BTS INFORMATIQUE ET RESEAUX POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES

Session 2010

EPREUVE E.4

Etude d'un Système Informatisé

AIDE A L'EXPLOITATION D'UN TRAMWAY ET INFORMATION DES VOYAGEURS

CORRIGE

B. ANALYSE DU CONTEXTE

B.1. Topologie du réseau du tramway

B1.1. Répondre aux questions suivantes sur la topologie du réseau du tramway, en cochant pour chaque question la case « Oui » ou « Non ».

questions	*Oui*	*Non*
Un parcours dispose d'exactement deux terminus	*	
Une ligne dispose d'exactement deux destinations		*
Un chaînage d'arrêts est la même chose qu'un parcours	*	
Une voie est un chaînage de tronçons orientés	*	
Plus de deux arrêts peuvent être présents sur un tronçon		*
L'ordre des arrêts desservis est indiqué par un parcours	*	
Les deux stations du tronçon t36 sont Isard et Cascade	*	
Il existe sept destinations	*	
Un itinéraire est identifié par les étiquettes de recalage		*

B1.2. Quel est le sens du parcours BUISSON-ACANTHE (ID_Parcours = 9) ?

Nb : attention à l'utilité des identifiants !

Lecture de la table PARCOURS : *Retour*

B1.3. A quoi sert le champ « ordre » dans la table DESSERT, donner un exemple pour PIVOINE-BUISSON (ID_Parcours = 12) ?

Les PARCOURS desservent les arrêts dans un certain ORDRE.

DESSERT		
ID_Parcours	ID_Arret	Ordre
12	21	1
12	22	2
12	23	3

Soit : ID_PArcours = 12 :

Arrêt 21 PIVOINE puis Arrêt 22 FONTAINE puis Arrêt 23 BUISSON

B1.4. En vous basant sur les parcours « ID_Parcours 4 » et « ID_Parcours 13 » compléter la table DESSERT « ID_Parcours 15 ».

DESSERT		
ID Parcours	**ID Arr**	Ordre
…	…	…
4	*4*	*1*
4	*5*	*2*
4	*6*	*3*
4	*3*	*4*
4	*7*	*5*
4	*8*	*6*

DESSERT(suite)		
13	*8*	*1*
13	*7*	*2*
13	*3*	*3*
13	*6*	*4*
13	*5*	*5*
13	*4*	*6*
14	*13*	*1*
14	*12*	*2*

DESSERT(suite)		
14	*11*	*3*
14	*10*	*4*
…	…	…
15	**27**	**1**
15	**26**	**2**
15	**25**	**3**
15	**24**	**4**
15	**10**	**5**

B.2. Système d'Exploitation Embarqué (SEE)

B2.1. Le système permettant de détecter la rotation (capteur + roue crénelée) peut-il être assimilé à un codeur incrémental ou un codeur absolu ? (justifier votre réponse)

Un codeur incrémental. En effet, chaque détection d'un créneau de la roue correspond à un incrément de distance. En comptant les incréments, on obtient une distante relative au début du comptage.

B2.2. Calculer la distance parcourue par le tramway entre deux impulsions générées par le capteur à effet Hall.

Distance = (π x Diamètre) / Nb créneau
= 3,14 x 0,6 / 64
= 0,0296 m = 29,6 mm

B2.3. Sachant que le compteur 32 bits représente une valeurs signée, donner les positions odométriques extrêmes (en m) que peut calculer le boitier odomètrique,

Soit D la distance entre 2 impulsions, les valeurs extrêmes sont :
mini : $(-1) \times 2^{31} \times D$
maxi : $(2^{31} - 1) \times D$
soit approx $+/-2^{31} \times 0{,}0296 = 63{,}6.10^{6}$m

B.3. Borne d'Information Voyageurs (BIV)

B3.1. Justifier le choix de l'afficheur SX502 – 440/03/ 2R/131/5A – M0 (Annexe 7).

