

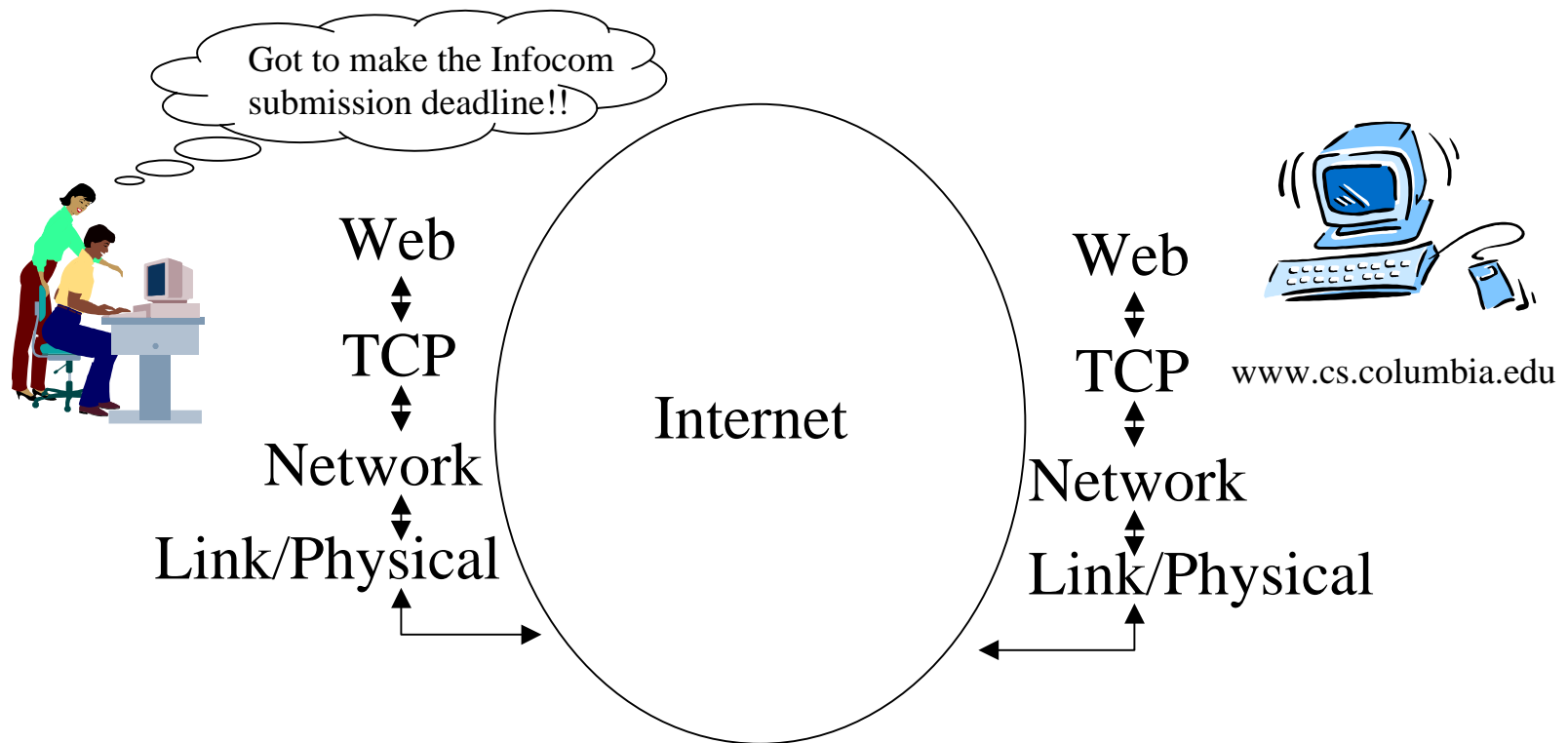
ns-2 101

Polly Huang

USC/ISI

huang@isi.edu

# The Internet



# ns-2

- Network Simulator Version 2 (ns-2)
- Help to debug problems in a controlled environment

# Outlines

- **Essentials**
- Getting Started
- Fundamental tcl, otcl and ns
- Case Studies
  - Web, TCP, Routing, Queuing

