

Introduction to NS-2

Dr. Donald C. Wunsch II, dwunsch@umr.edu

Dr. Larry Pyeatt, pyeattl@umr.edu

Tae-hyung Kim, tk424@umr.edu

Department of Electrical Computer Engineering

University of Missouri-Rolla, USA

<http://web.umr.edu/~tk424/>

The logo for the University of Missouri-Rolla (UMR) features the letters "UMR" in a bold, white, sans-serif font. The letters are set against a green rectangular background that has a slight 3D effect with a darker green shadow on the right side. The logo is positioned in the bottom left corner of the slide.

UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Agenda

- What/why is Network Simulator-2?
- What can you do with NS-2?
- Installation of NS-2
- Milestone to teach yourself NS-2
- Marc Greis Tutorial I

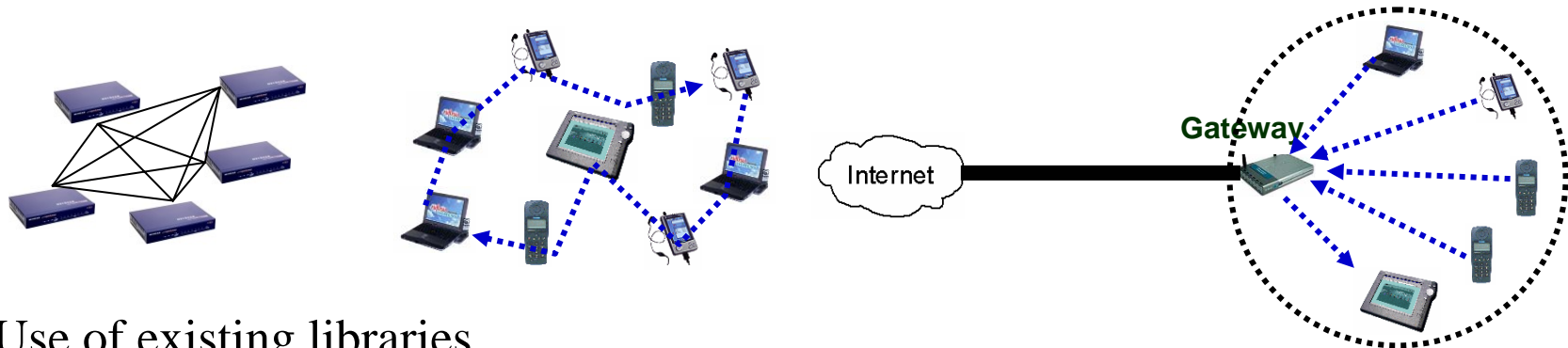
What/why is Network Simulator-2?

- NS (Network Simulator) is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.
- NS-2 is a free computer program with a large pool of libraries, written both in Tcl/OTcl and C/C++, for the purpose of simulating networks. Those libraries include variety of protocols such as TCP, routing, multicast, MAC protocols, and architectures.
- Proper use of existing libraries can save significant amount of time and effort in research. The large pool may help or hurt you.
- “NS-2 is an extremely powerful tool, but unfortunately along with all of the power of the tool comes an extremely steep learning curve. Although there are some tutorials to help get you started, the real knowledge required for doing "real" work (i.e. actually changing ns) will ONLY come with actual, intensive and daily work usage. You have been warned :-)” by Pedro Vale Estrela.

What can we do with Network Simulator-2? (1/3)

Simulating computer networks

- Wired networks
- Wireless networks
 - e.g. MANET (Mobile Ad-hoc NETwork)
- Hybrid networks
 - e.g. home-networks, Wireless Mesh Networks, Delay Tolerant Network



Use of existing libraries

- Demo1
- Demo2
- Demo3
- Tcl-level of NS-2 knowledge is enough to play with existing functionalities.

What can we do with Network Simulator-2? (2/3)

Extending NS-2

- NS-2 is a computer program written in Tcl/OTcl and C/C++, which means you can extend the program as much as you want.
- OTcl, C/C++-level understanding of NS-2 is prerequisite. Understanding OTcl/C++ linkage is also required.

