I-Installation NS2

1-Première étape, installez le compilateur gcc et créez

$]sudo apt-get update

$]sudo apt-get install gcc-4.8 g++-4.8

$]sudo apt-get install build-essential

$]sudo apt install perl libxt-dev libx11-dev libxmu-dev

$]wget http://archive.ubuntu.com/ubuntu/pool/universe/x/xgraph/xgraph\_12.1-17\_amd64.deb

$]wget http://archive.ubuntu.com/ubuntu/pool/main/libx/libx11/libx11-6\_1.6.4-3ubuntu0.2\_amd64.deb

$]wget http://archive.ubuntu.com/ubuntu/pool/main/g/glibc/libc6\_2.27-3ubuntu1\_amd64.deb

$]sudo dpkg -i \*.deb

2-Deuxième étape, installez les dépendances

$]sudo apt-get install tcl8.5 tcl8.5-dev tk8.5 tk8.5-dev #for tcl and tk

$]sudo apt-get install libxmu-dev libxmu-headers

3-Décompressez et compilez le fichier

$]tar xvf ns-allinone-2.35.tar.gz

$]cd ns-allinone-2.35/ns-2.35

et ouvre le fichier Make file.in et on s'assure que CC = gcc-4.8 et CPP = g++-4.8

$] gedit linkstate/ls.h et line number 137, change erase to this->erase

$]cd ns-allinone-2.35/

$]export CC=gcc-4.8 CXX=g++-4.8 && ./install

4-écrivez les paramètres variables dans l'environnement du terminal

$]sudo gedit ~/.bashrc

#LD\_LIBRARY\_PATH

OTCL\_LIB=/home/pradeepkumar/ns-allinone-2.35/otcl-1.14/

NS2\_LIB=/home/pradeepkumar/ns-allinone-2.35/lib/

USR\_Local\_LIB=/usr/local/lib/

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$OTCL\_LIB:$NS2\_LIB:$USR\_Local\_LIB

#TCL\_LIBRARY

TCL\_LIB=/home/pradeepkumar/ns-allinone-2.35/tcl8.5.10/library/

USR\_LIB=/usr/lib/

export TCL\_LIBRARY=$TCL\_LIBRARY:$TCL\_LIB:$USR\_LIB

#PATH

XGRAPH=/home/pradeepkumar/ns-allinone-2.35/xgraph-12.2/:/home/pradeepkumar/ns-allinone-2.35/bin:/home/pradeepkumar/ns-allinone-2.35/tcl8.5.10/unix:/home/pradeepkumar/ns-allinone-2.35/tk8.5.10/unix

NS=/home/pradeepkumar/ns-allinone-2.35/ns-2.35/

NAM=/home/pradeepkumar/ns-allinone-2.35/nam-1.15/

export PATH=$PATH:$NS:$NAM:$XGRAPH

$]source ~/.bashrc

$]cd ns-allinone-2.35/ns-2.35

$]./validate

5-Exécutez l'exemple

$]ns /home/pradeepkumar/ns-allinone-2.35/ns-2.35/tcl/ex/simple.tcl

II Clonage du protocole AODV

**4.2.1 Clonage du protocole aodv en taodv**

Pour le clonage et les modifications du protocole du protocole, nous utiliserons l’IDE Eclipse, qui est un logiciel un logiciel open-source contenant un environnement de développement et un éditeur de texte pour les langages C et C++.