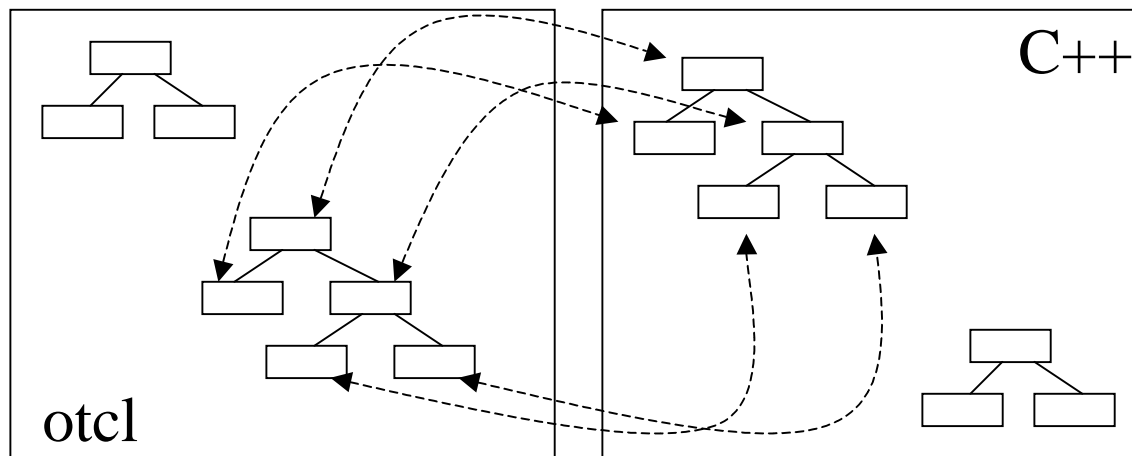
# Object-Oriented

- + Reusability
- + Maintainability
- Careful Planning Ahead

# C++ and otcl Separation

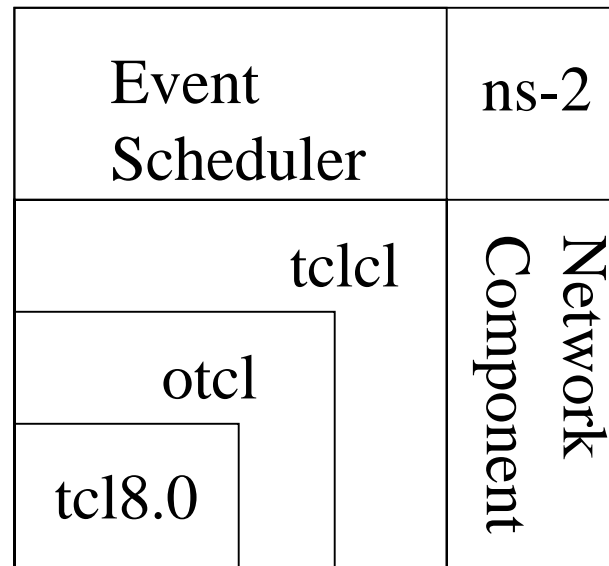
- C++ for data
    - per packet action
  - otcl for control
    - periodic or triggered action
- + Compromise between composibility and speed
- Learning & debugging

# otcl and C++: The Duality



- otcl to call C++ : `command()`, `tcl.result()`
- C++ to call otcl: `tcl.eval()`

# tcl Interpreter With Extents



- otcl: Object-oriented support
- tclcl: C++ and otcl linkage
- Discrete event scheduler
- Data network (the Internet) components



# Outlines

- Essentials
- **Getting Started**
- Fundamental tcl, otcl and ns
- Case Studies
  - Web, TCP, Routing, Queuing

# Installation

- Getting the pieces
  - tcl/tk8.0, otcl, tclcl, ns-2, (and nam-1)
- <http://www-mash.cs.berkeley.edu/ns/ns-build.html>
- ns-users@mash.cs.berkeley.edu
  - majordomo@mash.cs.berkeley.edu
  - subscribe ns-users yourname@address

# Hello World - Interactive Mode

```
swallow 71% ns
```

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

```
_o3
```

```
% $ns at 1 "puts \"Hello World!\""
```

```
1
```

```
% $ns at 1.5 "exit"
```

```
2
```

```
% $ns run
```

```
Hello World!
```

```
swallow 72%
```

# Hello World - Passive Mode

simple.tcl

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

```
$ns at 1 "puts \"Hello World!\""
```

```
$ns at 1.5 "exit"
```

```
$ns run
```

```
swallow 74% ns simple.tcl
```

```
Hello World!
```

```
swallow 75%
```

# Outlines

- Essentials
- Getting Started
- **Fundamental tcl, otcl and ns**
- Case Studies
  - Web, TCP, Routing, Queuing, Wireless

# Fundamentals

- tcl
- otcl
  - <ftp://ftp.tns.lcs.mit.edu/pub/otcl>
- ns-2
  - <http://www-mash.cs.berkeley.edu/ns/ns-man.html>
  - <http://www-mash.cs.berkeley.edu/ns/nsDoc.ps.gz>