Contributing your effort to NS-2 society

- You can also contribute your codes to the NS-2 society.
- No computational intelligence package!

What can we do with Network Simulator-2? (3/3)

• Limitation of NS-2

- NS-2 is a “simulator” of which model is necessarily a simplification of the real-world system.

e.g. The simulator model for Reno, NewReno, Sack, and Fack TCP are supported; however, there is no dynamic window advertisement, SYN/FIN connection establishment/teardown. Segment and ACK number computations are in units of packets.

- NS-2 is not a polished and finished product, rather the result of an on-going effort of research and development. Bugs in the software are still being discovered and corrected. Therefore, it is YOUR responsibility to validate correctness of the simulation results.

e.g. EDCF semi-package for 802.11e has/had timing bugs for 802.11a PHY.
(Atheros EDCF semi-package ns-06010 is developed from ns-2.1b7snapshot 20000808)

- Nevertheless, NS-2 is quite popular; more than hundreds of papers using NS-2 for research can be found in IEEE Xplore.

Installing NS-2 on the Linux (1/8)

Requirement for installing NS-2

- a Unix mounted computer, a C++ compiler, 320MB space.
- NS-2 is developed on several kinds of Unix such as FreeBSD, Linux, and Solaris.
- You can run NS-2 on windows if Linux emulator, cygwin, is installed.
- Cygwin is a Linux-like environment for Windows. It consists of two parts: A DLL (cygwin1.dll) which acts as a Linux API emulation layer providing substantial Linux API functionality. A collection of tools, which provide Linux look and feel.
- The cygwin install is still considered EXPERIMENTAL.

There are two ways to install NS-2, suggested by NS development team.

- Install from the pieces.
- Install from the all-in-one package.

NS-2 installation with all-in-one package.

- We will install ns-allinone-2.29 onto a Debian Linux machine.
- To install NS-2 on Windows+cygwin, please refer to

[http://nsnam.isi.edu/nsnam/index.php/Running Ns and Nam Under Windows 9x/2000/XI Using Cygwin](http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XI_Using_Cygwin)

Installing NS-2 on the Linux (2/8)

- Installing ns-allinone-2.29 onto a Debian Linux machine.

We'll use remote-terminal client programs to access the Linux workstation. The installation procedure is identical when you install on your own Linux machine except you don't use the client programs such as putty, X-Win32, and WinSCP3.

- step 1. download ns-allinone package from [the official NS-2 website](#).
- step 2. log into the Linux server, Wunsch-Linux.ece.UMR.edu, using putty+X-Win32.
- step 3. upload the file to your home directory, */home/ns2userName*, on the server, using a FTP client such as WinSCP3.
- step 4. untar (uncompress) the uploaded file
\$ tar xzvf filename
- step 5. Change directory to the *ns-allinone* directory, run the installation script, and wait until the installation is completed and successful. If the installation is not successful, you should ask the ns-mailinglist or fix the bug by yourself.

\$ cd ~/ns-allinone-2.29

\$./install

```
Ns-allinone package has been installed successfully.
here are the installation places:
tcl8.4.11:    /home/ns2root/ns-allinone-2.29/{bin,include,lib}
tk8.4.11:    /home/ns2root/ns-allinone-2.29/{bin,include,lib}
otcl:        /home/ns2root/ns-allinone-2.29/otcl-1.11
tclcl:       /home/ns2root/ns-allinone-2.29/tclcl-1.17
ns:          /home/ns2root/ns-allinone-2.29/ns-2.29/ns
nam:         /home/ns2root/ns-allinone-2.29/nam-1.11/nam
xgraph:      /home/ns2root/ns-allinone-2.29/xgraph-12.1
gt-itm:      /home/ns2root/ns-allinone-2.29/itm, edriver, sgb2alt, sgb2ns, sgb2comm
s, sgb2hierns
```

Installing NS-2 on the Linux (3/8)

