

An Introduction to NS2

Textbook: T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, Springer 2008.

Outline

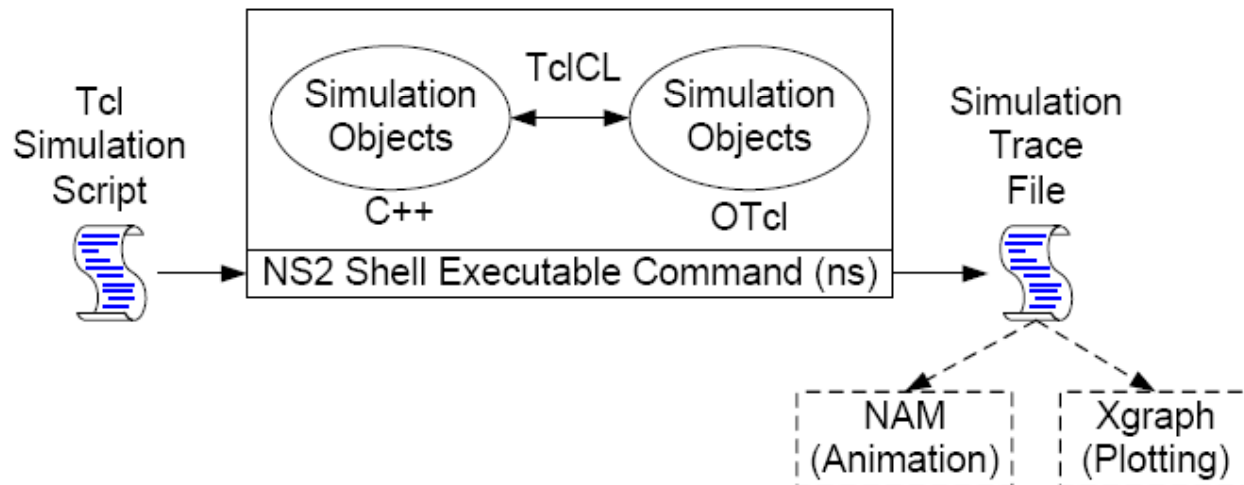
- Overview
- Installation
- An Example
- Incorporate C++ Modules into NS2
- Summary

Overview: Network Simulation

- 3 Simulation Main Steps
 1. Design and Implementation
 - Things to simulate
 - Assumptions
 - Performance measure
 - Code Implementation
 2. Simulation
 - Network Configuration Phase
 - Simulation Phase
 3. Result Compilation
 - Debugging and Tracing
 - Compute performance measures

Overview: NS2 Architecture

- NS2 = Network Simulator 2
- Consists of
 - C++: Internal mechanism
 - OTcl: User interface
 - TclCL: Connecting C++ to OTcl



Overview: NS2 Invocation

- Syntax

```
>> ns [<filename>] [<args>]
```

- No argument: Enter NS2 environment
- <filename> = Tcl simulation script; Use OTcl programming language
- In the Tcl file, <args> is stores in the internal variable (\$argv).

- Example

```
>> ns myfirst_ns.tcl 1 2 3
```

- To retrieve the second input argument (2), execute
(lindex \$argv 1)

The index of the element to retrieve, starting from 0

Outline

- Overview
- Installation
- An Example
- Incorporate C++ Modules into NS2
- Summary

Installation: NS2

1. Go to NS2 web page:

- NS2 Webpage: <http://www.isi.edu/nsnam/ns/>
- Download Link: <http://www.isi.edu/nsnam/ns/ns-build.html#allinone1>.

2. Get all-in-one package

- NS2, Tcl/Tk, OTcl, TclCL
- NAM, Zlib, Xgraph

3. Unzip all the files

4. Use the installation package `./install`

- Follow the instruction
- NS2 is designed for Unix
- For windows, install Cygwin

Note: We will focus text mode only → Not Animation !!