440 : 4 lignes de 40 caractères
03 : hauteur des caractères : 33/66/75mm
2R : LED Rouge pour application extérieure
Cela correspond bien aux spécifications demandées pour les BIV (voir sujet paragraphe A5)

B3.3. Déterminer les fontes de caractères pouvant être utilisées.

Les fontes « Acala7 » et « Acala7 extented » permettent d'avoir des caractères de 33mm de hauteur.

B. ORGANISATION DES DONNÉES

C1.1. Dans la liste des arrêts, l'arrêt 2 est manquant. Coder la balise « arret » correspondante

```
<arret id_arret='2' nom='sentier' />
```

C1.2. Quel est l'élément racine du fichier XML fourni ?

Element "référentiel_Embarque", ouvert ligne 3

C1.3 En vous aidant de la table « LIGNE » présentée dans l'Annexe 2 : « Topologie et référentiel SAE », proposer un codage XML de la classe « LIGNE » qui respecte les règles de l'encadré 1 : Règles de codage en XML du sujet.

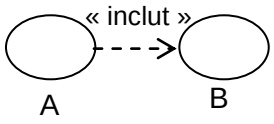
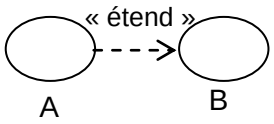
```
<liste_lignes>
  <ligne ID_Ligne='97' mnemo='A' libelle='A'/>
  <ligne ID_Ligne='98' mnemo='B' libelle='B'/>
  <ligne ID_Ligne='99' mnemo='C' libelle='C'/>
```

```
</liste_lignes>
```

C. CONCEPTION DU SYSTÈME EMBARQUÉ

D.2. Conception générale du SEE

D2.1. Quelle est la différence entre « inclut » et « étend » dans le diagramme de cas d'utilisation de la Figure 3?

Relation	Signification
	*Un cas d'utilisation A inclut un cas d'utilisation B si le comportement décrit par le cas A inclut le comportement du cas B. Lorsque A est sollicité, B l'est obligatoirement, comme une partie de A.*
	*Un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Exécuter B peut éventuellement entraîner l'exécution de A. Contrairement à l'inclusion, l'extension est optionnelle.*

D2.2. Compléter, dans le document réponse, les types de relations de 1 à 3 entre classes et la cardinalité en 4 du Diagramme des classes de la figure 5. Consulter pour cela l'Annexe 1.

N° de relation	***Type de relation***
1	*Composition (ou agrégation)*
2	*Composition (ou agrégation)*
3	*Héritage (ou généralisation)*

Cardinalité de 5 = *2*

D2.3. Relier par un trait les événements, l'activité et les actions proposés avec les numéros correspondants de 1 à 5 du diagramme d'état de la Figure 5.

	N°	***Intitulé***
Evénements	***1***	*Étiquette non reconnue*
	2	*Étiquette lue = Aval*
	3	*Étiquette non reconnue*
Activité	***4***	*Localisation tramway*
Actions	***5***	*Arrêt de l'odomètre*
	6	*Recalage odomètre*

D2.4. Quels sont les objets et les fonctions membres qui réalisent les actions en sortie et les activités de l'état « En ligne » du diagramme d'état de la Figure 6 ?

Action/activité	***Classe***	***Fonction membre***
Obtention n° étiquette	*LecteurEtiquettes*	*FournirNumeroEtiquette*
Obtention trame odomètre	*Odometre*	*DelivrerDonneesOdometre( )*
Localisation tramway	*Localisateur*	*LocaliserTramway ( )*
Envoi localisation	*Localisateur*	*EnvoyerLocalisationTramway()*
Arrêt de l'odomètre	*Odometre*	*Arreter()*
Recalage odomètre	*Odometre*	*Recaler()*

D2.5. Compléter la matrice Événement/État qui correspond au diagramme d'état Figure 6 et qui montre la transition qui s'opère quand un événement se produit lors d'un état donné.

