Building ns-2 on Mac M1

Notes from Tom Henderson, Jan 17, 2023

Prerequisites: This guide is based on use of a mac M1 that has XCode Command Line Tools (i.e., not the full XCode) and Homebrew packages to support ns-3.

Caveats: This guide only covers getting ns-2 up and running. nam-1 (the animator) requires X11/XQuartz support and is probably doable but I haven't tried it yet.

Installed software: These are the macOS tool versions that I was using.

```
Xcode Command Line Tools version:
=================================
package-id: com.apple.pkg.CLTools_Executables
version: 14.2.0.0.1.1668646533
volume: /
location: /
install-time: 1672250806
groups: com.apple.FindSystemFiles.pkg-group

Clang version:
==============
Apple clang version 14.0.0 (clang-1400.0.29.202)
Target: arm64-apple-darwin22.2.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin

Operating system version:
=========================
13.1
```

For the record, these are the Homebrew packages that I had previously installed on this machine; however, I'm not sure whether any of them are relevant to building ns-2:

```
% brew leaves
ccache
clang-format
cmake
mercurial
ninja
pkg-config
python@3.10
sphinx-doc
texlive
wget
```

References:

Brian Adamson's guide to building ns-2.35 on an older (Intel) Mac from a few years ago is useful, although some of his hints and fixes are in the ns-2.36 release candidate used herein: https://www.mail-archive.com/ns-users@isi.edu/msg20094.html

Step 1: download latest ns-2.36 release candidate

Download the most recent release candidate for ns-2, release 2.36, which was ns-allinone-2.36.rc2.tar.gz (note: ns-2.36 was never officially released). This file is from November 2014. I have posted it here:

https://www.nsnam.org/release/ns-2/ns-allinone-2.36.rc2.tar.gz

Untar this and cd to the ns-allinon-2.36.rc2 directory

```
% tar xfz ns-allinone-2.36.rc2.tar.gz
% cd ns-allinone-2.36.rc2
```

Step 2: patch tcl8.5.17/generic/tclInt.h

There are four prerequisites that are required to be built before building ns-2: Tcl, Tk, tclcl, and otcl.

There is an `'install` bash script that attempts to automate this. It won't run to completion without patches.

To make it build the four prerequisites: Tcl, Tk, tclcl, otcl, one file must be patched as follows. The file is tcl-8.5.17/generic/tclInt.h (the type of ptrdiff_t must be changed from 'int' to 'long'):

```
--- tcl8.5.17/generic/tclInt.h  2014-10-10 13:12:01
+++ ../../ns-allinone-2.36.rc2/tcl8.5.17/generic/tclInt.h       2023-01-12
07:23:00
@@ -51,7 +51,7 @@
 #ifdef STDC_HEADERS
 #include <stddef.h>
 #else
-typedef int ptrdiff_t;
+typedef long ptrdiff_t;
 #endif
```

Step 3: Run 'install' script until it fails to build ns-2

In the allinone directory, run the './install' script. It first attempts to build some optional packages (xgraph, zlib, GT-ITM, ...). Some will build with warnings and some will fail. They are not needed, though.

It will then build tcl-8.5.17, tk-8.5.17, tclcl-1.20, and otcl-1.14. The script should report that otcl-1.14 and tclcl-1.20 were installed successfully:

```
otcl-1.14 has been installed successfully.
tclcl-1.20 has been installed successfully.
```

However, it will fail to build ns-2 (until it is patched further). The error may show:

```
% ./version:1:1: error: expected unqualified-id
%  2.35
```

Step 4: Rename VERSION files to VERSION.txt

There is an issue with macOS and C++20 version header, due to the case insensitivity of the macOS operating system. So, any file named 'VERSION' will collide with the following include used in the macOS C++ libraries:

```
#include <version>
```

You can read more about it here: https://gitlab.com/qemu-project/qemu/-/issues/129

