

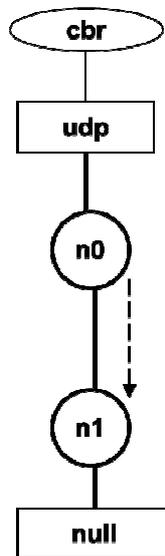
## TP 1: Introduction au Simulateur Réseaux NS2

Ce TP a pour but d'apprendre comment utiliser le simulateur Network Simulator (NS2), et en particulier comment définir le scénario de simulation avec le langage TCL ainsi que interpréter les résultats de simulation avec l'analyse des fichiers traces, mais aussi avec des outils comme *nam* (qui permet de visualiser la simulation).

### Ex. 1 :

Le premier exercice est une simulation d'une topologie simple de deux nœuds avec un lien direct (comme illustré dans la figure suivante).

Dans ce scenario, une source CBR (Constant Bit Rate) envoie des paquets de taille 500 octets (paramètre *packetSize\_*) avec un intervalle de transmission (entre un paquet et le suivant) de 0.005 secondes (paramètre *interval\_*).



**#Déclaration d'un nouveau objet Simulator (nécessaire pour la simulation)**

```
set ns [new Simulator]
```

**#Déclaration des fichiers qui vont contenir les résultats**

```
set ftrace [open trace1.tr w]
```

```
set nf [open tp1.nam w]
```

```
$ns namtrace-all $nf
```

**#La topologie du réseau : 2 nœuds et 1 lien full-duplex**

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

**#Niveau Transport : Agents UDP et Null**

```
set udp [new Agent/UDP]
```

```
set null [new Agent/Null]
```

```
$ns attach-agent $n0 $udp
```

```
$ns attach-agent $n1 $null
```

```
$ns connect $udp $null
```

**#Creation de la Source de Trafic : CBR (Constant Bit Rate) Application**

```
set cbr [new Application/Traffic/CBR]
```

```
$cbr set packetSize_ 500
```

```
$cbr set interval_ 0.005
```

```
$cbr attach-agent $udp
```

**#Procédure qui termine la simulation**

```

proc finish {} {
    global ns nf ftrace
    $ns flush-trace
    close $nf
    exec nam tp1.nam &
    exit 0
}

```

### #Faire partir la source CBR et lancer la simulation

```

$ns at 0.5 "$cbr start"
$ns at 4.5 "$cbr stop"
$ns at 5.0 "finish"
$ns run

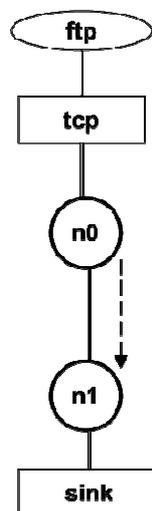
```

Une fois le script terminé (et sauvegardé, par exemple, dans un fichier « exe1.tcl »), le lancer avec ns (ns.exe exe1.tcl). En suite visualiser le résultat avec *nam* (la trace est contenue dans le fichier « tp1.nam »). Explorer les fonctionnalités de visualisation de *nam* (visualisation de la topologie, monitoring des paquets, de la capacité utilisée sur le lien etc.)

- Question 1.1 : calculer le débit moyen de transmission (en bits/secondes) de la source CBR.
- Question 1.2 : changer le paramètre *interval\_* afin que le débit de transmission de la source CBR soit égal à la moitié de la capacité du lien (c'est-à-dire, 500000 bit/s). Re-exécuter la simulation et observer le résultat avec *nam*.

### Ex. 2 :

Dans la même topologie (un seul lien), nous voulons maintenant observer comment une source TCP gère le trafic. Nous voulons tout d'abord simuler le scénario illustré dans la figure qui suit, avec une application FTP (File Transfer Protocol, c.à.d. une application qui gère la transmission d'un grand fichier, et donc d'une très longue séquence de paquets).



Changer le script de l'exercice 1 pour simuler ce scénario avec une source FTP/TCP. Les commandes qui suivent sont utiles pour créer le scénario :

```

set tcp [new Agent/TCP]
set tcpsink [new Agent/TCPSink]
$ns attach-agent $n0 $tcp
$ns attach-agent $n1 $tcpsink
$ns connect $tcp $tcpsink
set ftp [new Application/FTP]
$ftp attach-agent $tcp

```

Il faut, bien sûr, changer aussi les deux lignes suivantes :

```
$ns at 0.5 "$cbr start"  
$ns at 4.5 "$cbr stop"
```

... car maintenant la source de trafic s'appelle « ftp », et pas « cbr ».