État / *Événement*	*Début*	*En ligne*	*Arrêt*	*Hors Ligne*	*Panne*	*Fin*
EtiquetteLue = sortieTerminus	*En ligne*					
1s écoulee		***En ligne***	*Arrêt*	***HorsLigne***		
10s écoulées		***En ligne***	*Arrêt*	*HorsLigne*		
EtiquetteLue = Amont		***Arret***		***Arret***		
EtiquetteLue = Aval			*En ligne*	*En ligne*		
Etiquette Non Reconnue		***HorsLigne***	*HorsLigne*	*HorsLigne*		
EtiquetteLue = entréeTerminus		***Fin***		*Fin*		
Etiquette Non Lue Depuis 10min		***Panne***		*Panne*		
Etiquette Non Lue Depuis 20min			***Panne***			
Panne Signalée					*Fin*	

D. PROGRAMMATION DU SYSTÈME EMBARQUÉ

E.1. Programmation de la classe Odomètre

E1.1. Compléter le constructeur de la classe `Odometre`

```
Odometre:: Odometre(CPortSerie  *port, short diamRoue, short nbCren){
      //affecter données privées

      this-> diametreRoue = diamRoue;
      this-> nbCreneaux = nbCren
      this->serieBoitierOdo = port;
      //configurer port série
      this->serieBoitierOdo ->ouvrir(_PORTSERIE_CFG) ;
      // étalonner le module odomètre
      this->Etalonner();
}
```

E1.2. Ecrire dans la ligne de code permettant à la classe `Odometre` de lire une trame du boîtier odomètre et de la placer dans la donnée membre privée `trameBrute`

```
SerieBoitierOdo->RecevoirTrame(trameBrute) ;
```

E1.3. En vous aidant de l'annexe 3, compléter dans le document réponse cette série de « `#define` ».

```
#define I_STX    0  // indice du tableau ou l'on trouve STX
#define I_VL     1  // indice du tableau ou l'on trouve vitesseLineaire
#define I_P_ODO  8  // indice du tableau ou l'on trouve positionOdometrique
#define I_T_ROUL 16 // indice du tableau ou l'on trouve tempsRoulage
#define I_T_IMMO 22 // indice du tableau ou l'on trouve tempsImmobilisation
#define I_STX    28 // indice du tableau ou l'on trouve ETX
```

E1.4. Compléter le code de la fonction membre
`void Odometre::extraireDonneesOdo ().`

```
void Odometre::extraireDonneesOdo (){
     donneesOdometre.vitesseLineaire = atof(&trameBrute[I_VL]);
     donneesOdometre.positionOdometrique=
atof(&trameBrute[I_P_ODO]);
     donneesOdometre.tempsRoulage=
atoi(&trameBrute[I_T_ROUL]);
     donneesOdometre.tempsImmobilisation =
atoi(&trameBrute[I_T_IMMO]);
}
```

E.2. Programmation de la classe Lecteur étiquettes

Question E2.1. Coder en C++ la fonction membre privée `Char2NumeroEtiquette`.

Version courte :

```
numeroEtiquette_t Char2NumeroEtiquette (char cfort, char cfaible){
      return ((numeroEtiquette_t)cfort << 8 | cfaible);
}
```

Version longue :

```
numeroEtiquette_t Char2NumeroEtiquette (char cfort, char cfaible){
      numeroEtiquette_t result ;
      result = cfort ;
      result = result << 8 ;
      result = result | cfaible ;
      return (result);
}
```

Question E2.2. Coder en C++ la fonction membre privée `RecevoirTrame`

Solution utilisant l'opérateur >> :

```
bool RecevoirTrame(char* tampon) {
    const char STX = 02;
    int i;
    char check;
    // reception et test premier caractère
    liaisonBalogh >> tampon[0];
    if (tampon[0] != STX) return false;
    // reception 5 caractères suivants – calcul check à la volée
    check = tampon[0];
    for(i=1; i<=5; i++) {
        liaisonBalogh >> tampon[i];
        check ^= tampon[i];
    }
    // réception checksum
    liaisonBalogh >> tampon[6];
    return (tampon[6] == check);
}
```

