



**TP Réseaux locaux industriels
Commande en réseau**

Valentin DOSIAS, Maxence NEUS

Novembre 2021

Contents

1	Test de l'application exemple	2
2	Version sans mode "Réseau CAN"	2
2.1	Token-Bus	2
2.2	CSMA/CD	3
3	Réseau temps réel ?	4
4	Version avec orange	5
5	Robuste aux perturbations ?	5

1 Test de l'application exemple

2 Version sans mode "Réseau CAN"

2.1 Token-Bus

```
int main(int argc, char *argv[])
{
    int valId, capteur1, capteur2, msg;
    valId=litId();
    capteur2=litEtatCapteur();
    etatVert();
    int fr =0;
    while(1)
    {
        capteur1=capteur2;
        capteur2=litEtatCapteur();
        if ((capteur1==1) && (capteur2==0))// front descendant
        {
            fr = 1;
            if (valId!=7){
// Permet aux trains d'arriver en boucle
                etatRouge();
            }
        }
        msg=ecouteReseau();

        if (msg==valId+1+10) etatVert();
        if (msg%10 == valId-1) {
            emetMsg(fr*10+valId);
            fr =0;
        }
    }
}
```

Figure 1: Code du Token Ring

2.2 CSMA/CD

```
int main(int argc, char *argv[])
{
    int valId, capteur1, capteur2, msg;
    valId=litId();
    capteur2=litEtatCapteur();
    etatVert();
    while(1)
    {
        capteur1=capteur2;
        capteur2=litEtatCapteur();
        if ((capteur1==1) && (capteur2==0))// front descendant
        {
            emetMsg(valId);
            while (ecouteReseau()==-2) {
//Attend qu'il n'y ai pas de collision
                attendMs(valId*20);
                emetMsg(valId);
            }
            if (valId!=7){
// Permet aux trains d'arriver en boucle
                etatRouge();
            }
        }
        msg=ecouteReseau();
        if (msg==valId+1) etatVert();
    }
}
```

Figure 2: Code du CSMA/CD

3 Réseau temps réel ?

Avec l'algorithme du Token Ring, on peut dire que le pire des cas correspond à un noeud qui veut émettre sur le réseau alors que le Token est actuellement au noeud suivant, dans ce cas le noeud devra attendre que le Token fasse un tour complet pour lui revenir. On peut prédire ce temps à partir de l'architecture du système. Il est donc bien temps réel.

Pour l'algorithme CSMA/CD, il est possible qu'un noeud voulant émettre se voit bloqué par des collisions répétées sur le réseau, il est impossible de borner le nombre de ces collisions et il est donc également impossible de borner le temps de réponse. Il n'est donc pas temps réel.

Pour notre application, l'arrivée d'un train n'a pas besoin de se passer avant une date précise, dans la limite du raisonnable. Cette application ne nécessite donc pas d'être temps réel.

4 Version avec orange

```
int main(int argc, char *argv[])
{
    int valId, capteur1, capteur2, msg;
    valId=litId();
    capteur2=litEtatCapteur();
    etatVert();
    int fr =0;
    int cases[2] = {0, 0};
    while(1) {
        capteur1=capteur2;
        capteur2=litEtatCapteur();
        if ((capteur1==1) && (capteur2==0)) { // front descendant
            if (valId != 7) cases[0]++;
            fr = 1;
        }
        msg=ecouteReseau();
        // Mise a jour des compteurs pour les cases n+1 et n+2
        if (msg==10+valId+1)
        {
            cases[0]--;
            cases[1]++;
        }
        if (msg==10+valId+2) cases[1]--;
        if (cases[0]==1) etatRouge(); // Un train dans la case n+1
        else if (cases[1]==1 && cases[0]==0) etatOrange(); // Un train dans la case n+2
        else etatVert(); // pas de trains proche
        if (msg%10 == valId-1 ) {
            emetMsg(fr*10+valId);
            fr =0;
        }
    }
}
```

Figure 3: Code inclusion du Orange

5 Robuste aux perturbations ?

Pour gérer les perturbations, on ecoute le réseau juste après l’envoi et si on n’entend rien, alors on renvoie le message.

```

int main(int argc, char *argv[])
{
    int valId, capteur1, capteur2, msg;
    valId=litId();
    capteur2=litEtatCapteur();
    etatVert();
    int fr =0;
    int cases[2] = {0, 0};

    while(1)
    {
        capteur1=capteur2;
        capteur2=litEtatCapteur();
        if ((capteur1==1) && (capteur2==0))// front descendant
        {
            if (valId != 7) cases[0]++;
            fr = 1;
        }
        msg=ecouteReseau();
        // Mise a jour des compteurs pour les cases n+1 et n+2
        if (msg==10+valId+1)
        {
            cases[0]--;
            cases[1]++;
        }
        if (msg==10+valId+2) cases[1]--;

        if (cases[0]==1) etatRouge(); // Un train dans la case n+1
        else if (cases[1]==1 && cases[0]==0) etatOrange(); // Un train dans la case n+2
        else etatVert(); // pas de trains proche

        if (msg%10 == valId-1 ) {
            emetMsg(fr*10+valId);
            // Si le message n'est pas parti on attend et on le renvoie
            while (ecouteReseau()==-1) {
                attendMs(valId);
                emetMsg(fr*10+valId);
            }
            fr =0;
        }
    }
}

```

Figure 4: Ajout de la gestion des perturbations