WOSS Custom Integration Framework with Acoustic Toolbox with NS3-dev (with Database, NETCDF4 and HDF5 support)

===

This repository aims to introduce custom `WOSS` support for `ns3 users`.

`WOSS` is a multi-threaded C++ framework that permits the integration of any existing underwater channel simulator that expects environmental data as input and provides as output a channel realization.

Currently, `WOSS` integrates the `Bellhop ray-tracing` program.

Thanks to its automation the user only has to specify the location in the world and the time where the simulation should take place. This is done by setting the simulated date and the wanted latitude and longitude of every node involved. The simulator automatically handles the rest (see technical description).

`WOSS` can be integrated in any network simulator or simulation tool that supports `C++`.

'woss-ns3' relies on the following libraries:

```
- WOSS
- NetCDF
- Acoustic Toolbox
```

First make sure you have `gfortran`, `gcc` and `gcxx compiler`.

Please checked if the GNU Fortran compiler was in my system by typing `gfortran --version`:

```
GNU Fortran (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

If you go for the GNU compiler, type:

```
export FC=gfortran
```

Install latest `Acoustic Toolbox` (March 2019)

```
cd ${HOME}/Documents
wget http://telecom.dei.unipd.it/ns/woss/files/at.zip
tar -xzf at.zip
cd at/at
make
sudo make install
```

Once installed, let's tell the system where to find our new libraries:

```
export PATH = /home/ns/Documents/at/at/Bellhop:/home/ns/Documents/at/at/:$PATH
```

You have to build support for `NETCDF4` and `HDF5` for `NS3`, `WOSS` and `Acoustic Toolbox`, Please make sure you follow this steps for installation :

export environment

```
export F77=gfortran
export FC=gfortran
export CC=gcc
export CXX=g++
export CFLAGS=-fPIC
```

Step 1. Install compilers and building tools

First let's check which Linux are you running with the command:

```
lsb_release -ds
```

Will return something like:

```
Debian GNU/Linux 9.8 (stretch)
```

- For *Debian/Ubuntu/Linux Mint*:

```
sudo apt-get update
sudo apt-get install wget nano gfortran m4 build-essential
```

Step 2. Install `NETCDF`

Until version 4.1.3, `NETCDF` was bundled in a single package. Since then, has been split off into independent distributions (`netCDF-C`, `netCDF-Fortran`, `netCDF-Java`, `netCDF-Python`, `netCDF-C++` and so on).

Let's start with downloading and installing `netCDF-C` in a new folder called `netcdf` in user home directory (e.g. `/home/ns/`):

```
wget http://telecom.dei.unipd.it/ns/woss/files/netcdf-c-4.7.3.tar.gz
tar -xzf netcdf-c-4.7.3.tar.gz
cd netcdf-c-4.7.3
./configure --prefix=${HOME}/netcdf --disable-dap --disable-netcdf-4 --enable-shared
make
make check
sudo make install
cd ${HOME}
```

Step 3. Now please install `HDF5` support for `NETCDF`

```
wget http://telecom.dei.unipd.it/ns/woss/files/hdf5-1.8.13.tar.gz
tar -xzf hdf5-1.8.13.tar.gz
cd hdf5-1.8.13
./configure --enable-shared --prefix=${HOME}/netcdf --disable-dap --enable-fortran #If above command didn't work then please
try ./configure --enable-shared --prefix=${HOME}/netcdf
make
make check
sudo make install
cd ${HOME}
```

Step 4. Again configure `NETCDF` with `HDF5` support

```
cd netcdf-c-4.6.3
./configure --prefix=${HOME}/netcdf --disable-dap --enable-netcdf-4 --enable-shared CPPFLAGS="$CPPFLAGS -I${HOME}/netcdf/inc
lude" LDFLAGS="$LDFLAGS -L${HOME}/netcdf/lib" --enable-fortran --enable-cxx
make
make check
sudo make install
cd ${HOME}
```

Step 5. Now install `NETCDF4 C++` support

```
wget http://telecom.dei.unipd.it/ns/woss/files/netcdf-cxx4-4.3.1.tar.gz
tar -xzf netcdf-cxx4-4.3.1.tar.gz
cd netcdf-cxx4-4.3.1
./configure --prefix=${HOME}/netcdf --enable-shared CPPFLAGS="$CPPFLAGS -I${HOME}/netcdf/include" LDFLAGS="$LDFLAGS -L${HOM
E}/netcdf/lib"
make
make check
sudo make install
cd ${HOME}
```

Once installed, let’s tell the system where to find our new libraries and export variables to .bashrc:

```
export NETCDF=${HOME}/netcdf
export PATH=$NETCDF/bin:$PATH
export NETCDF_INCDIR=$NETCDF/include
export NETCDF_LIBDIR=$NETCDF/lib
export LD_LIBRARY_PATH=$NETCDF/lib:$LD_LIBRARY_PATH
export PATH NETCDF
```

Step 6. Install `WOSS` library support

```
wget http://telecom.dei.unipd.it/ns/woss/files/WOSS-v1.9.0.tar.gz
tar -xzf woss-1.9.0.tar.gz
cd woss-v1.9.0
./autogen.sh
./configure --with-netcdf4=${HOME}/netcdf --with-pthread
make
make check
sudo make install
wget http://telecom.dei.unipd.it/ns/woss/files/WOSS-dbs-v1.3.0.tar.gz
tar -xzf WOSS-dbs-v1.3.0.tar.gz
cd dbs/bathymetry
wget https://www.bodc.ac.uk/data/open_download/gebco/GEBCO_15SEC/zip/  #this is almost ~ 12 GB database.
tar -xzf GEBCO_2019.zip #This is database used by WOSS, more details are available on http://telecom.dei.unipd.it/ns/woss/do
xygen/database.html
```

`NS3-dev` Installation

```
cd Documents/
mkdir workspace
cd workspace
wget https://www.nsnam.org/release/ns-allinone-3.30.tar.bz2
tar xjf ns-allinone-3.30.tar.bz2
cd ns-allinone-3.30/
./build.py --enable-examples --enable-tests
cd ns-3.30/
./waf --run scratch/scratch-simulator #To check everything working fine
```

Please don't use sudo to build ns3-dev as it makes conficts when woss try to call bellhop.exe in runtime, to chech you have correctly set your PATH for acoustic toolbox,

```
which bellhop.exe
```

It will show you path of your bellhop.exe, same like this,

```
/home/ns/Documents/at/bin/bellhop.exe
```

configure `WOSS` libraries for `ns3-dev`

```
cd ${HOME}/Documents/workspace/ns-allinone-3.30/ns-3.30/src
wget https://github.com/MetalKnight/woss-ns3.git
tar xjf woss-ns3-master.zip #Remove zip file from src folder as this will conflict while building NS-3
mv woss-ns3-master woss-ns3 #Rename folder as woss-ns3, make sure folder name is correct to avoid config errors
cd ..
./waf -d debug --enable-tests --enable-examples --enable-sudo --with-woss-source=/home/medit/woss-1.9.0/ --with-woss-library
=/home/medit/woss-1.9.0/woss --with-netcdf4-install=${HOME}/netcdf configure CXXFLAGS="-Wall -Werror -Wno-unused-variable"
#this configure your ns3, Please make sure woss source(in my case i.e /home/medit/woss-1.9.0/) is correct and installed woss
library (in my case i.e /home/medit/woss-1.9.0/woss) is provided with correct path, also make sure you put the correct path
 for netCDF(in my case i.e${HOME}/netcdf)
```

You can see, `./waf` will configure and please check if you got message in second last command line saying

```
# WOSS Integration Framework    : enabled
```

In my case, It looks like,

```
medit@medit-Vostro-230:~/Documents/workspace/ns-allinone-3.30/ns-3.30$ ./waf -d debug --enable-tests --enable-examples --ena
ble-sudo --with-woss-source=/home/medit/woss-1.9.0/ --with-woss-library=/home/medit/woss-1.9.0/woss --with-netcdf4-install=
${HOME}/netcdf configure
Setting top to                           : /home/ns/Documents/workspace/ns-allinone-3.30/ns-3.30
Setting out to                           : /home/ns/Documents/workspace/ns-allinone-3.30/ns-3.30/build
Checking for 'gcc' (C compiler)          : /usr/bin/gcc
Checking for cc version                  : 7.4.0
Checking for 'g++' (C++ compiler)        : /usr/bin/g++
Checking for compilation flag -Wl,--soname=foo support : ok
Checking for compilation flag -std=c++11 support       : ok
Checking boost includes                                : headers not found, please provide a --boost-includes argument (see
help)
```