- step 6. Configure your *.bashrc* file in the home directory to use NS-2.
.bashrc is a configuration file for bash where rc stands for “run commands”
\$ cd ~ or *cd /home/ns2userName*
\$ gedit .bashrc &
- To configure your *.bashrc* file, go to the last line of the *.bashrc* and add the path of the NS-2 executable *ns* and *tcl*, *LD_LIBRARY_PATH*, and *TCL_LIBRARY*.
For example,

...In your .bashrc

ns-allinone-2.29. This line is a comment using the # character.

export PATH="\$PATH:/home/ns2userName/ns-allinone-2.29/bin:/home
/ns2userName/ns-allinone-2.29/tcl 8.4.11/unix:/home/ns2userName/ns-allinone-
2.29/tk8.4.11/unix"

export LD_LIBRARY_PATH="/home/ns2userName/ns-allinone-2.29/otcl-1.11,
/home/ns2userName/ns-alli none-2.29/lib"

export TCL_ LIBRARY="/home/ns2userName/ns-allinone-2.29/tcl8.4.11/library"

Installing NS-2 on the Linux (4/8)

- The precise directories are found at the end of the installation messages.

```
ns2root@wunsch-linux: /home/ns2root/ns-allinone-2.29

Please put /home/ns2root/ns-allinone-2.29/bin:/home/ns2root/ns-allinone-2.29/tcl8.4.11/unix:/home/ns2root/ns-allinone-2.29/tk8.4.11/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/ns2root/ns-allinone-2.29/otcl-1.11, /home/ns2root/ns-allinone-2.29/lib,
into your LD_LIBRARY_PATH environment variable.
If it complains about X libraries, add path to your X libraries
into LD_LIBRARY_PATH.
If you are using csh, you can set it like:
    setenv LD_LIBRARY_PATH <paths>
If you are using sh, you can set it like:
    export LD_LIBRARY_PATH=<paths>

(2) You MUST put /home/ns2root/ns-allinone-2.29/tcl8.4.11/library into your TCL_
LIBRARY environmental
variable. Otherwise ns/nam will complain during startup.

(3) [OPTIONAL] To save disk space, you can now delete directories tcl8.4.11
and tk8.4.11. They are now installed under /home/ns2root/ns-allinone-2.29/{b
in,include,lib}

After these steps, you can now run the ns validation suite with
cd ns-2.29; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list ar
chive
for related posts.

ns2root@wunsch-linux:~/ns-allinone-2.29$
```

Installing NS-2 on the Linux (5/8)

- step 7. Exit the current terminal, run a new terminal, and check if the new .bashrc works.

```
$ exit
```

```
$ ns ;# At the home directory in a new terminal
```

```
%
```

If the path setup is correct, you'll see %. Otherwise, *bash: ns: command not found*

- step 8. (optional) Run an example Tcl script with X-Win32 turned on to test if NS works properly.

```
$ cd ~/ns-allinone-2.29/ns-2.29/tcl/ex
```

```
$ ns simple.tcl
```

```
210
```

```
0.0037499999999999999
```

```
running nam...
```

- step 9. (optional) Go to the NS directory and validate the installation.

Be aware that this validation process takes much longer than installation.

```
$ cd ~/ns-allinone-2.29/ns-2.29
```

```
$ ./validate
```