E.3. Programmation multithread

Question E.3.1. Compléter en C++ le constructeur de la classe `LecteurEtiquettes` de façon à lancer la fonction membre `RecevoirBalogh()` dans le thread pointé par `pThreadRecevoirBalogh` , avec une priorité par défaut.

```
LecteurEtiquettes::LecteurEtiquettes(char *cheminLiaisonBalogh){
      liaisonBalogh.open(cheminLiaisonBalogh, ios::in) ;
      pThreadRecevoirBalogh = new Thread(RecevoirBalogh);
}
```

E.3.2. Entourer la **section critique** dans le code suivant :

```
etatLecture_t LecteurEtiquettes::FournirNumeroEtiquette(
                    numeroEtiquette_t * pNumEtiquette,
                    time_t*             pDateLecture  )
{
    etatLecture_t ret;

    ret = _etatDerniereLecture;
    if (ret ==  LECTOK ) {
        *pNumEtiquette = _numeroEtiquette;
    }
    if (pDateLecture != NULL) {
        *pDateLecture = _dateLecture;
    }
    return ret;
}
```

E3.3. Avec NTR++, quel type d'objet peut-on utiliser pour implémenter une exclusion mutuelle ?

Un sémaphore, initialisé à 1.

E3.4. Indiquer la donnée membre privée qu'il faut ajouter à la classe `LecteurEtiquettes` pour implémenter cet objet.
Ecrire les lignes de code à ajouter avant et après la section critique.

En donnée membre privée : `Semaphore` **`sem;`**
Avant la section critique : `sem.wait();`
Après la section critique : `sem.post();`

F. INFORMATION STATION DES VOYAGEURS

F.1. Etude de l'interconnexion des BIV

F1.1 Donner le nombre et le type d'interfaces séries présentes sur les BIV.

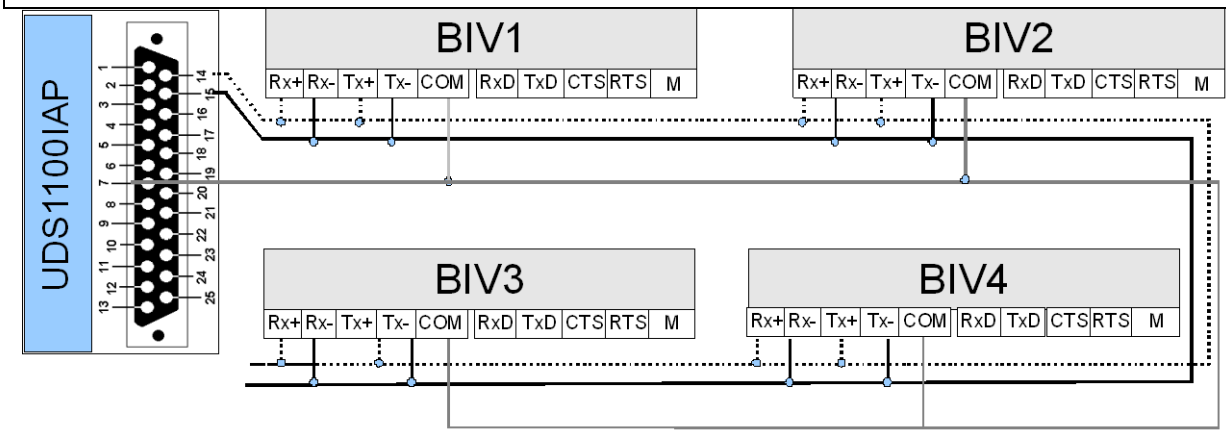
Les BIV possèdent deux interfaces série :

- *Une RS232*
- *Une RS485*

F1.2 Justifier l'utilisation de la RS485 pour le raccordement des BIV.

