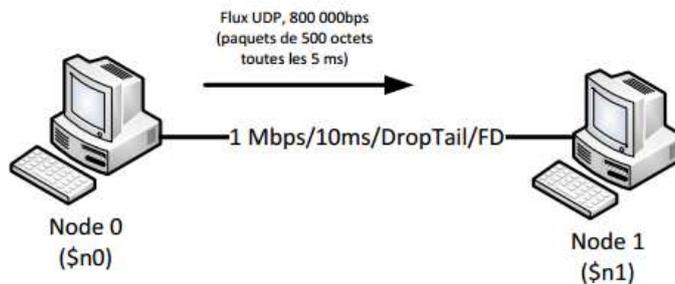


## - TP6 - : Introduction à NS2

### Scénario 1 (réseau filaire)



### Description :

- Vous allez créer deux nœuds NS-2 (node), reliés par un lien full duplex (duplex-link) supportant un débit de 1Mbps, un temps d'accès au medium de 10 ms, et qui utilise l'algorithme de file d'attente DropTail pour la gestion des congestions.
- Vous allez ensuite créer un agent UDP (Agent/UDP) attaché au nœud n0 (attach-agent) qui permet à n0 de transmettre des paquets UDP sur le réseau.
- Puis, vous allez créer un envoi de paquets à débit constant (Constant Bit Rate –CBRApplcation/Traffic/CBR) attaché à l'agent UDP défini précédemment (attach-agent).
- Vous allez après créer un agent vide, destiné à recevoir des paquets UDP ou TCP sans les traiter (Agent/Null). On l'attache au nœud n1 (attach-agent) puis on connecte l'agent UDP et l'agent vide (connect).
- Enfin, on indique que les traces de simulation doivent être sauvegardées, qu'on souhaite lancer NAM à la fin de la simulation, que celle-ci va durer 5 secondes et que le CBR ne sera émis qu'à partir de la 0.5ème seconde de simulation, et jusqu'à la 4.5ème seconde.
- On lance ensuite la simulation et la visualisation du résultat à l'aide de la commande :  
**\$ ns RTIC\_01.tcl**

Le script de description est le suivant :

```

# Création d'un simulateur
set ns [new Simulator]
# Création du fichier de trace utilisé par le visualisateur
set nf [open out.nam w]
# Indique à NS de logger ses traces dans le fichier $nf (out.nam)
$ns namtrace-all $nf
# Création du fichier de trace et indication à ns de l'utiliser
set f0 [open trace.tr w]
$ns trace-all $f0
# Lorsque la simulation sera terminée, cette procédure sera appelée
# pour lancer automatiquement le visualisateur (NAM)
proc finish { } {
# Force l'écriture dans le fichier des infos de trace
global ns nf f0
$ns flush-trace
close $nf
close $f0
# Lance l'outil de visualisation nam
exec nam out.nam &
# Quitte le script TCL
exit 0}
# création de deux noeuds
set n0 [$ns node]
set n1 [$ns node]
# Création d'une liaison de communication full duplex entre les noeuds n0 & n1
# Fonctionne à 1Mbps, 10ms de délai, et utilise l'algorithme de file DropTail
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
# création d'un agent UDP implanté dans n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
# Création d'un traffic CBR pour le noeud 0 générateur de paquets à vitesse constante
# Paquets de 500 octets (4000 bits), générés toutes les 5 ms.
# ---> Ceci représente un trafic de 800 000 bps (inférieur à la capacité du lien)
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
# Ce traffic est attaché à l'agent UDP udp0
$cbr0 attach-agent $udp0
# Création d'un agent vide, destiné à recevoir les paquets dans le noeud n1
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
# Le trafic issu de l'agent udp0 est envoyé vers null0
$ns connect $udp0 $null0
# Début de l'envoi du CBR à 0.5s après le début de la simulation
$ns at 0.5 "$cbr0 start"
# Fin de l'envoi du CBR à 4.5s après la fin de la simulation
$ns at 4.5 "$cbr0 stop"
# La simulation s'arrête après 5 secondes, et appelle la procédure
# TCL nommée "finish" définie précédemment
$ns at 5.0 "finish"
# Démarrage du moteur de simulation
$ns run