Les différentes étapes du clonage et modifications sont les suivantes :

**Etape1** : Aller dans le #ns-allinone-2.35/ns-2.35, et y créer le répertoire TAODV.

**Etape 2** : Aller dans #ns-allinone-2.35\ns-2.35\common\packet.h

Au-dessus de la ligne static *packet\_t PT\_NTYPE = 74;*

Insérer la ligne ***static const packet\_t PT\_TAODV = 73;***

Au-dessous de la ligne de *type == PT\_AODV*||

Insérer la ligne ***type == PT\_TAODV ||***

Au-dessous de la ligne *name\_[PT\_DCCP\_RESET] ="DCCP\_Reset";*

Insérer la ligne ***name\_[PT\_TAODV]= "TAODV";***

**Etape 3** : Aller dans #ns-allinone-2.35\ns-2.35\trace\cmu-trace.h

Au-dessous de la ligne *void format\_aodv(Packet \*p, int offset);*

Insérer la ligne ***void format\_taodv(Packet \*p, int offset);***

**Etape 4** : Aller dans *#ns-allinone-2.35\ns-2.35\trace\cmu-trace.cc*

Ajouter la ligne **#include <taodv/taodv\_packet.h>**

De la ligne 862 à 965 et 1528 à1530 remplacer : tous les *aodv* par ***taodv*** et *AODV* par ***TAODV***

**Etape 5** : Aller dans #ns-allinone-2.35\ns-2.35\tcl\lib\ns-packet.tcl

Au-dessous de la ligne *AODV* ***#*** *routing protocol for ad-hoc networks*

Insérer la ligne ***TAODV # routing protocol for ad-hoc networks***

**Etape 6** : Aller dans #ns-allinone-2.35\ns-2.35\tcl\lib\ns-lib.tcl

Au-dessous de AODV *{*

*set ragent [$self create-aodv-agent $node]*

*}*

Insérer ***TAODV***

***{***

***set ragent [$self create-taodv-agent $node]***

***}***

Ajouter

***Simulator instproc create-taodv-agent { node } {***

***# Create TAODV routing agent***

***set ragent [new Agent/TAODV [$node node-addr]]***

***$self at 0.0 "$ragent start" ;# start BEACON/HELLO messages***

***$node set ragent\_ $ragent***

***return $ragent***

***}***

**Etape 7 :** #ns-allinone-2.35\ns-2.35\queue\priqueue.cc

Au-dessous de la ligne *case PT\_AODV:*

Inserer la ligne ***case PT\_TAODV:***

**Etape 8:** #ns-allinone-2.35\ns-2.35\Makefile

*OBJ\_CC = \*

......

......

*aodv/aodv\_logs.o aodv/aodv.o \*

*aodv/aodv\_rtable.o aodv/aodv\_rqueue.o \*

**taodv/taodv\_logs.o taodv/taodv.o \**

**taodv/taodv\_rtable.o taodv/taodv\_rqueue.o \**

......

......

*$(OBJ\_STL)*

**Etape 9:** #ns-allinone-2.35\ns-2.35\tcl\lib\ns-agent.tcl

Au-dessous de la *Agent/AODV set dport\_ 0*

Inserer ***Agent/TAODV instproc init args {***

***$self next $args***

***}***

***Agent/TAODV set sport\_ 0***

***Agent/TAODV set dport\_ 0***

**Etape 10:** #ns-allinone-2.35\ns-2.35\tcl\lib\ns-mobilenode.tcl

......

......

*# Special processing for AODV*

*set aodvonly [string first "AODV" [$agent info class]]*

*if {$aodvonly != -1 } {*

*$agent if-queue [$self set ifq\_(0)] ;# ifq between LL*

*and MAC*

*}*

*#*Nouvelles lignes

***# Special processing for TAODV***

***set taodvonly [string first "TAODV" [$agent info class]]***

***if {$taodvonly != -1 } {***

***$agent if-queue [$self set ifq\_(0)] ;# ifq between LL and MAC***

***}***

**Etape 11 :** #ns-allinone-2.35\ns-2.35\routing\rtable.h

Au-dessous de laligne *friend class AODV;* et au-dessus de la ligne *friend class rt\_entry;*

Insérer laligne ***friend class TAODV;***

}

class rt\_entry {

friend class rttable;

*friend class AODV;*

***friend class TAODV; //nouvelle ligne***

***friend class TAODVLocalRepairTimer; //ligne modifiée***

.....