```
Checking boost includes                                   : headers not found, please provide a --boost-includes argument (see
help)
Checking for program 'python'                             : /usr/bin/python3
Checking for python version >= 2.3                        : 3.6.9
python-config                                             : not found
Checking for library python3.6m in LIBDIR                 : not found
Checking for library python3.6m in python_LIBPL           : yes
Checking for header Python.h                              : Distutils not installed? Broken python installation? Get python-con
fig now!
Checking for click location                               : not found
Checking for program 'pkg-config'                         : /usr/bin/pkg-config
Checking for 'gtk+-3.0'                                   : not found
Checking for 'libxml-2.0'                                 : not found
checking for uint128_t                                    : not found
checking for __uint128_t                                  : yes
Checking high precision implementation                    : 128-bit integer (default)
Checking for header stdint.h                              : yes
Checking for header inttypes.h                            : yes
Checking for header sys/inttypes.h                        : not found
Checking for header sys/types.h                           : yes
Checking for header sys/stat.h                            : yes
Checking for header dirent.h                              : yes
Checking for header stdlib.h                              : yes
Checking for header signal.h                              : yes
Checking for header pthread.h                             : yes
Checking for header stdint.h                              : yes
Checking for header inttypes.h                            : yes
Checking for header sys/inttypes.h                        : not found
Checking for library rt                                   : yes
Checking for header sys/ioctl.h                           : yes
Checking for header net/if.h                              : yes
Checking for header net/ethernet.h                        : yes
Checking for header linux/if_tun.h                        : yes
Checking for header netpacket/packet.h                    : yes
Checking for NSC location                                 : not found
Checking for 'sqlite3'                                    : not found
Checking for header linux/if_tun.h                        : yes
Checking the given WOSS source code path                  : /home/ns/woss-1.9.0/ (given)
Checking the given WOSS library path                      : /home/ns/woss-1.9.0/woss (given)
Checking the given NetCDF4 and HDF5 install path          : /home/ns/netcdf (given)
WOSS source code path is valid                            : /home/ns/woss-1.9.0/woss
WOSS source code path is valid                            : /home/ns/woss-1.9.0/woss/woss_def
WOSS source code path is valid                            : /home/ns/woss-1.9.0/woss/woss_db
Checking the given WOSS library                           : yes
NetCDF4 and HDF5 source code path                         : /home/ns/netcdf/include
Checking the given NETCDF4 and HDF5 libraries             : yes
Checking for program 'sudo'                               : /usr/bin/sudo
Checking for program 'valgrind'                           : not found
Checking for 'gsl'                                        : not found
libgcrypt-config                                          : not found
Checking for compilation flag -fstrict-aliasing support : ok
Checking for compilation flag -fstrict-aliasing support : ok
Checking for compilation flag -Wstrict-aliasing support : ok
Checking for compilation flag -Wstrict-aliasing support : ok
Checking for program 'doxygen'                          : not found
---- Summary of optional NS-3 features:
Build profile                 : debug
Build directory               :
BRITE Integration             : not enabled (BRITE not enabled (see option --with-brite))
DES Metrics event collection  : not enabled (defaults to disabled)
Emulation FdNetDevice         : enabled
Examples                      : enabled
File descriptor NetDevice     : enabled
GNU Scientific Library (GSL)  : not enabled (GSL not found)
Gcrypt library                : not enabled (libgcrypt not found: you can use libgcrypt-config to find its location.)
GtkConfigStore                : not enabled (library 'gtk+-3.0 >= 3.0' not found)
MPI Support                   : not enabled (option --enable-mpi not selected)
NS-3 Click Integration        : not enabled (nsclick not enabled (see option --with-nsclick))
NS-3 OpenFlow Integration     : not enabled (Required boost libraries not found)
Network Simulation Cradle     : not enabled (NSC not found (see option --with-nsc))
PlanetLab FdNetDevice         : not enabled (PlanetLab operating system not detected (see option --force-planetlab))
PyViz visualizer              : not enabled (Python Bindings are needed but not enabled)
Python Bindings               : not enabled (Python library or headers missing)
Real Time Simulator           : enabled
SQlite stats data output      : not enabled (library 'sqlite3' not found)
Tap Bridge                    : enabled
Tap FdNetDevice               : enabled
Tests                         : enabled
Threading Primitives          : enabled
Use sudo to set suid bit      : enabled
WOSS Integration Framework    : enabled
XmlIo                         : not enabled (library 'libxml-2.0 >= 2.7' not found)
'configure' finished successfully (2.808s)
```

Check out second last line to make sure everything is perfectly configure.

Then build your `ns3` again,

```
./waf build CXXFLAGS="-Wall -Werror -Wno-unused-variable"
```

you'll see the following,

```
Waf: Leaving directory `/home/ns/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (11m35.055s)

