



المعهد الوطني للبريد والمواصلات
የኢትዮጵያ ፖስታና ቴሌኮሙኒኬሽን ቢሮ
Institut National des Postes et Télécommunications

INSITUT NATIONAL DES POSTES ET TELECOMMUNICATIONS

ATELIER D'INGÉNIERIE
RAPPORT

Simulation d'un Réseau Token Ring

Realisé par :

Baba SALMANE

Najmi YASSIN

Ouarach ABDERRAHMANE

Encadrant :

Bensaid HICHAM

21 janvier 2025

Table des matières

1	Introduction	2
2	Description du projet	2
2.1	Contexte	2
2.2	Objectifs	2
3	Implémentation	2
3.1	Structure du code	2
3.2	Code source	3
4	Résultats	4
4.1	Dépôt GitHub	4
4.2	Initialisation du projet	5
4.3	Push vers GitHub	5
5	Conclusion	5

1 Introduction

Ce rapport présente la simulation d'un réseau Token Ring réalisée dans le cadre du projet d'ingénierie. L'objectif de ce projet est de comprendre le fonctionnement d'un réseau Token Ring en implémentant un simulateur en langage C. Ce simulateur permet de visualiser la circulation du jeton entre les nœuds et la transmission des données.

2 Description du projet

2.1 Contexte

Un réseau Token Ring est une topologie de réseau où les stations sont connectées en anneau. Un jeton circule entre les stations, et seule la station qui possède le jeton peut transmettre des données. Ce projet consiste à simuler ce comportement en utilisant le langage C.

2.2 Objectifs

Les objectifs du projet sont les suivants :

- Implémenter un simulateur de réseau Token Ring.
- Permettre de configurer le nombre de nœuds dans le réseau.
- Visualiser la circulation du jeton et la transmission des données.

3 Implémentation

3.1 Structure du code

Le code est structuré autour de deux principales entités :

- **Node** : Représente une station dans le réseau. Chaque nœud a un identifiant unique et peut posséder le jeton.
- **Pipes** : Des tubes (pipes) sont utilisés pour simuler la communication entre les nœuds. Chaque nœud lit et écrit dans des pipes pour passer le jeton.

3.2 Code source

Voici le code source implémentant le réseau Token Ring :

Listing 1 – Extrait du code source

```

//@Salmane BABA - Abderrahmane OUARACH - Yassine NAJMI
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#include <time.h>

void node(int nodeid,int readfd,int writefd) {
    char token;
    while(1) {
        read(readfd, &token,1);
        printf("Noeud %d: jeton re u\n",nodeid);
        if(rand()%2) {
            printf("Noeud %d : transmission de trame\n",nodeid);
            sleep(rand()%3+1);
        }
        printf("Noeud %d : passe le jeton au noeud suivant \n",nodeid);
        write(writefd,"T",1);
    }
}

int main() {
    int numnodes;
    printf("Entrez le nombre de noeuds dans le r seau : ");
    scanf("%d",&numnodes);

    int pipes[numnodes][2];
    for(int i=0;i<numnodes; i++) pipe(pipes[i]);

    srand(time(NULL));
    for(int i=0; i<numnodes; i++) {
        int pid=fork();
        if(pid==0) {
            for(int j=0; j<numnodes; j++) {
                if(j!=i) close(pipes[j][0]);
                if(j!=(i+1)%numnodes) close(pipes[j][1]);
            }
            node(i+1,pipes[i][0],pipes[(i+1)%numnodes][1]);
            exit(0);
        }
    }

    write(pipes[0][1],"T",1);
    for(int i=0; i<numnodes; i++) close(pipes[i][0]), close(pipes[i][1]);
    ;
    for(int i=0; i<numnodes; i++) wait(NULL);
    return 0;
}

```

Le code complet est disponible dans GitHub ([Lien vers le dépôt GitHub](#))