Installing NS-2 on the Linux (6/8)

```
ns2root@wunsch-linux: /home/ns2root/ns-allinone-2.29/ns-2.29
<![10.000224](node 11) no coordinator found for association.
[10.319616](node 7) beacon transmission successful [channel:11] [PAN_ID:0]
[10.364736](node 2) beacon transmission successful [channel:11] [PAN_ID:0]
[10.373696](node 16) beacon transmission stopped [channel:11] [PAN_ID:0]
[10.374848](node 0) beacon transmission stopped [channel:11] [PAN_ID:0]
--- startDevice [11] ---
[11.000224](node 11) performing active channel scan ...
[11.789664](node 11) sending association request to [channel:11] [PAN_ID:0] [Coo
[12.292672](node 11) association successful (non-beacon enabled) [channel:11] [P
[12.292672](node 11) begin to transmit beacons
[12.293440](node 11) beacon transmission successful [channel:11] [PAN_ID:0]
[15.033408](node 4) synchronization loss
[15.033408](node 3) synchronization loss
[16.517184](node 3) coordinator relocation failed --> try to reassociate ...
--- startDevice [3] ---
[16.517184](node 3) performing active channel scan ...
[16.517952](node 4) coordinator relocation failed --> try to reassociate ...
--- startDevice [4] ---
[16.517952](node 4) performing active channel scan ...
[17.305984](node 3) sending association request to [channel:11] [PAN_ID:0] [Coor
[17.306752](node 4) sending association request to [channel:11] [PAN_ID:0] [Coor
[17.802720](node 3) association successful (non-beacon enabled) [channel:11] [PA
[17.805088](node 4) association successful (non-beacon enabled) [channel:11] [PA
Test output agrees with reference output
All test output agrees with reference output.
Mon Jan  9 05:00:32 CST 2006
These messages are NOT errors and can be ignored:
    warning: using backward compatibility mode
    This test is not implemented in backward compatibility mode
validate overall report: all tests passed
[2]+  Done                  gedit .bashrc  (wd: ~)
(wd now: ~/ns-allinone-2.29/ns-2.29)
ns2root@wunsch-linux:~/ns-allinone-2.29/ns-2.29$
```

Installing NS-2 on the Linux (7/8)

More on *.bashrc* setup for NS-2

- Bash shell is the most popular shell and *.bashrc* is a configuration file for bash. *.bashrc* follows the syntax for (bash) shell script. Even if the 1st line of a shell script starts from *#!/bin/bash*, *.bashrc* does not start from the line.
- The variable information the shell maintains is called the shell's environment where variables are chunks of computer memory that hold changeable values.
- *export variable* is a command to make a variable an environment variable, which is global to shell and subshells. *var=value* defines a variable. For example, *export name="kuna"* assigns *kuna* to the variable *name* and make the variable an environment variable. Note: no space around the equal sign “=“.
- *PATH* is one of the standard shell variables, holding the search path for commands. The list of directories is separated by colon “:”.
- Bash uses the *\$* character that substitution needs to be made. “*\$*” is used in the script to access the value of a variable. Therefore *\$PATH* returns the value in the environment variable *PATH*. You can see the current value of *PATH*. Try, *\$ echo \$PATH*

Installing NS-2 on the Linux (8/8)

- The # character comments the rest of the line.
- `LD_LIBRARY_PATH` and `TCL_LIBRARY` are environment variables showing the paths to the libraries. Execution of NS requires the libraries.
- The `.bashrc` configuration for NS-2 revisited. In `.bashrc`, the following lines should be added, typically, at the end of the file.

```
# ns-allinone-2.29. This line is a comment using the # character. export
PATH="$PATH:/home/ns2userName/ns-allinone-2.29/bin:/home/ns2userName/ns-
allinone-2.29/tcl 8.4.11/unix:/home/ns2userName/ns-allinone-2.29/tk8.4.11/unix"
export LD_LIBRARY_PATH="/home/ns2userName/ns-allinone-2.29/otcl-1.11,
/home/ns2userName/ns-allinone-2.29/lib"
export TCL_LIBRARY="/home/ns2userName/ns-allinone-2.29/tcl8.4.11/library"
```

Q: What's gonna happen if a path `LD_LIBRARY_PATH` isn't set up properly?