Installation: Cygwin

- Cygwin = Linux emulation for windows
1. Go to Cygwin Webpage: <http://www.cygwin.com/>
 2. Get the package
 3. Install the basic package
 4. Install the following additional packages:

Category	Packages
Development	gcc, gcc-objc, gcc-g++, make
Utils	patch
X11	xorg-x11-base, xorg-x11-devel

Note: You may ignore warning about X11 packages

Installation: Environment

- Add NS2 path and the variable path
- In file `<home>/ .bashrc`, add the following lines

```
PATH=${PATH} :~/ns-new-2.33/ns-2.33
```

```
export PATH
```

```
LD_LIBRARY_PATH=~/ns-new-2.33/otcl-1.13:~/ns-new-2.33/lib
```

```
export LD_LIBRARY_PATH
```

```
TCL_LIBRARY=~/ns-new-2.33/tcl8.4.18
```

```
export TCL_LIBRARY
```

- For convenience,

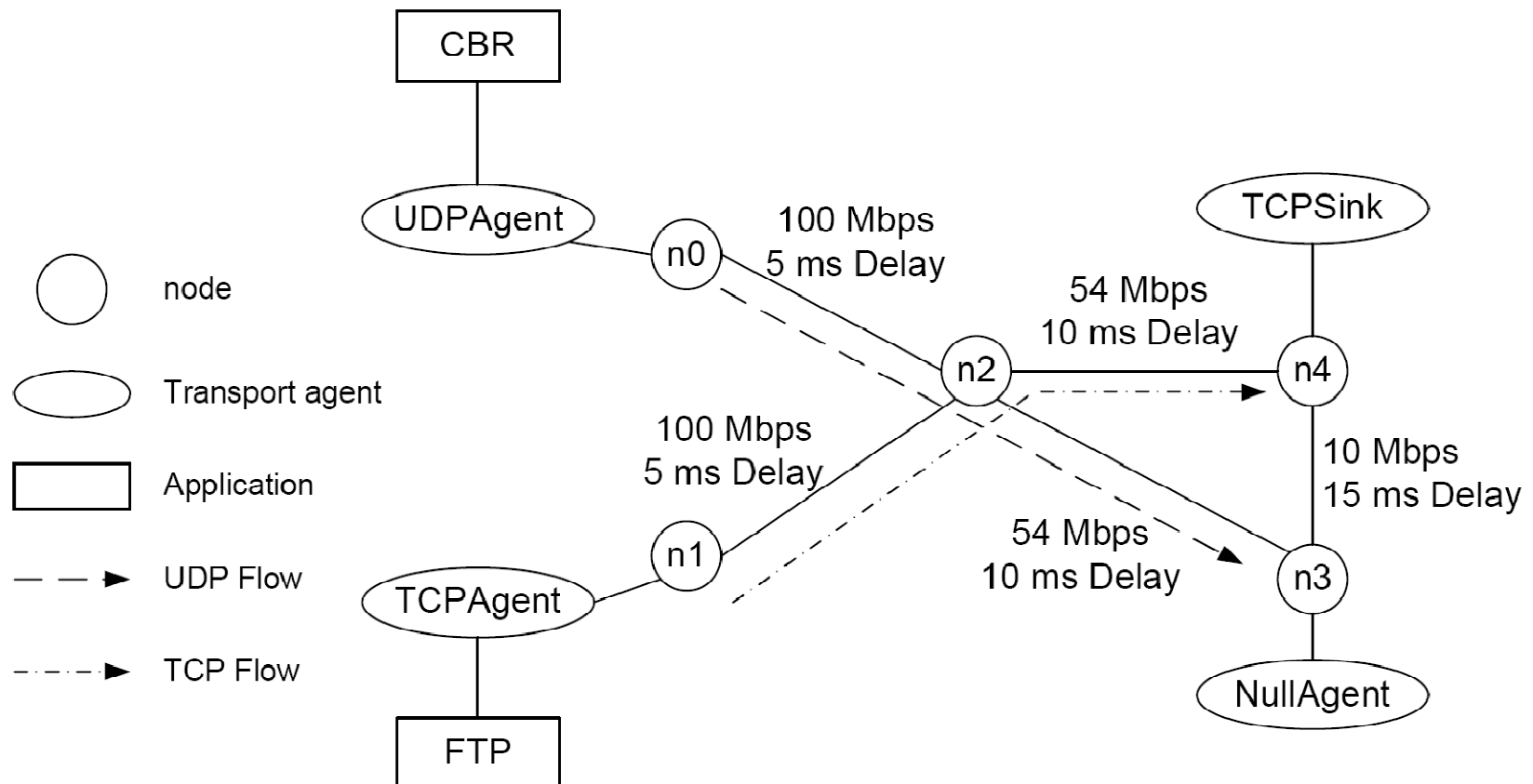
```
cd /usr/local/bin
```

```
ln -s ~/ns-new-2.33/ns-2.33/ns
```