# Basic tcl

```
proc test {} {  
    set a 43  
    set b 27  
    set c [expr $a + $b]  
    set d [expr [expr $a - $b] * $c]  
    for {set k 0} {$k < 10} {incr k} {  
        if {$k < 5} {  
            puts "k < 5, pow= [expr pow($d, $k)]"  
        } else {  
            puts "k >= 5, mod= [expr $d % $k]"  
        }  
    }  
}  
  
test
```

# Basic otcl

## **Class** mom

```
mom instproc greet {} {  
    $self instvar age_  
    puts "$age_ years old mom:  
    How are you doing?"  
}
```

```
set a [new mom]  
$a set age_ 45  
set b [new kid]  
$b set age_ 15
```

```
$a greet  
$b greet
```

## **Class** kid -**superclass** mom

```
kid instproc greet {} {  
    $self instvar age_  
    puts "$age_ years old kid:  
    What's up, dude?"  
}
```



# Basic ns-2

- Creating network
- Computing routes
- Creating connection
- Creating traffic
- Inserting errors
- Monitoring

# Creating Network

- Nodes
  - set ns [new Simulator]
  - set n0 [\$ns node]
  - set n1 [\$ns node]
- Links & Queuing
  - \$ns duplex-link \$n0 \$n1 <bandwidth> <delay>  
<queue\_type>
  - <queue\_type>: DropTail, RED, CBQ, WFQ, SFQ, DRR

# Computing routes

- Unicast
  - \$ns rproto <type>
  - <type>: Static, Session, DV, cost, multi-path
- Multicast
  - Simulator set EnableMcast\_ 1
  - Simulator set NumberInterfaces\_ 1
  - \$ns mrtproto <type>
  - <type>: CtrMcast, DM, dynamicDM, pimDM

# Creating Connection

- 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

# Creating Traffic

- FTP
  - set ftp [new Application/FTP]
  - \$ftp attach-agent \$tcp
- Telnet
  - set telnet [new Application/Telnet]
- Web
  - set session [new httpSession \$ns <numPages>  
<clientNode>]

# Inserting Errors

- Creating Error Module
  - set loss\_module [new ErrorModel]
  - \$loss\_module set rate\_ 0.01
  - \$loss\_module unit pkt
  - \$loss\_module ranvar [new RandomVariable/Uniform]
  - \$loss\_module drop-target [new Agent/Null]
- Inserting Error Module
  - \$ns lossmodel \$loss\_module \$n0 \$n1

# Tracing

- Trace packets on all links
  - `$ns trace-all [open test.out w]`

```
<event> <time> <from> <to> <pkt> <size>--<flowid> <src> <dst> <seqno> <aseqno>
+ 1 0 2 cbr 210 ----- 0 0.0 3.1 0 0
- 1 0 2 cbr 210 ----- 0 0.0 3.1 0 0
r 1.00234 0 2 cbr 210 ----- 0 0.0 3.1 0 0
```

- Trace packets on all links in nam-1 format
  - `$ns namtrace-all [open test.nam w]`

# Outlines

- Essentials
- Getting Started
- Fundamental tcl, otcl and ns-2
- **Case Studies**



# Case Studies

- Routing - Multicast
- TCP
- Web
- Queuing - RED

# Visualization Tools

- nam-1 (Network AniMator Version 1)
- xgraph

# Other ns-2 Features

- Mathematical Support
- Multi-access Media (some link layer support)
- Network Dynamics
- Simulation Scale
  - 460 nodes, 100,000 TCP connections
- Emulation Interface

# ns-2 102 Preview

- Flow of code
  - Packet forwarding mechanism walk through
  - Routing walk through
  - TCP walk through
  - Web code walk through
- Creating your own agent (adding & changing)

# Online Resources

- <http://netweb.usc.edu/vint>
- <http://www-mash.cs.berkeley.edu/ns/>
- Mailing lists:
  - [ns-users@mash.cs.berkeley.edu](mailto:ns-users@mash.cs.berkeley.edu)
  - [ns-announce@mash.cs.berkeley.edu](mailto:ns-announce@mash.cs.berkeley.edu)
- To subscribe:
  - [majordomo@mash.cs.berkeley.edu](mailto:majordomo@mash.cs.berkeley.edu)

# Credits

- UCB

- Elan Amir, Hari Balakrishnan, Tom Henderson, Steven McCanne, Giao Nguyen, Venkat Padmanabhan, Teck-Lee Tung

- USC/ISI

- Lars Eggert, Deborah Estrin, Padma Haldar, Mark Handley, John Heidemann, Ahmed Helmy, Polly Huang, Satish Kumar, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Vikram Visweswariah, Ya Xu, Haobo Yu

- LBNL

- Kevin Fall, Sally Floyd

- Xerox Parc

- Sandeep Bajaj, Lee Breslau, Scott Shenker