Pearls in Life

Technical reflections and records of thoughts

- RSS

Navigate...

- Home
- Projects
- Be Productive!
-
- Blog
- Archives

Simulate Random MAC Protocol in NS2 (Part II)

Dec 15th, 2013

In previous post, we wrote an NS2 simulation program that fits the project specification, except that we're using the standard 802.11 MAC protocol. In this post, we'll discuss how to add our own MAC protocol to NS2.

Compile NS2 from Source

To add an new protocol to NS2, we actually need to download the whole NS2 source tree and add some extra CPP files there, which is embarrassingly inconvenient. But for now, we have to live with it.

Anyways, download the NS2 all-in-one package from here, put the tarball somewhere in your home, say `~/projects/`, then extract it.

```
1 cd ~/projects
2 tar xvf ns2-allinone-2.35.tar.gz
3 cd ns2-allinone-2.35
4 ./install
```

The install script will only generate the binaries in current directory, and will NOT actually copy them to anywhere. After the compilation is done, you'll will find the `ns` executable in the `ns-2.35` subdirectory.

Suppose you put your project files (e.g., the TCL file we wrote) in `~/projects/network/ns2`, then it's convenient to have an symbol link to the `ns` binary.

```
1 cd ~/projects/network/ns2
2 ln -svf ~/projects/ns2-allinone-2.35/ns-2.35/ns myns
```

The you'll have an symbol link called `myns`, which points the actual executable. Then you can run your simulation this way.

```
1 myns random_mac.tcl
```

In which the `random_mac.tcl` is the TCL file we wrote in last post.

Add a New Mac Protocol

To add a new MAC protocol, say `RMAC`, we need to do the following. Suppose you're in the `ns2-allinone-2.35/ns-2.35` directory.

- Create `rmac.cc` and `rmac.h` file in the `mac` subdirectory, for now, just leave them empty.
- Edit `Makefile`, find the line contains `mac/smac.o` (around line 249), add one line like this

```
1 .....
2 mac/mac-802_3.o mac/mac-tdma.o mac/smac.o \
3 mac/rmac.o\
4 .....
```

So now, when you do `make` inside the `ns-2.35` directory, our source file `rmac.cc` and `rmac.h` will be compiled. Of course, at this point, there is no content at those two files, which we'll add later.

Adapt the `SimpleMac` Protocol

The NS2 source contains a simple MAC protocol called `SimpleMac`, which is a good start point for us to adapt.

Just copy all the contents in `mac/mac-simple.h` to `mac/rmac.h`, and `mac/mac-simple.cc` to `mac/rmac.cc`. Then change `Mac/Simple` to `Mac/RMAC` line 60 of the `rmac.cc` file. You should be able to compile using the `make` command in `ns-2.35` directory.

If everything is OK, go back to the project directory `~/projects/network/ns2`, change the MAC protocol to `Mac/RMAC` (previously `Mac/802.11`), you should be able to run the simulation using `myns`, which points to the `ns` binary we just compiled.

Change History

- **Sun Dec 15 12:54:11 2013**
  new post

Posted by Jinghao Shi Dec 15th, 2013 In

Related Posts

- OS161 Tool Chain Setup
- Simulate Random MAC Protocol in NS2

View on Github
network
Tagged with: c++, mac, ns2, tcl

Tweet 0 g+1 0

Category: (Part I)

- Simulate Random MAC Protocol in NS2 (Part III)
- Using Cscope INSIDE Vim
- OS161 Swapping

« Simulate Random MAC Protocol in NS2 (Part I) Simulate Random MAC Protocol in NS2 (Part III) »

Comments

0 Comments Pearls in Life Login

Sort by Best Share Favorite

Start the discussion…

ALSO ON PEARLS IN LIFE WHAT'S THIS?

Persist and Synchroize VIM undo History using Dropbox
1 comment • 5 months ago
Avatardilys13 — Hi, I like icecream!!!

Fight Against the 'Address alrady in use' Error
2 comments • 4 months ago
AvatarJinghao Shi — Thanks! Good luck with your other projects!

Quick switch between source and header files in Vim
4 comments • 11 months ago
AvatarJinghao Shi — Thanks for the link. I've updated the blog.

Simulate Random MAC Protocol in NS2 (Part I)
3 comments • 2 months ago
AvatarJinghao Shi — What do you mean? Your project has that requirement?

Did you know...

Simply by pulling on both ends, Chuck Norris can stretch diamonds back into coal.

Blog Stats

136 hits

Popular Posts

- OS161 TLB Miss and Page Fault
- OS161 Coremap
- OS161 fork System Call
- OS161 File System Calls
- OS161 execv System Call

Recent Posts

- Tap Notification to Send Email
- Sum of N Largest Numbers in Google Spreadsheet
- OS161 Tool Chain Setup
- Simulate Random MAC Protocol in NS2 (Part IV)
- Simulate Random MAC Protocol in NS2 (Part III)

Tags

AlertDialog C DownloadManager MLFQ SO_REUSEADDR addrspace backup beamer bind binutils bison bmake c c++ caption cat chrome close compare_xml coremap cron cscope ctags dispatch django dropbox dup2 email exit fd_set fdtable fedora figure function gcc gdb getifaddrs glic google calendar graphicspath graphicx grep heap intent jekyll lfs libpopller linux lock lseek mac migration notification ns2 open ota page fault page table pdflatex pid plugin popular posts prototype python query read readline rsync ruby rwlock sbrk scheduling scp sed select service signapk socket sort spreadsheet ssh stack swap synchronization sys161 syscall table tcl texinfo tlb toolchain undodir vimrc vm waitpid wordpress write yacc yum sty zip

Categories

Android (6)
errors (9)
latex (8)
linux (7)
network (8)
octopress (5)
os161 (25)
tricks (1)
vim (4)

GitHub Repos

- latex.template.slides

  Beamer template

- dotfiles

  Various configuration files

- jhshi.github.com

  Personal Blog

- latex.template.acm-proc

  Latex template for course project proposals, reports.

- latex.template.homework

  Latex template for homework.

@jhshi on GitHub

Stack Overflow

Page Link

Copyright © 2014 - Jinghao Shi - Powered by Octopress