Outline

- Overview
- Installation
- An Example
- Incorporate C++ Modules into NS2
- Summary

A Network Simulation Example



Tcl Simulation Script

- Filename "myfirst_ns.tcl"
- Create a simulator

```
# Create a Simulator  
1 set ns [new Simulator]
```

- Create trace objects

```
# Create a trace file  
2 set mytrace [open out.tr w]  
3 $ns trace-all $mytrace  
# Create a NAM trace file  
4 set myNAM [open out.nam w]  
5 $ns namtrace-all $myNAM
```

A Network Simulation Example

- Define a "finish" procedure

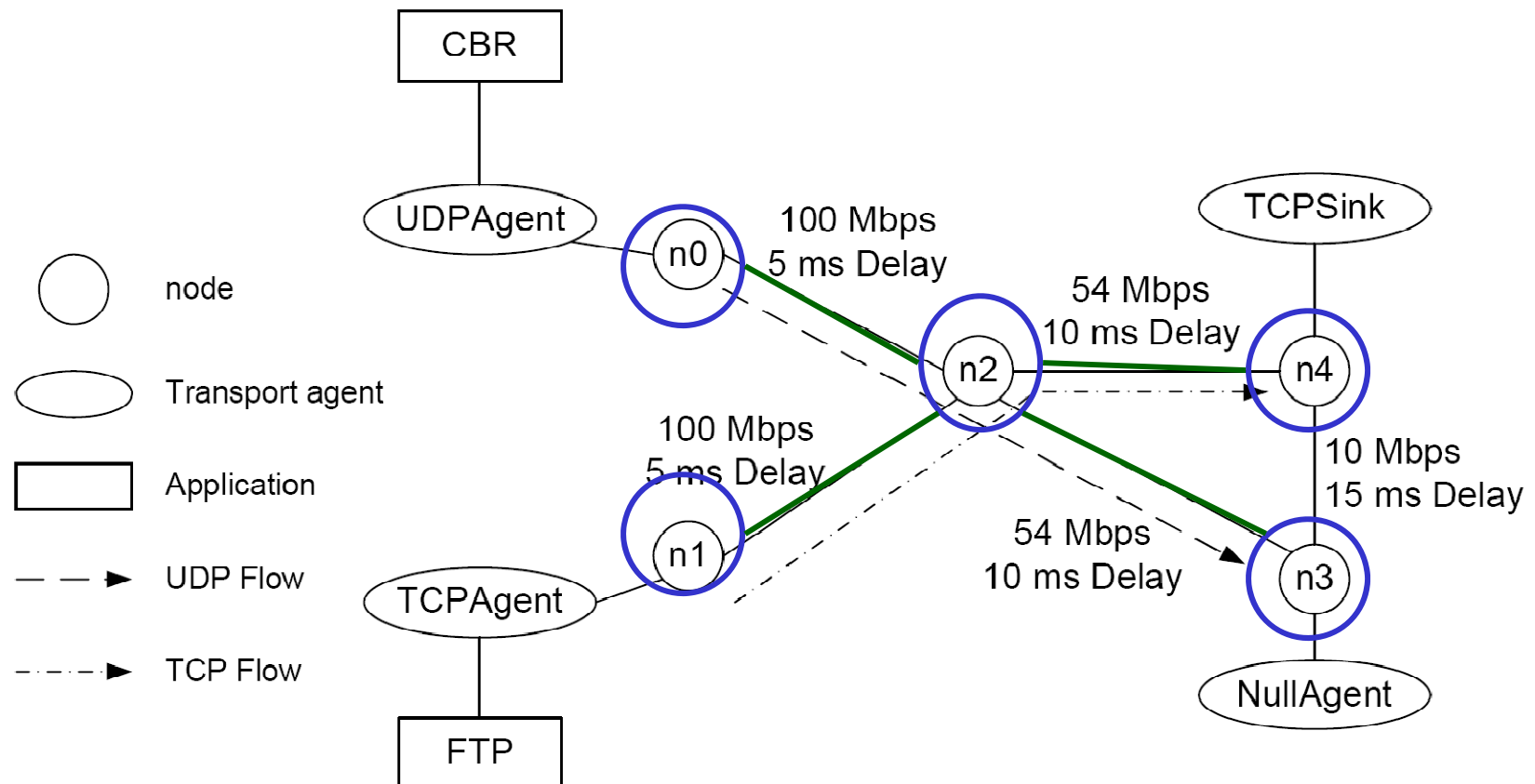
```
# Define a procedure finish
6  proc finish { } {
7      global ns mytrace myNAM
8      $ns flush-trace
9      close $mytrace
10     close $myNAM
11     exec nam out.nam &
12     exit 0
13 }
```

