المعهد الوطني للبريد والمواصلات
ⵓⵔⵣⵓⵔ ⵓⵏⵎⵙⵙⵓ ⵏ ⵜⵓⵏⵓⵖⵜ ⴰ ⵙⵎⵔⵓⵍⴻ
Institut National des Postes et Télécommunications

Rapport de Projet

Ateliers d'ingenierie

Simmulation du reseau Token Ring

Meryam el qemary

Safae el ati

Hind Zrigou

Rapport de Projet

L'objectif :

Le but de ce projet c'était la simulation du réseau Token Ring , en exploitant les notions vue aux cours et aux ateliers à savoir la manipulation des pipes pour gérer la synchronisation ,l'utilisation des logiciels de gestion de versions **(git) ,** et l'hébergement sur GitHub.
On a travaillé ce projet en groupe de trois personnes et grâce à git on a pu organiser le travail entre nous , En fin on a publié notre projet dans GitHub .

Table des matières

0.1 C'est quoi un réseau Token Ring 3
0.2 La simulation de réseau avec Les pipes 5
0.3 Notre Feedback 6
0.3.1 Les diffécultés qu'on a trouvé 7

0.1 C'est quoi un réseau Token Ring

Le **réseau Token Ring** est une architecture de réseau local (LAN) dans laquelle les ordinateurs ou périphériques sont connectés en boucle circulaire (anneau)

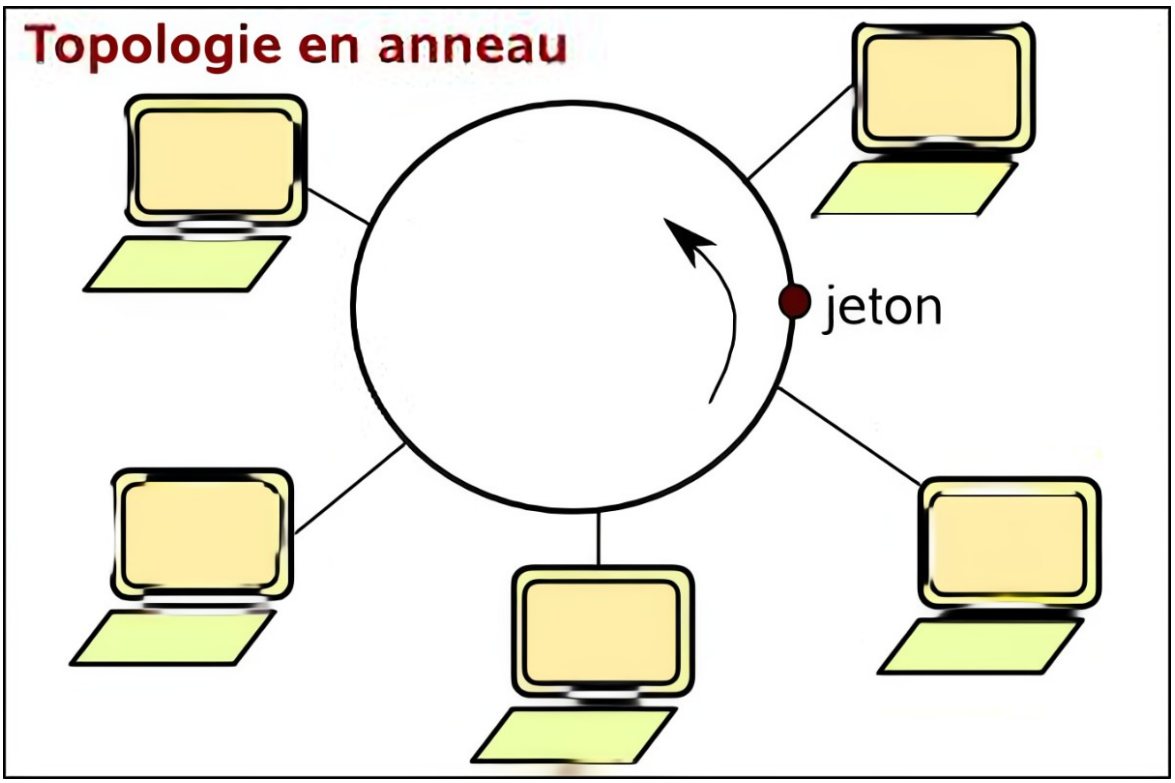


FIGURE 1 – La topologie en anneau

1. Structure et connexion

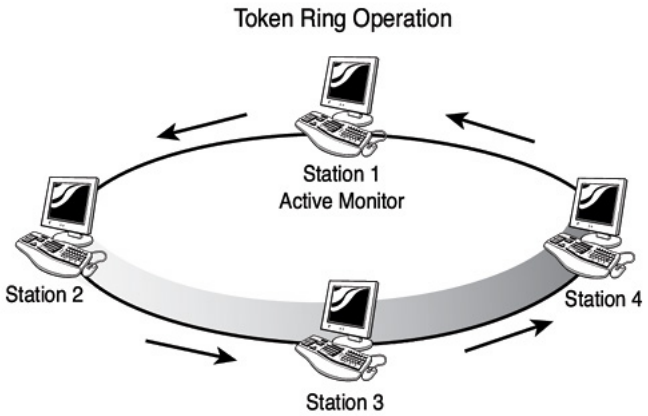
— Chaque appareil est connecté à deux autres, formant un cercle ou "anneau".
— Les données circulent dans une seule direction , de manière séquentielle, d'un appareil à l'autre.