A: If you run NS by typing, `$ ns <tcl_script>`, you'll see an error message such as

`$ ns :error in loading shared libraries : libtk8.3.so : can not open shared`

`$ object file : no suchfile or directory`

From <http://mailman.isi.edu/pipermail/ns-users/2000-November.txt>

Installing NS-2 on Windows+cygwin (1/4)

- Since NS on Windows+cygwin is still considered experimental, installing NS on the Linux is preferred. However, it is sometimes useful to use NS on Windows+cygwin. For example, demonstration for presentations.
- Cygwin is a Linux emulator providing Linux-like environment for Windows. Therefore, installation process/use of NS on cygwin is similar to the Linux environment. This implies that the NS user should eventually familiarize him/herself to the Linux environment.
- An excellent guideline to install NS-2 on Windows+cygwin is available at, [http://nsnam.isi.edu/nsnam/index.php/Running Ns and Nam Under Windows 9x/2000/XP Using Cygwin](http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin)
- The summary of the above installation process on cygwin is two-fold:
 - (1) configure cygwin for NS installation,
See “Requirements and Installation Tips”
 - (2) install NS like on the Linux
See “Building NS-allinone Under Cygwin”

Installing NS-2 on Windows+cygwin (2/4)

(1) configure cygwin for NS installation,

- Make sure you have installed cygwin with the UNIX text type.

\$ mount/ grep textmode

You should be fine if you don't see anything.

- Make sure your cygwin installation directory does not contain any spaces.

The default setup, *c:/cygwin*, is a good choice.

- Make sure you don't have spaces in your login name.

- Install the necessary packages for NS installation.

1. X11: either Xfree86 or X.org

- Xfree86: *XFree86-base, XFree86-bin, XFree86-prog, XFree86-lib, XFree86-etc*

- X.org: *xorg-x11-bin, xorg-x11-bin-dlls, xorg-x11-devel, xorg-x11-libs-data, xorg-x11-etc*

2. GNU Compiler Collection: *gcc, gcc-g++*

3. GNU awk: *gawk*

4. Compression utility programs: *tar, gzip*

5. Other required programs: *make, patch, perl*

6. (Opt) A text editor: *nedit*

Installing NS-2 on Windows+cygwin (3/4)

Note: Make sure you don't miss any packages required for NS installation.

```
~/ns-allinone-2.29
tk424@wunsch-laptop4 ~/ns-allinone-2.29
$ ./install
=====
* Testing for Cygwin environment
=====
Cygwin detected
Note: Cygwin install is still considered EXPERIMENTAL

Checking Cygwin version is >= 1.3.12... 1.5.18 (should be ok)
Checking filesystems are mounted as UNIX filetype... yes
Checking default mode is binmode... yes
Checking legitimate login name... ok
Checking legitimate path name... ok
Checking for gcc... ok
Checking for gcc-g++... ok
Checking for gawk... ok
Checking for tar... ok
Checking for gzip... ok
Checking for make... ok
Checking for patch... ok
Checking for perl... ok
Checking for w32api... ok
Checking for diff... ok
Checking for X... X.org
Checking for xorg-x11-bin... ok
Checking for xorg-x11-bin-dlls... ok
Checking for xorg-x11-devel... ok
Checking for xorg-x11-libs-data... ok
Checking for xorg-x11-etc... ok
Patching Tcl/Tk for Cygwin.
The original tcl8.4.11/generic/tcl.h is backed up as tcl8.4.11/generic/tcl.h.orig
The original tk8.4.11/generic/default.h is backed up as tk8.4.11/generic/default.h.orig
Patching sgb for Cygwin.
The original sgb/Makefile is backed up as sgb/Makefile.orig
Setting executable format to .exe...
=====
* Build XGraph-12.1
=====
```

Installing NS-2 on Windows+cygwin (4/4)

(2) install NS like on the Linux

The cygwin environment is like Linux. Step 5 in the slide “Installation of NS-2 (2/8)” is your next step if we keep the consistency of the slides. In short,

```
$ cd ~/ns-allinone-2.29
```

```
$ ./install
```

The options of commands may differ from a shell to a shell. Therefore, the option of a command for Linux can be different from that for cygwin. For example,

Linux: *\$ tar xvfz filename*

cygwin: *\$ tar -zxvf filename*

Milestone to teach yourself NS-2 (1/2)

Part I

- Installing NS-2.
- Running example scripts.
“Marc Greis tutorial,” <http://www.isi.edu/nsnam/ns/tutorial/>
- Learning Tcl.
- Install tcl-debug
- Tcl level debugging.