4 Résultats

Le simulateur fonctionne comme prévu. Le jeton circule entre les nœuds, et chaque nœud peut transmettre des données lorsqu'il possède le jeton. Voici un exemple de sortie du programme :

```
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ gcc ring.c -o ring
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ ./ring
Entrez le nombre de noeuds dans le réseau : 5
Noeud 1 : jeton reçu
Noeud 1 : passe le jeton au noeud suivant
Noeud 2 : jeton reçu
Noeud 2 : passe le jeton au noeud suivant
Noeud 3 : jeton reçu
Noeud 3 : passe le jeton au noeud suivant
Noeud 4 : jeton reçu
Noeud 4 : passe le jeton au noeud suivant
Noeud 5 : jeton reçu
Noeud 5 : passe le jeton au noeud suivant
Noeud 1 : jeton reçu
Noeud 1 : transmission de trame
Noeud 1 : passe le jeton au noeud suivant
Noeud 2 : jeton reçu
Noeud 2 : transmission de trame
Noeud 2 : passe le jeton au noeud suivant
Noeud 3 : jeton reçu
```

FIGURE 1 – Exemple de sortie du programme montrant la circulation du jeton.

4.1 Dépôt GitHub

Le projet a été versionné sur GitHub. Voici une capture d'écran du dépôt :

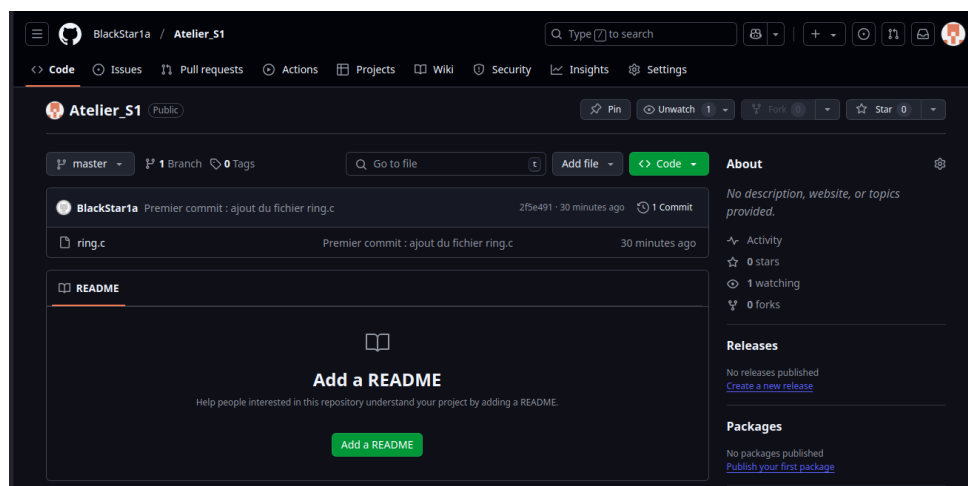


FIGURE 2 – Capture d'écran du dépôt GitHub montrant les fichiers et les commits.

Lien vers le dépôt GitHub

4.2 Initialisation du projet

Le projet a été initialisé avec Git, et le fichier source a été ajouté et commité. Voici une capture d'écran du processus d'initialisation :

```
blackstar@blackstar:/media/blackstar/Disk/Documents$ mkdir ring
blackstar@blackstar:/media/blackstar/Disk/Documents$ cd ring
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /media/blackstar/Disk/Documents/ring/.git/
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ vim ring.c
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ git add ring.c
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ git commit -m "Premier commit : ajout du fichier ring.c"
[master (root-commit) 2f5e491] Premier commit : ajout du fichier ring.c
 1 file changed, 91 insertions(+)
 create mode 100644 ring.c
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ git remote add origin https://github.com/BlackStar1a/Atelier_S1.git
```

FIGURE 3 – Initialisation du projet avec Git.

4.3 Push vers GitHub

Le code a été poussé vers le dépôt distant sur GitHub. Voici une capture d'écran du processus de push :

```
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ git remote add origin https://github.com/BlackStar1a/Atelier_S1.git
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$ git push -u origin master
Username for 'https://github.com': BlackStar1a
Password for 'https://github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.27 KiB | 1.27 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/BlackStar1a/Atelier_S1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
blackstar@blackstar:/media/blackstar/Disk/Documents/ring$
```

FIGURE 4 – Push du code vers GitHub.

5 Conclusion

Ce projet a permis de comprendre le fonctionnement d'un réseau Token Ring en implémentant un simulateur en langage C. Le code est fonctionnel et permet de visualiser la circulation du jeton et la transmission des données. Des améliorations pourraient être apportées, comme la gestion des pannes ou l'ajout d'une interface utilisateur.