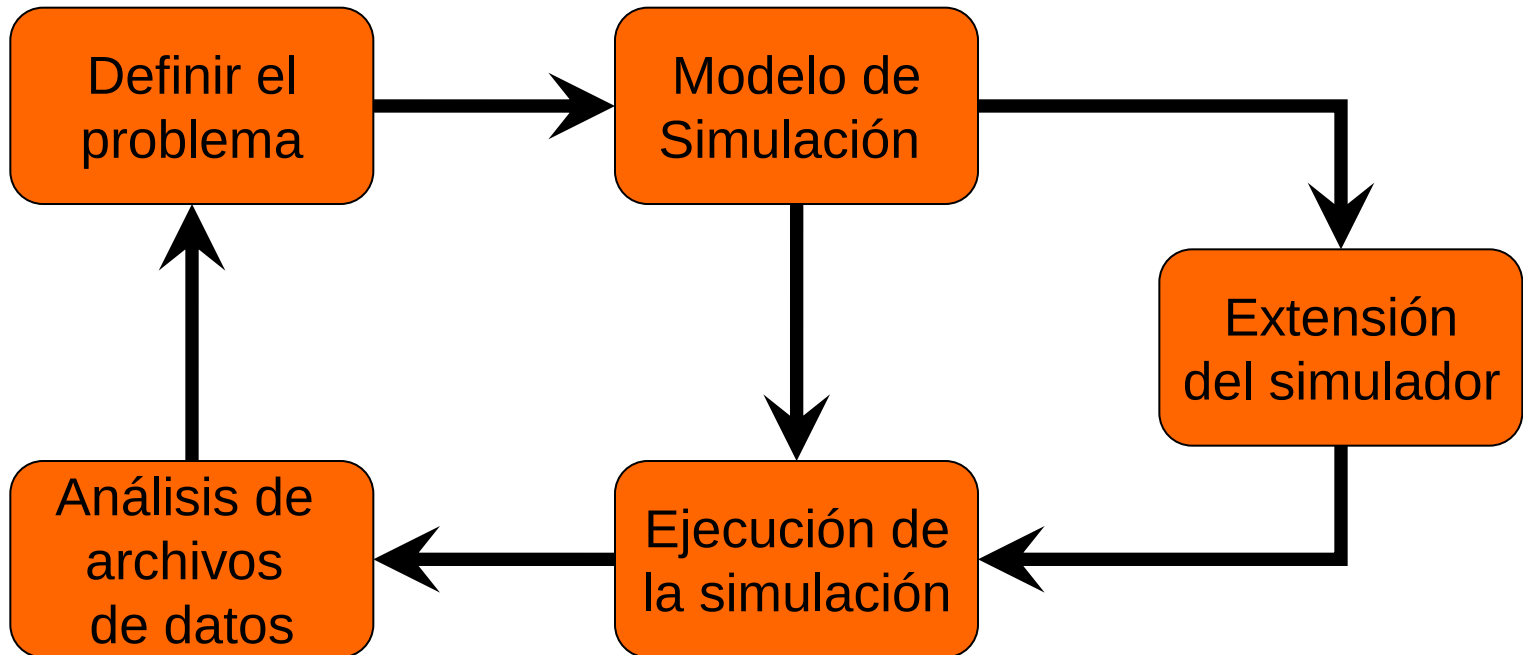
```
$a greet
```

```
$b greet
```

Dualidad entre OTcl y C++



Uso de ns-2



Estructura de Programas NS



- ⌘ Crear el Administrador de Eventos
- ⌘ Habilitar las trazas
- ⌘ Crear la red
- ⌘ Establecer el enrutamiento
- ⌘ Crear los agentes de los protocolos
- ⌘ Generar tráfico
- ⌘ Llamar al administrador de eventos
- ⌘ Hacer reportes

Trazas de la simulación

⌘ Hacer una traza de los paquetes en todos los enlaces

📁 \$ns trace-all [open test.out w]

<evento>	<tiempo>	<paq>	<tamaño>		<fuente>	<destino>	<#sec.>	<attr>
+	1	cbr	210	-----	0.0	3.1	0	0
-	1	cbr	210	-----	0.0	3.1	0	0
r	1.0023	cbr	210	-----	0.0	3.1	0	0

⌘ Trazas para el nam -Animator-

📁 \$ns namtrace-all [open test.nam w]

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id
-------	------	--------------	------------	-------------	-------------	-------	-----	-------------	-------------	------------	-----------

Creación de la Red

⌘ Nodos

- ⊞ set n0 [\$ns node]