Part II

- Learning OTcl.
- C++ level debugging in NS-2.
- OTcl/C++ binding.
- Customize simulations with scripting (awk, bash shell).
- C++ class hierarchy in NS-2.
- Read the main manual for NS-2

Part III

- Write your own code. You are more than ready to go.



UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Milestone to teach yourself NS-2 (2/2)

Working directory

- Make a directory *work_files* under your home directory

\$ cd ~ or cd /home/ns2userName

\$ mkdir work_files

\$ cd work_files

\$ mkdir Part1

\$ mkdir Part2

\$ mkdir Part3

- This working directory will be denoted *~/work_files*, because *~/* denotes the home directory of the user. (Try *cd ~*)
- Please organize your working files with proper directory structure.

Practice

- You can use any text editor. I recommend *gedit* or *nedit*, if these are installed.
\$ gedit & or \$ nedit &
- When you learn NS-2 codes, I recommend to type each line with a text editor

Marc Greis Tutorial I: Table of Contents (1/2)

Why Marc Greis tutorial? <http://www.isi.edu/nsnam/ns/tutorial/index.html>

- Marc Greis' *ns* tutorial is an invaluable introduction. It's now maintained by the *ns* developers. – Lloyd Wood
- This is the classic and most important tutorial. – Pedro Vale Estrela
- Sandeep Gupta NS tutorial, <http://wcc.iiita.ac.in/ns/nstutorial.pdf>, is cross-referenced .

Contents

IV. The first Tcl script, *example1.tcl*

V. Making it more interesting, *example2.tcl*

VI. Network dynamics, *example3.tcl*

VII. A new protocol for ns (We'll cover this in Part II: lecture)

VIII. Creating Output files for Xgraph, *example4.tcl*

IX. Running wireless simulations in ns

X. Creating Wired-cum-Wireless and MobileIP Simulations

XI. Generating traffic-connection and node-movement files for large wireless scenarios (We don't need this right away.)

Note: These slides are summary of Marc Greis tutorial and Sandeep Gupta NS tutorial. The copyright belongs to the original authors.

Marc Greis Tutorial I: Table of Contents (2/2)

- We will learn from the Marc Greis tutorial as the class time allows. We will come back to this tutorial after learning the basics of Tcl/OTcl language. Then, you can have better understanding of the NS Tcl scripts. We can distinguish the pure Tcl/OTcl commands from the NS Tcl/OTcl commands.

- Running Tcl scripts

Type in the following command to run a Tcl script in the directory where the Tcl script is located.

\$ ns tclScriptName

For example, type in *\$ ns example1.tcl* in the directory where *example1.tcl* is located to run *example1.tcl*.

- IV. The first Tcl script, *example1.tcl*

To understand *example1.tcl* step-by-step, please run *example1.tcl*, and see what happens. You may run one Tcl line after another. You may run *example1_1.tcl*, *example1_2.tcl*, *example1_3.tcl* and read the Tcl codes.

Conclusions

- We've defined NS (Network Simulator) as a discrete event simulator for networking research. Another interpretation of NS is a computer program with a large pool of built-in libraries written in Tcl/OTcl and C/C++, as a result of research effort by other researchers. Therefore, we can save time/effort thanks to other researchers' effort. ☺
- We've learnt how to install NS-2, release 2.29 on Linux. Now, you are ready to "rock".
- NS-2 requires knowledge on Linux commands, Tcl/OTcl, C/C++, OTcl/C++ binding, scripting (Bash shell, awk and/or perl), and so on.
- We've also seen the milestone to teach yourself NS-2. We will build up the required knowledge by the "milestone".

The next Class

- The next topic is “Introduction to TCL (Tool Command Language).”
 - Installation of TclTutor2.0, an interactive Tcl learning computer program by Clif Flynt
 - Learning Tcl using TclTutor2.0
- Assignment 2 is coming at the end of lecture 3. Make sure to install NS-2.



UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Thank you.



UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.