A Network Simulation Example

- Create nodes

```
# Create Nodes
14 set n0 [$ns node]
15 set n1 [$ns node]
16 set n2 [$ns node]
17 set n3 [$ns node]
18 set n4 [$ns node]
```

A Network Simulation Example



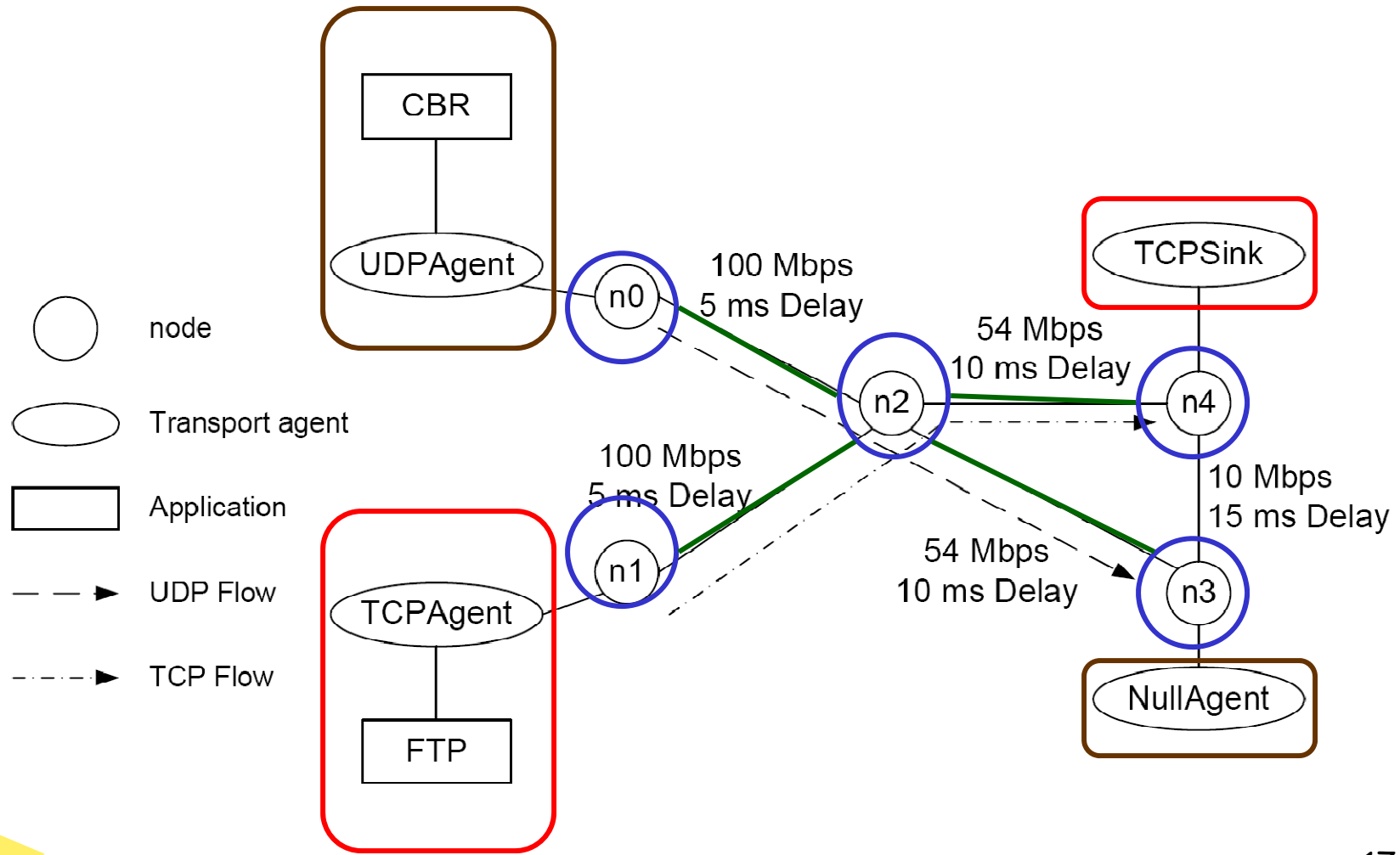
A Network Simulation Example

- Connect node with "duplex" links

```
# Connect Nodes with Links
```

```
19 $ns duplex-link $n0 $n2 100Mb 5ms DropTail
20 $ns duplex-link $n1 $n2 100Mb 5ms DropTail
21 $ns duplex-link $n2 $n4 54Mb 10ms DropTail
22 $ns duplex-link $n2 $n3 54Mb 10ms DropTail
23 $ns simplex-link $n3 $n4 10Mb 15ms DropTail
24 $ns queue-limit $n2 $n3 40
```


A Network Simulation Example



A Network Simulation Example

- Create a flow from n0 to n3

- Create a UDP flow

```
25 set udp [new Agent/UDP]
26 $ns attach-agent $n0 $udp
27 set null [new Agent/Null]
28 $ns attach-agent $n3 $null
29 $ns connect $udp $null
30 $udp set fid_ 1
```

- Attach a CBR source to the UDP flow

```
31 set cbr [new Application/Traffic/CBR]
32 $cbr attach-agent $udp
33 $cbr set packetSize_ 1000
34 $cbr set rate_ 2Mb
```

A Network Simulation Example

- Create a flow from n1 to n4
 - Create a TCP flow

```
35 set tcp [new Agent/TCP]
36 $ns attach-agent $n1 $tcp
37 set sink [new Agent/TCPSink]
38 $ns attach-agent $n4 $sink
39 $ns connect $tcp $sink
40 $tcp set fid_ 2
```

- Attach a FTP source to the TCP flow

```
41 set ftp [new Application/FTP]
42 $ftp attach-agent $tcp
```

A Network Simulation Example

- Schedule Events

```
43 $ns at 0.05 "$ftp start"  
44 $ns at 0.1  "$cbr start"  
45 $ns at 60.0 "$ftp stop"  
46 $ns at 60.5 "$cbr stop"  
47 $ns at 61  "finish"
```