.....

.....

**Etape 12:** #ns-allinone-2.35\ns-2.35\wpan\p802\_15\_4nam.cc

Au-dessous de la ligne *(strcmp(packet\_info.name(PT\_AODV),name) == 0)?PT\_AODV:*

Insérer la ligne ***(strcmp(packet\_info.name(PT\_TAODV),name) == 0)?PT\_TAODV:***

**Etape 13 :** #ns-allinone-2.35\dei80211mr-1.1.4\src\InitTCL.cc

Au-dessous des lignes :

*PacketHeaderManager set tab\_(PacketHeader/SR) 1\n\*

*PacketHeaderManager set tab\_(PacketHeader/AODV) 1\n\*

Insérer la ligne ***PacketHeaderManager set tab\_(PacketHeader/TAODV) 1\n\***

**Etape 14** : Copier tous les fichiers du répertoire AODV vers TAODV en les renommant en tous ceux qui ont pour nom aodv en taodv et AODV en TAODV.

**Etape 15 :** Renommer toutes les classes ***Timer*** dans taodv.h et taodv.cc

*Exemple dans taodv.h*

*class* ***TAODVBroadcastTimer*** *: public Handler {*

*public:*

***TAODVBroadcastTimer****(TAODV\* a) : agent(a) {}*

*void handle(Event\*);*

*private:*

*TAODV \*agent;*

*Event intr;*

*};*

**Etape 16 :** Recompilation du code source:

*Etape 1* : Recompiler ``packet.cc`` comme ``packet.h'' est modifié.

*Etape 2* : ./configure (si cette commande échoue aller à l’étape 17)

*Etape 3*: make clean

*Etape 4*: make

*Etape 5*: make install

**Etape 17** : Si ./configure échoue alors exécuter ./install

$cd ns-allinone-2.35

$./install

$cd ns-allinone-2.35/ns-2.35

$sudo make install

III- Activation du mode promiscuité

1) We need to modify in total 3 files, so it's good to take a backup of it.  
    Files are:

* + ns-allinone-2.34/ns-2.34/aodv/aodv.cc
  + ns-allinone-2.34/ns-2.34/aodv/aodv.h
  + ns-allinone-2.34/ns-2.34/tcl/lib/ns-mobilenode.tcl

2) Open the file ns-allinone-2.34/ns-2.34/aodv/aodv.h in your favorite editor  
    and make the changes as shown in blue color.

**#include <mac.h>**

  class AODV: **public Tap,** public Agent {

  public:

**void tap(const Packet \*p);**

  ......

  protected:

**Mac \*mac\_;**

  ......

 }

3) Open the file ns-allinone-2.34/ns-2.34/aodv/aodv.cc and make the changes as

    shown in blue color.

int  
AODV::command(int argc, const char\* const \* argv) {

......

else if(argc == 3) {

......

**else if (strcmp(argv[1], "install-tap") == 0) {  
mac\_ = (Mac\*)TclObject::lookup(argv[2]);  
if (mac\_ == 0) return TCL\_ERROR;  
mac\_->installTap(this);  
return TCL\_OK;  
}**

}  
return Agent::command(argc, argv);

}

**void  
AODV::tap(const Packet \*p) {**

**// put your code here**

**}**

4) Open the file ns-allinone-2.34/ns-2.34/tcl/lib/ns-mobilenode.tcl and make the

    changes as shown in blue color.

Node/MobileNode instproc add-target { agent port } {

$self instvar dmux\_ imep\_ toraDebug\_ **mac\_**

......

# Special processing for AODV  
set aodvonly [string first "AODV" [$agent info class]]  
if {$aodvonly != -1 } {  
$agent if-queue [$self set ifq\_(0)] ; # ifq between LL and MAC  
  
**$agent install-tap $mac\_(0)**

**......**  
  
}