```

Après exécution, voici un extrait de ce que nous pouvons visualiser dans le fichier de trace :

```

r 1.064 0 2 cbr 500 ----- 1 0.0 3.0 110 220
r 1.064 1 2 cbr 500 ----- 2 1.0 3.0 110 221
+ 1.065 0 2 cbr 500 ----- 1 0.0 3.0 113 226
- 1.065 0 2 cbr 500 ----- 1 0.0 3.0 113 226
+ 1.065 1 2 cbr 500 ----- 2 1.0 3.0 113 227
- 1.065 1 2 cbr 500 ----- 2 1.0 3.0 113 227
r 1.069 0 2 cbr 500 ----- 1 0.0 3.0 111 222
r 1.069 1 2 cbr 500 ----- 2 1.0 3.0 111 223

```

Voici un descriptif de chaque élément du fichier de trace :

opération	temps	Source	dest	Type pck	taille	drapeaux	Flot id	@src	@dest	n°pck	Id pck
+	1.065	1	2	cbr	500	----	2	1.0	3.0	113	227

Interprétation du fichier de trace

- 1ere colonne :**  
+ mise en file d'attente  
- sortie de la file d'attente  
d suppression de la file d'attente  
r réception au noeud  
l perte(suite a une erreur binaire)

## Travail demandé :

Rédiger un rapport sur :

- 1- Ecrire un script **awk** qui permet de générer un fichier **analyse1\_lan.dat** qui contient deux colonnes (Temps, Id\_packet)
- 2- Tracer la courbe du fichier **analyse1\_lan.dat**
- 3- Ecrire un script **awk** qui permet de générer un fichier **analyse2\_lan.dat** qui contient deux colonnes (Temps\_moy, Nb\_packet) où  
Temps\_moy : est l'intervalle entier du temps d'exécution 1.2....5  
Nb\_packet : est le nombre de paquets reçu (r) dans chaque seconde
- 4- Tracer la courbe du fichier **analyse2\_lan.dat**

## Scénario 2 (Réseaux ad hoc)

Dans la première partie, on définit un certain nombre de variables globales qui sont utilisées dans tout le programme. Les nœuds par exemple ont les mêmes propriétés, notamment au niveau de la couche physique où on retrouve les points suivants :

### **# La capacité du médium est de 2Mbit/s**

```
Phy/WirelessPhy set bandwidth_ 2Mb
```

### **# La puissance de transmission est de 0.28183815 dB**

```
Phy/WirelessPhy set Pt_ 0.28183815
```

### **#Configuration de l'environnement**

```
set val(chan)      Channel/WirelessChannel ; # channel type
set val(prop)      Propagation/TwoRayGround ; # radio-propagation model
set val(netif)     Phy/WirelessPhy ; # network interface type
set val(mac)       Mac/802_11 ; # MAC type
set val(ifq)       Queue/DropTail/PriQueue ; # interface queue type
set val(ll)        LL ; # link layer type
set val(ant)       Antenna/OmniAntenna ; # antenna model
set val(ifqlen)    50 ; # max packet in ifq
set val(nn)        4 ; # number of mobilenodes
set val(rp)        AODV ; # routing protocol
set val(x)         500 ; # X dimension of topography
set val(y)         500 ; # Y dimension of topography
```

### **#Creation d'un simulateur ainsi que les fichiers traces**

```
set ns [new Simulator]
set tracefd [open simple.tr w] ;# toutes les traces de la simulation
set namtrace [open simwrls.nam w] ;# trace de simulation pour NAM
```

### **#Procédure de fin d'exécution**

```
proc finish {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $namtrace
    close $tracefd
    exec nam simwrls.nam &
    exit 0
}
```

```
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

### **#Creation de topographie de simulation**

```
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
```

### **#Configuration des noeuds**

```
$ns node-config -adhocRouting $val(rp) \
  -llType $val(ll) \
  -macType $val(mac) \
  -ifqType $val(ifq) \
  -ifqLen $val(ifqlen) \
  -antType $val(ant) \
  -propType $val(prop) \
  -phyType $val(netif) \
  -channelType $val(chan) \
  -topoInstance $topo \
  -agentTrace ON \
  -routerTrace ON \
  -macTrace OFF \
  -movementTrace ON
```

### **#Creation des noeuds**

```
for {set i 0} {$i < $val(nn)} {incr i} {
  set node_($i) [$ns node]
  $ns initial_node_pos $node_($i) 30; }
```

### **#Position initial des noeuds**

```
$node_(0) set X_ 50.0
$node_(0) set Y_ 50.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 200.0
$node_(1) set Y_ 350.0
$node_(1) set Z_ 0.0
$node_(2) set X_ 300.0
$node_(2) set Y_ 100.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 350.0
$node_(3) set Y_ 200.0
$node_(3) set Z_ 0.0
```

### **#Génération de la mobilité dans les noeuds**