- Start simulation

```
48 $ns run
```

A Network Simulation Example

- We are in step **two** of network simulation

Step 1 = ()

Step 3 = ()

	Sub-Step		The part of the code
1.	()	()	()
2.	()	()	()
3.	()	()	()

A Network Simulation Example

- Results? → ()

```

...
+ 0.110419 1 2 tcp 1040 ----- 2 1.0 4.0 5 12
+ 0.110419 1 2 tcp 1040 ----- 2 1.0 4.0 6 13
- 0.110431 1 2 tcp 1040 ----- 2 1.0 4.0 5 12
- 0.110514 1 2 tcp 1040 ----- 2 1.0 4.0 6 13
r 0.11308 0 2 cbr 1000 ----- 1 0.0 3.0 2 8
+ 0.11308 2 3 cbr 1000 ----- 1 0.0 3.0 2 8
- 0.11308 2 3 cbr 1000 ----- 1 0.0 3.0 2 8
r 0.11316 0 2 cbr 1000 ----- 1 0.0 3.0 3 9
+ 0.11316 2 3 cbr 1000 ----- 1 0.0 3.0 3 9
- 0.113228 2 3 cbr 1000 ----- 1 0.0 3.0 3 9

```

Type Identifier	Time	Source Node	Destination Node	Packet Name	Packet Size	Flags	Flow ID	Source Address	Destination Address	Sequence Number	Packet Unique ID
-----------------	------	-------------	------------------	-------------	-------------	-------	---------	----------------	---------------------	-----------------	------------------

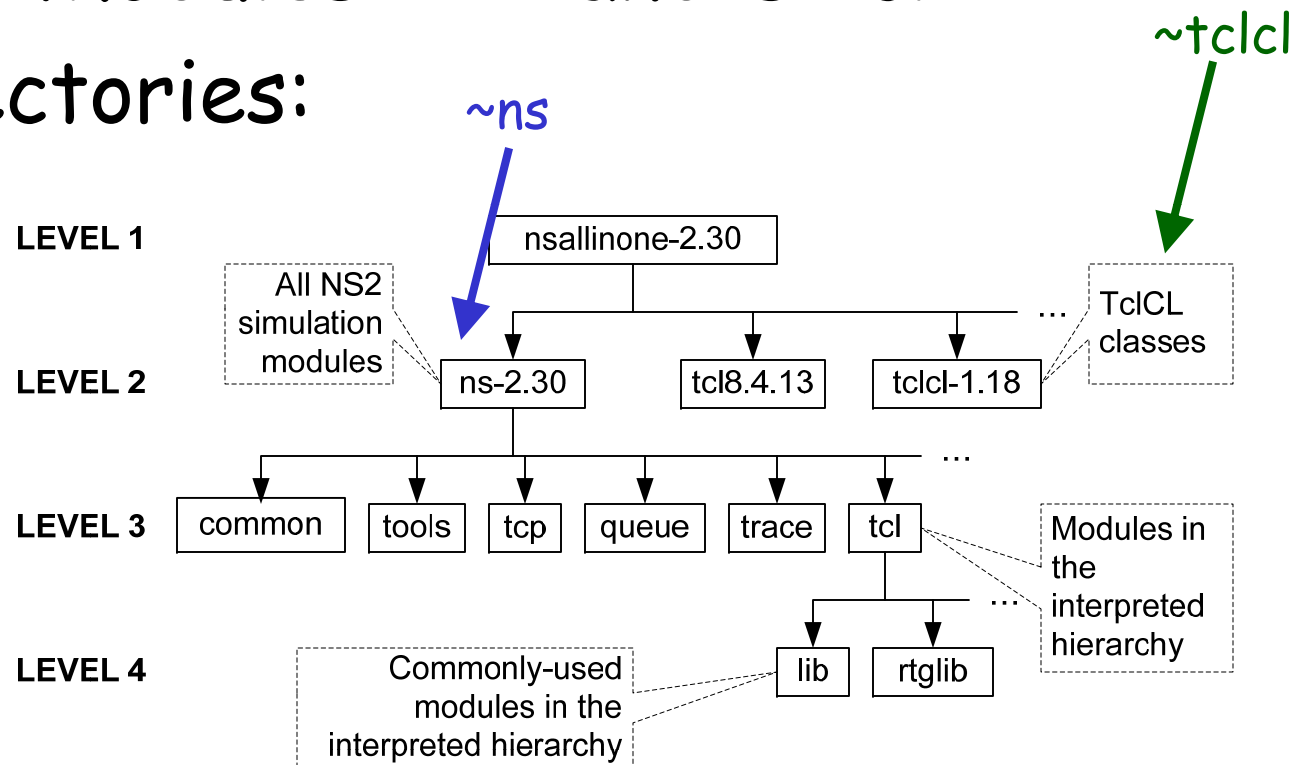
{enqueue(+), deque(-), receive(r), drop(d)}