Modules built:
antenna                   aodv                      applications
bridge                    buildings                 config-store
core                      csma                      csma-layout
dsdv                      dsr                       energy
fd-net-device             flow-monitor              internet
internet-apps             lr-wpan                   lte
mesh                      mobility                  mpi
netanim (no Python)       network                   nix-vector-routing
```

```
olsr                      point-to-point            point-to-point-layout
propagation               sixlowpan                 spectrum
stats                     tap-bridge                test (no Python)
topology-read             traffic-control           uan
virtual-net-device        wave                      wifi
wimax                     woss-ns3 (no Python)

Modules not built (see ns-3 tutorial for explanation):
brite                     click                     openflow
visualizer

```

Please run one of the example from scratch,

```
./waf --run scratch/scratch-simulator
```

If you see the below output, Hola ! Its working now.

```
Waf: Entering directory `/home/ns/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory `/home/ns/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.749s)
Scratch Simulator
```

Then Run WOSS Aloha Example with following example (please note that scratch folder is not yet added to module so you have to manually run your examples from the woss-ns3/examples folder itself.)

```
medit@medit-Vostro-230:~/Documents/workspace/ns-allinone-3.30/ns-3.30$ ./waf --run src/woss-ns3/examples/woss-aloha-example
Waf: Entering directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.731s)
WossManagerResDbMT::checkConcurrentThreads() 6
WossManagerResDbMT::checkConcurrentThreads() 4
Received a packet of size 1000 bytes
Received a total of 1000 bytes at sink
```

Troubleshoot

1. If you're running example and run into problem with tap-creator like following then you have to manually change few settings.

```
./waf --run src/woss-ns3/examples/woss-aloha-example --command-template="gdb --args %s <args>"
```

```
Waf: Entering directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
* Several tasks use the same identifier. Please check the information on
   https://waf.io/apidocs/Task.html?highlight=uid#waflib.Task.Task.uid
  - object 'SuidBuild_task' (
    {task 140252058325512: SuidBuild_task  -> }) defined in 'tap-creator'
  - object 'SuidBuild_task' (
    {task 140252058325624: SuidBuild_task  -> }) defined in 'tap-creator'
  - object 'SuidBuild_task' (
    {task 140252058325736: SuidBuild_task  -> }) defined in 'tap-creator'
Waf: Leaving directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.892s)
```

first figure out the problem with gdb command by running following command,

```
./waf --run src/woss-ns3/examples/woss-aloha-example --command-template="gdb --args %s <args
```

Then maybe the easier one for the time being, open the /home/usr/netcdf/include/ncGroup.h
and remove the line 18 from that file, which is not used by the library anyway.

2. If you're not able to find or use the common shared files/library,

```
medit@medit-Vostro-230:~/Documents/workspace/ns-allinone-3.30/ns-3.30$ ./waf --run src/woss-ns3/examples/woss-aloha-example
Waf: Entering directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.713s)
/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build/src/woss-ns3/examples/ns3.30-woss-aloha-example-debug: error
while loading shared libraries: libWOSS.so.0: cannot open shared object file: No such file or directory
Command ['/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build/src/woss-ns3/examples/ns3.30-woss-aloha-example-deb
ug'] exited with code 127
```

please do following to solve the error,

Edit /etc/ld.so.conf or create something in /etc/ld.so.conf.d/ to add /usr/local/lib and /usr/local/lib64. Then run ldconfig.

In my case,

```
sudo nano /etc/ld.so.conf
```

Please make sure you have the correct lib in the same file, in my case it was,

```
include /etc/ld.so.conf.d/*.conf
add /usr/local/lib:/usr/local/lib64  //added by Jay to access common or shared libraries
```

save the same file and configure again by

```
sudo ldconfig
```

It should work now. In my case, it worked

```
medit@medit-Vostro-230:~/Documents/workspace/ns-allinone-3.30/ns-3.30$ ./waf --run src/woss-ns3/examples/woss-aloha-example
Waf: Entering directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory `/home/medit/Documents/workspace/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.731s)
WossManagerResDbMT::checkConcurrentThreads() 6
WossManagerResDbMT::checkConcurrentThreads() 4
```

```
Received a packet of size 1000 bytes
Received a total of 1000 bytes at sink
```

Author

- ***Jay Patel*** patel.jay@dal.ca

References

1. http://telecom.dei.unipd.it/ns/woss/, Special Thanks to Federico Guerra
2. https://www.nsnam.org/
3. https://github.com/javirg/SWAN-Support/blob/master/recipes/build_linux_netcdf.md