Until a cleaner solution is found, you should execute the following commands from the allinone directory:

```
% mv nam-1.15/VERSION nam-1.15/VERSION.txt
% mv ns-2.36/diffusion3/VERSION ns-2.36/diffusion3/VERSION.txt
% mv ns-2.36/VERSION ns-2.36/VERSION.txt
% mv otcl-1.14/VERSION otcl-1.14/VERSION.txt
% mv tclcl-1.20/VERSION tclcl-1.20/VERSION.txt
```

Step 5: Patch ns-2.36 to fix compilation errors

Download a small patch named 'ns-2.36.rc2.mac-m1.patch' from the nsnam web site, and place it in the ns-2 directory:

https://www.nsnam.org/release/ns-2/ns-2.36.rc2.mac-m1.patch

Use the Unix patch command to install it:

```
% cd ns-2.36
% patch -p2 -i ns-2.36.rc2.mac-m1.patch
```

You should see:

```
patching file Makefile.in
patching file 'common/tclAppInit.cc'
patching file 'common/tkAppInit.cc'
patching file 'mdart/mdart_adp.cc'
patching file 'mdart/mdart_function.h'
patching file 'nix/nixnode.cc'
```

Step 6: Configure and build ns-2

Within the ns-2 directory, type:

```
% ./configure LIBS="-framework CoreFoundation"
% make
```

If compilation is successful, you should see an executable program called 'ns' left in the ns-2 directory.

Step 7: Run the validation tests

Use the 'validate' script to run ns-2 tests.

```
% ./validate
```

You should see almost all tests pass; the ones that did not pass for me are listed below. Most users should be unconcerned about these specific test failures (unless they are using the PackMime, TMix, or DOCSIS models):

```
validate overall report: some tests failed:
     ./test-all-packmime ./test-all-tmix ./test-all-docsislink
to re-run a specific test, cd tcl/test; ./test-all-TEST-NAME
```

Troubleshooting:

1. Old version of command line tools

If tclcl fails to build, with an error like:

ld: library not found for -lSystem
collect2: error: ld returned 1 exit status

This may be due to outdated command line tools SDK. On my machine, the SDK directory looks like:

```
% pwd
/Library/Developer/CommandLineTools/SDKs
% ls -al
total 0
drwxr-xr-x 7 root wheel  224 Dec 28 10:06 .
drwxr-xr-x 5 root wheel  160 Dec 28 10:06 ..
lrwxr-xr-x 1 root wheel   14 Dec 28 10:06 MacOSX.sdk -> MacOSX13.1.sdk
drwxr-xr-x 7 root wheel  224 Dec 28 10:06 MacOSX12.3.sdk
lrwxr-xr-x 1 root wheel   14 Dec 28 10:05 MacOSX12.sdk -> MacOSX12.3.sdk
drwxr-xr-x 7 root wheel  224 Nov 12 13:18 MacOSX13.1.sdk
lrwxr-xr-x 1 root wheel   14 Dec 28 10:05 MacOSX13.sdk -> MacOSX13.1.sdk
```

If yours is an older version like version 11 or 12, remove and then reinstall Command Line Tools. See these links for more information:

https://mac.install.guide/commandlinetools/6.html

https://mac.install.guide/commandlinetools/7.html

Then try the 'install' script again.

2. Building with gcc/g++

The instructions above are for clang++ compilation. However, if you have installed g++ via Homebrew and the toolchain is preferring it, you may see a build error in ns-2 such as:

```
/opt/homebrew/Cellar/gcc/12.2.0/include/c++/12/bits/stl_iterator.h:1748:5:
note:   template argument deduction/substitution failed:
```

```
/opt/homebrew/Cellar/gcc/12.2.0/include/c++/12/bits/stl_algobase.h:259:15:
note:   'const TracedInt' is not derived from 'const
std::move_iterator<_IteratorL>'
  259 |       if (__a < __b)
      |           ~~~~^~~~~
make: *** [tcp/tcp-fack.o] Error 1
```

If so, try these commands to force usage of clang++:

```
make clean
CXX=clang++ ./configure LIBS="-framework CoreFoundation"
CXX=clang++ make
```