Outline

- Overview
- Installation
- An Example
- Incorporate C++ Modules into NS2
- Summary

New Modules

- NS2 Modules: C++ and OTcl
- Directories:



See Also: <http://www-rp.lip6.fr/ns-doc/ns226-doc/html/hierarchy.htm>

New Modules

- New C++ or OTcl Modules
 - Need to recompile and links all NEW modules
 - Use `make` utility
 - Take about 5 seconds to run.
- When adding a new OTcl module
 - We can avoid running `make` utility,
 - Define the new module in your simulation script

Make Utility

- “make” contains a set of things that will be done when “make” is executed from the command prompt.
- Make usage
 - >> `make [-f <filename>]`
 - Executed what specified in <filename>
 - <filename> is called a “file descriptor”
 - No file is given → use file “Makefile”
- File descriptor
 - Syntax
 - <target1> [<target2> ...] : [<dep1> <dep2> ...]
 - <command>
 - **Targets** are remade if any of the **dependency** is changed
 - The change is specified in the **command**.

Make Utility

- Example

```
channel : main.o fade.o model.o
        cc -o channel main.o fade.o model.o
```

```
main.o : main.c
        cc -c main.c
```

```
fade.o : fade.c
        cc -c fade.c
```

```
model.o : model.c model.h
        cc -c model.c
```

Execute () if
() is changed

Make Utility

- Example: Use variables OBJS and COM

```
OBJS = main.o fade.o model.o
COM = cc
channel : ${OBJS}
    ${COM} -o channel ${OBJS}

main.o : main.c
    ${COM} -c main.c

fade.o : fade.c
    ${COM} -c fade.c

model.o : model.c model.h
    ${COM} -c model.c
```

Makefile for NS2

- Located in `~ns`
- Key components:
 - `INCLUDES = :` Directory
 - `OBJ_CC = :` All NS2 object files
 - `NS_TCL_LIB = :` All NS2 OTcl files
- Put your files in these three components

Makefile for NS2: Example


```
OBJ_CC = \  
tools/random.o tools/rng.o tools/ranvar.o common/misc.o  
common/scheduler.o common/object.o common/packet.o \  
...
```

Use *.o



```
INCLUDES = \  
-I. \  
-I/home/TB/ns-allinone-2.30/tclcl-1.18 \  
-I./tcp -I./sctp -I./common -I./link -I./queue \  
...
```

Use “\” to
separate lines



```
NS_TCL_LIB = \  
tcl/lib/ns-compat.tcl \  
tcl/lib/ns-default.tcl \  
...
```

Outline

- Overview
- Installation
- An Example
- Incorporate C++ Modules into NS2
- Summary

Summary

- NS2 consists of
 - OTcl () and
 - C++ ()
- Installation: NS2 + Cygwin
- An Example
- Make utility
 - INCLUDE: Directory
 - OBJ_CC: C++ Modules
 - NS_TCL_LIB: OTcl Modules