- ⊞ set n1 [\$ns node]

⌘ Enlaces

- ⊞ \$ns duplex-link \$n0 \$n1 <Ancho de banda> <Retardo>
<Tipo de cola>

⌘ Conexiones

⊞ UDP

- ⊞ set src [new Agent/UDP]

- ⊞ set rcv [new Agent/Null]

- ⊞ \$ns connect \$src \$rcv

⊞ TCP

- ⊞ set tcp [new Agent/TCP]

- ⊞ set tcpsink [new Agent/TCPSink]

- ⊞ \$ns connect \$tcp \$tcpsink

Creación de la Red



⌘ Tráfico

⌘ FTP

⌘ set ftp [new Application/FTP]

⌘ \$ftp attach-agent \$tcp

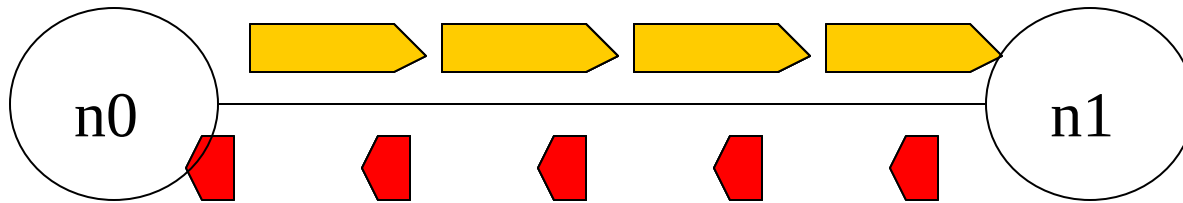
⌘ Telnet

⌘ set telnet [new Application/Telnet]

⌘ Web

⌘ set session [new httpSession \$ns <númPág.>
<NodoCliente>]

Ejemplo Básico



```
set ns [new Simulator]
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1
    1.5Mb 10ms DropTail
set tcp [$ns create-
connection TCP $n0
TCPSink $n1]
```

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.2 "$ftp start"
$ns at 1.2 "exit"
$ns run
```

Otro Ejemplo Básico

#Crea un objeto simulador

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

#Abre una traza de todos los eventos

```
set nf [open out.tr w]
```

```
$ns trace-all $nf
```

#Define el procedimiento de terminar

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exit 0  
}
```

#Crea dos nodos

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

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

#Crea un enlace entre ellos

```
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

#Crea un agente UDP en el nodo n0

```
set udp0 [new Agent/UDP]
```

```
$ns attach-agent $n0 $udp0
```

Create una fuente de tráfico CBR asociado a udp0

```
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 set packetSize_ 500
```

```
$cbr0 set interval_ 0.005
```

```
$cbr0 attach-agent $udp0
```

#Crea un agente nulo en el nodo 1 que consuma el tráfico CBR

```
set null0 [new Agent/Null]
```

```
$ns attach-agent $n1 $null0
```

#Conecta la fuente y el consumidor de tráfico

```
$ns connect $udp0 $null0
```

#Programa los eventos para el agente CBR

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

```
$ns at 4.5 "$cbr0 stop"
```

#Llama al procedimiento finish después de 5 segundos de simulación

```
$ns at 5.0 "finish"
```

#Corre la simulación

```
$ns run
```

Even-to	Hora	Desde	Hacia	Tipo	Tamaño	Ban- deras	ID flujo	fuelle	destino	Num.Se- cuencia	ID pa- quete
+	0.5	0	1	cbr	500	-----	0	0.0	1.0	0	0
-	0.5	0	1	cbr	500	-----	0	0.0	1.0	0	0
+	0.505	0	1	cbr	500	-----	0	0.0	1.0	1	1
-	0.505	0	1	cbr	500	-----	0	0.0	1.0	1	1
+	0.51	0	1	cbr	500	-----	0	0.0	1.0	2	2
-	0.51	0	1	cbr	500	-----	0	0.0	1.0	2	2
r	0.514	0	1	cbr	500	-----	0	0.0	1.0	0	0
+	0.515	0	1	cbr	500	-----	0	0.0	1.0	3	3
-	0.515	0	1	cbr	500	-----	0	0.0	1.0	3	3
r	0.519	0	1	cbr	500	-----	0	0.0	1.0	1	1
+	0.52	0	1	cbr	500	-----	0	0.0	1.0	4	4

Teerawat Issariyakul
Ekram Hossain

Introduction to Network Simulator NS2

Second Edition 2012