LANCE Florian

MATHEOSSIAN Dimitri

NAVILLE Vincent

# Exercice 2 – Premiers Pas

*1. A quel moment est appelée la procédure fnish ? que fait-elle ?*

Au bout de 5 secondes elle est appelée.

Un fichier de trace est créé -> out.nam

*2. Quelle est la différence entre TCP et UDP? Et dans leurs implémentations NS ?*

TCP : liaison assurée par des ACK

UDP : liaison non sécurisée

Sur NS on a :

set tcp [new Agent/TCP]

set udp [new Agent/UDP]

*3. Sur quelle couche d'application est utilise le flux FTP ?*

FTP est associé à TCP.

*4. Pourquoi avant de communiquer, les agents doivent être attachés ?*

Ils sont attachés à un nœud pour avoir un point de départ (pour ensuite utiliser les liaisons)

*5. Quelles est la taille par défaut d'un paquet CBR ?*

$cbr set packet\_size\_ 1000

Donc 1000 bits pour un paquet CBR

# Exercice 3 - Topologie du réseau et routage dynamique

*1. A quoi correspondent les petits paquets qu'on peut voir sur NAM. A quels moments sont-ils échangés ? Pourquoi ?*

Ils sont échangés au début à cause du protocole de routage "Distance vector".

Une table de routage est transmise aux nœuds voisins et permet de communiquer les modifications de topologie du réseau.

*2. Etude d'une propriété de k-connexités : définir une topologie en grille de 9 nœuds (de n1 à n9).*

*(a) Combien y a-t-il de routes différentes entre n1 et n9 ? Quel est le chemin le plus court ?*

Il y a 10 chemins si on ne repasse pas par les mêmes nœuds.

Chemin le plus court :

1 - 2 - 3

| | |

4 - 5 - 6

| | |

7 - 8 - 9

N1-2-3-6-9

N1-2-5-6-9

N1-2-5-8-9

...

Au minimum on aura 4 transitions.

*(b) Même question pour n3 et n8.*

Chemin le plus court :

N3-N6-N9

*(c) Lancez un flux constant entre n3 et n8.*

*(d) Donnez Le pire scenario que peut rencontrer l'algorithme DV pour la route n3 - n8 :*

N3-N2-N1-N4-N5-N6-N7-N8

*3. Refaire la question précédente pour une grille torique (tous les sommets du graphe auront un degré égale à 4).*

*(a) Combien y a-t-il de routes différentes entre n1 et n9 ? Quel est le chemin le plus court* ?

Il y a 30 chemins si on ne repasse pas par les mêmes nœuds.

Chemin le plus court :

1 - 2 - 3 - 1

| | | |

4 - 5 - 6 - 4

| | | |

7 - 8 - 9 - 7

N1-N7-N9

N1-N3-N9

Au minimum on aura 3 transitions.

*(b) Même question pour n3 et n8.*

C'est exactement le même nombre que dans la question *(a)*.

Chemin le plus court :

N3-N2-N8

N3-N9-N8

On peut y accéder en 2 transitions.

*(c) Lancez un flux constant entre n3 et n8.*

*(d) Donnez Le pire scenario que peut rencontrer l'algorithme DV pour la route n3 - n8.*

Pareil que dans la question *2)* si on ne repasse pas par les mêmes points :

N3-N2-N1-N4-N5-N6-N7-N8

# Exercice 4 - Echanges de messages

*Donnez une description détaillée de tout ce qui se passe dans cette simulation.*

# Envoie d'un premier message de n0 vers n1 de type ping(42)

$ns at 0.1 "$udp0 send 724 ping(42)"

#n1 reçoit "ping(42)". Il envoie alors un paquet de taille 100 avec un countdown d'une valeur de 5 a n0. Lorsque n0 le reçoit, la valeur du paquet décrémente de 1, N0 le renvoie a 1, et ainsi de suite, jusqu’à ce que la valeur du countdown soit à zéro.

Quand le countdown se fini, N0 envoie le message "ignore this message please" dont le MSS est 500

$ns at 0.2 "$udp1 send 100 countdown(5)" $ns at 0.3 "$udp0 send 500 {ignore this message please}"

N1 envoie de nouveau un message "ping" de MSS 828 et N0 lui répond "pong" de MSS 100

$ns at 0.4 "$udp1 send 828 {ping (12345678)}"

$ns at 1.0 "finish"

C'est la fonction ci-dessous qui renvoie un pong après réception d'un ping, et qui décrémente le countdown.

if {[regexp {ping \*\(([0-9]+)\)} $data entirematch number]} {

$self send 100 "pong($number)"

} elseif {[regexp {countdown \*\(([0-9]+)\)} $data entirematch number] && $number > 0 } {

incr number -1

$self send 100 "countdown($number)"

}

# Exercice 5 - Réseaux sans fil

On ajoute dans le .tcl :

source "setdest" #équivalent à un include (voir le .cpp pour la fonction)