2. Transmission des données

— Un **"jeton" (token)** est utilisé pour gérer l'accès au réseau.
— Le jeton est un petit paquet de contrôle qui circule en continu sur l'anneau.
— Un appareil peut transmettre des données **uniquement lorsqu'il possède le jeton** .

3. Étapes du fonctionnement

1. **"jeton" (Jeton libre :)** Le jeton circule librement sur l'anneau.
2. **"jeton" (Prise du jeton :)** Lorsqu'un appareil veut envoyer des données, il capture le jeton.
3. **"jeton" (Transmission des données :)**
   — L'appareil modifie le jeton pour y inclure les données et l'adresse du destinataire.
   — Le paquet de données circule sur l'anneau jusqu'à atteindre le destinataire.
4. **Accusé de réception :** Le destinataire copie les données et marque le paquet comme reçu.
5. **Libération du jeton :** Une fois les données transmises, le jeton est libéré et peut être utilisé par un autre appareil.



0.2 La simulation de réseau avec Les pipes

```c
#include <sys/wait.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
int main() {
    int nbr_noeuds= 5; // Nombre de nœuds dans le reseau
    int jeton= 1; // Jeton initial
    int pipes[5][2];  // Pipes entre les nœuds

    pid_t pid;
```

FIGURE 2 – Importation des bibliothéques et déclaration des variables

Commentaire :

On a importé les bibliothéques nécessaires à savoir : string.h , sys/wait.h pour manipuler les chaines de caracteres et les pipes

```c
    // Créer les processus pour chaque nœud
    for (int i = 0; i < nbr_noeuds; i++) {
        pid = fork();

        if (pid < 0) {
            printf("Erreur lors du creation du fils");
            // Echoue au cours de la création
        }


        if(pid == 0) {  // Processus enfant (nœud)
            int jeton_reçus;

            while (1) {
                // Lire le jeton depuis le pipe d'entrée
                read(pipes[i][0], &jeton_reçus, sizeof(jeton_reçus));
                printf("Nœud %d : Jeton reçu = %d\n", i, jeton_reçus);

                // Simuler une action du nœud
                if (jeton_reçus > 0) {
                    printf("Nœud %d : Utilise le jeton.\n", i);
                }

                // Passer le jeton au prochain nœud
                int noeud_suivant = (i + 1) % nbr_noeuds;
                write(pipes[noeud_suivant][1], &jeton_reçus, sizeof(jeton_reçus));
                // Pause pour ralenti l'exécution
                sleep(1);
            }
            exit(0);
        }
    }
```

FIGURE 3 – création des processus et simulation de l'action des noeuds

Commentaire :

1 - On a procédé par la méthode fork () pour créer des processus qui vont jouer le role des stations de notre réseau . 2 - Dans chaque processus fils , on a ouvrit le pip (dans lequel on passe le jeton) qui lui correspond en implémentant le mécanisme de synchronisation entre les noeuds . 3 - Le passage de jeton d'un nœuds à l'autre se fait par l'écriture dans le pip (write)

N.B : On a utilisé l'opérateur modulo pour que le passage de jeton se fait dans une boucle comme montre le nom du réseau ;

```
45    // Processus parent (nœud initial)
46    printf("Lancement du réseau Token Ring.\n");
47    write(pipes[0][1], &jeton, sizeof(jeton));  // Envoyer le jeton au premier nœud
48    // Attendre les processus enfants (inactif ici car la simulation est infinie)
49    for (int i = 0; i < nbr_noeuds; i++) {
50        wait(NULL);
51    }
52    return 0;
53    }
```

FIGURE 4 – Lancement de réseau Token Ring

Commentaire :

Pour passer le jeton dans une boucle on a donné la main au processus numéro ' 0 ' qui simmule le noeud initial

0.3 Notre Feedback

Ce projet était une exelente opportunité pour nous , à fin de nous nous familiariser avec les commandes de git à savoir comment tracker un fichier , comment valider les changement et comment les pousser . Aussi la publication dans gitHub nous a donné une idée sur comment partager nos projets avec les autres et surtout comment organiser le travail en groupe , parce qu'on sait qu'en tant que des ingénieurs on devrait travailler - presque toujours - en groupe , raison pour laquelle nous vous remercions pour nous donner cette chance

0.3.1 Les diffécultés qu'on a trouvé

- Dans un premier lieu , on a voulu instancier les noeuds à l'aide de l'orienté objet tout en définissant des classes ,des attributs et des méthodes , mais le probleme c'était au niveau de synchronisation raison pour laquelle on a procédé par les pipes .