```
$ns at 1.0 "$node_(0) setdest 300.0 250.0 60"
$ns at 1.0 "$node_(3) setdest 200.0 250.0 80"
$ns at 1.0 "$node_(1) setdest 200.0 450.0 30"
$ns at 1.5 "$node_(2) setdest 300.0 100.0 10"
```

## Travail demandé :

- On vous demande créer des agents udp0 et l'attaché au nœud(0) avec un flux de données CBR (500 octets, toute les 10ms).
- Ensuite créer un Agent Null attaché au nœud(1)
- Créer une connexion entre les nœuds 0 et 1
- Programmer une transmission CBR qui démarre à l'instant 1 s et qui termine à l'instant 4.5
- Programmer la fin de simulation à l'instant 5.0s
- Lancer la simulation et visualiser l'exécution avec l'outil NAM.

### Analyser l'exécution:

.....

.....

.....

.....

- Modifier la taille de la file d'attente de 50 à 200

### Analyser l'exécution:

.....

.....

.....

.....

- Remettre la taille de la file d'attente à 50
- Changer la position du nœud 2 comme suit :
- Position (200, 200, 0)
- Destination 50, 50
- Vitesse de du nœud 80

### Analyser l'exécution:

.....

.....

.....

.....

## APERÇU DU FICHIER TRACE AODV

```
s 21.500275000 _0_ MAC --- 0 AODV 106 [0 ffffffff 0 800] ----- [0:255 -1:255 30 0] [0x2 1 4 [1 0] [0 10]] (REQUEST)
```

### Evènements

s : send (envoyé) r : receive (reçu) d : drop (perdu) f : forward (relayé)

### Informations de base sur les traces:

- Le nœud «\_0\_» envoie (c'est-à-dire «s») au temps «21,500275» seconde.
- Le niveau de trace est au niveau de la couche «MAC».
- Le paquet a un ID unique «0», qui contient un type de données «AODV» et a une taille de «106» octets.
- Les adresses MAC source et destination sont respectivement «0» et «ffffffff».
- Le protocole MAC suppose que le délai sur le canal sans fil est égal à zéro «0».
- Il s'agit d'un paquet IP fonctionnant sur un réseau Ethernet (c'est-à-dire «800»).

### Informations de trace IP:

- Les adresses IP source et de destination de «0» et «1», respectivement.
- Les ports pour la source et la destination sont «255».
- La durée de vie et l'adresse du nœud de saut suivant sont respectivement «30» et «0».

### Informations de trace AODV:

- Il s'agit d'un paquet RREQ étiqueté avec l'ID «0 x 2».
- Le compteur de saut est à «1» et l'ID de diffusion est à «4».
- L'adresse IP de destination et le numéro de séquence sont respectivement 1 et 0.
- L'adresse IP source et le numéro de séquence sont respectivement 0 et 10.
- La chaîne «(REQUEST)» confirme qu'il s'agit du paquet RREQ.

### Source (NS)

K. Fall et K. Varadhan, «The ns manual». 2010.

## **Travail demandé (suite TP 4):**

### **G1:**

Analyse de la perte des paquets pour les trois scenarios précédents

- 1- Tracer les courbes d'évaluation (dans la même graphique) des paquets perdus durant le temps de transmission. Ensuite, analyser le résultat.
- 2- Tracer les courbes d'évaluation (dans la même graphique) des paquets perdus sur un intervalle de temps entier [1s, 2s ... 5s]. Ensuite, analyser le résultat.

### **Rappel : superposition de plusieurs courbes**

```
gnuplot> plot "fichier.txt" using 1:2 title "RREQ" with lines, " fichier.txt" using  
1:3 title "RREP" with lines
```

## **Travail demandé (suite TP 4):**

### **G2:**

- 1- Tracer la courbe d'évaluation des paquets perdus durant le temps de transmission dans les deux premiers scénarios. . Ensuite, analyser le résultat.
- 2- Tracer deux courbes dans la même graphique (paquets de control) qui permettent de visualiser le nombre des paquets RREQ, RREP dans un intervalle de temps entier [1s, 2s ... 5s]. Ensuite, analyser le résultat.

### **Rappel : superposition de plusieurs courbes**

```
gnuplot> plot "fichier.txt" using 1:2 title "RREQ" with lines, " fichier.txt" using  
1:3 title "RREP" with lines
```