Une fois le script terminé, le lancer avec ns et visualiser le résultat avec nam.

- Question 2.1 : Ecrire le script TCL utilisé pour simuler ce scénario. Comparer les résultats avec ce qu'on a observé avec la source CBR.

Changer le script pour simuler une source Telnet/TCP. Les commandes qui suivent sont utiles pour créer le scénario :

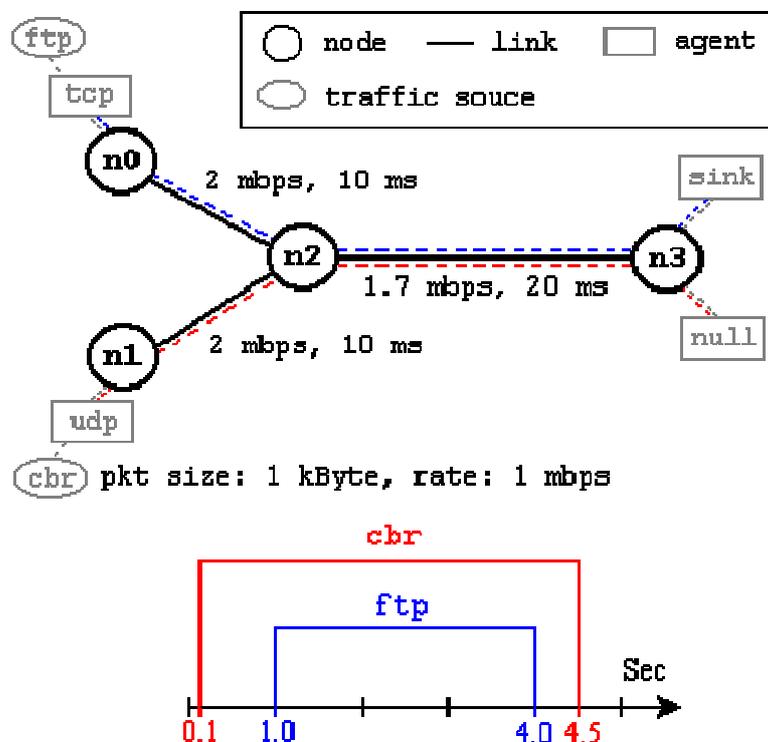
```
set telnet [new Application/Telnet]  
$telnet attach-agent $tcp
```

### Ex. 3 :

Simuler le réseau illustré dans la figure qui suit. La taille du buffer entre n2 et n3 est de 10 paquets (commande `$ns queue-limit $n2 $n3 10`).

Colorier les deux flux différemment (pour ensuite les distinguer dans la visualisation avec nam), grâce à des commandes comme :

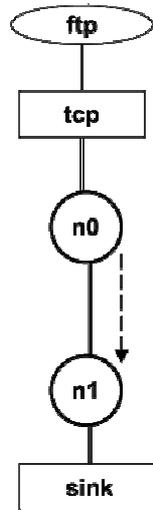
```
$ns color 1 Blue  
$agent1 set class_1
```



- Question 3.1 : Ecrire le script TCL utilisé pour simuler ce scénario.

### Ex. 4 : TCP

Nous considérons maintenant la même topologie de l'exercice 2, avec un seul lien full-duplex et une source FTP/TCP.



Cette fois, nous utilisons l'agent `Agent/TCP/RFC793edu`, qui permet d'étudier une connexion TCP en faisant varier ses variables d'état et paramètres. Par exemple, les commandes suivantes permettent de créer l'agent TCP RFC793edu, de fixer la *Receiver Window* (RCVWND) à 1 et la longueur des paquets (MSS = Maximum Segment Size) à 500 bytes :

```
set TCPedu0 [new Agent/TCP/RFC793edu]
$TCPedu0 set window_ 1
$TCPedu0 set packetSize_ 500
```

La capacité du lien n0-n1 est de 10 Mbit/s, le délai de propagation de 4 ms.

- Question 4.1: après avoir terminé et exécuté le script, observer (à travers *nam*) la transmission de la source TCP.
- Question 4.2 : changer maintenant la dimension de la RCVWND (paramètre *window\_*), à 5, 10, 15, et 20, et observer le résultat de la simulation en utilisant *nam*.
- Question 4.3 : pour avoir une transmission continue, quelle valeur de la fenêtre RCVWND est-il nécessaire ?