La RS232 n'est pas une liaison multipoint, afin de connecter plusieurs BIV sur la passerelle il est nécessaire d'utiliser la RS485 qui elle est une liaison multipoint.
De plus la RS485 utilise un support de transmission en mode différentiel ce qui permet de limiter l'influence des perturbations extérieures.

F1.3 Compléter sur le schéma l'interconnexion des BIV et de cette passerelle.



F.2. Analyse des trames de commande des BIV

F2.1 Quel est le principe de l'arbitrage maitre / esclave utilisé par Modbus?

Le principe d'arbitrage est un maitre qui gère la communication avec des esclaves.
Le maître interroge les esclaves et ces derniers répondent au maître.
Les esclaves ne peuvent pas communiquer directement entre eux, seul le maître peut initier une communication?

F2.2 . La passerelle ModbusTCP / ModbusRTU est-elle maitre ou bien esclave sur le réseau RS485 des BIV (justifiez votre réponse) ?

La passerelle est le maitre Modbus, les SX502 sont des esclaves Modbus RTU(Annexe 8).

F2.3 Indiquer sur quel réseau cette trame a été capturée. Justifier votre réponse.

C'est une trame Modbus TCP, elle à donc été capturé sur le réseau RMS. Dans les champs de cette trame on peut remarquer les champs des couches Ethernet et TCP/IP qui ne sont pas présent dans une trame Modbus « pure ».

Question F2.4

- Quel est le code fonction Modbus utilisé pour piloter l'afficheur (Annexe7) ?
- Donner sa signification.
- Quels sont les noms des champs de la requête Modbus associés à ce code fonction (Annexe 8)?

Code fonction Modbus n° 16 (Ox10) : Write Multiple Register

Paramètres de Write Multiple register :

- Adresse du premier mot : 2 octets
- Nombre de mot à écrire : 2 octets
- Nombre d'octet de donnée : 1 octet
- Donnée à écrire

Question F2.5 : Donner la commande à envoyer à l'afficheur permettant l'affichage de la 3ième ligne de l'exemple fig10 (Annexe 7).

24 4c 30 33 : $L03 écriture sur la troisième ligne

2d 50 69 63 20 20 20 20 20 3a 20 34 20 6d 69 6e : -PIC : 4 min

Question F2.6 Analyse de l'entête MBAP Header la trame Modbus TCP du tableau 2 :

- Trouver et donner dans l'entête ModbusTCP (MBAP Header, annexe 8) le nom du champ qui contient l'adresse Modbus de l'afficheur.
- Donner l'adresse Modbus de l'afficheur qui va recevoir cette trame.

Dans l'entête MBAP Header de ModbusTCP, le champ "Unit Identifier" permet de passer l'adresse Modbus du périphérique Modbus esclave qui recevra cette trame par l'intermédiaire de la passerelle ModbusTCP /modbus

Dans la trame l'adresse Modbus de l'afficheur qui va recevoir la trame est : 5

F2.7. Interpréter les 6 premiers octets de la requête Modbus contenus dans la trame ModbusTCP du tableau 2 : Capture d'une trame ModbusTCP.

10 : code fonction : Write Multiple Register

00 00 : Adresse du premier registre à écrire

00 1a : 26 mots à écrire

34 : 52 : octets à écrire

F2.8. Extraire de la requête Modbus les commandes de manipulation de texte envoyées à l'afficheur (annexe 7).

24 46 31 : $F1 (clignotement des caractères)

24 4d 31 : $M1(choix de la police) : Acala7

24 4c 30 34 : $L04 : écriture à la ligne 4

…

24 46 30 : $F0 (arrêt clignotement)

F2.9. Donner le résultat de cette commande en terme d'affichage .

I	n	f	o	r	m	a	t	i	o	n		v	o	y	a	g	e	u	r		:		l	i	g	n	e		C		e	n		p	a	n	n	e	

Le message suivant est écrit sur la ligne 4 : Information voyageur : ligne C en panne

G. RÉSEAU MULTI SERVICE

G.1. Ethernet industriel

G.1.1. Avec la méthode CSMA/CD, peut-on garantir le temps d'accès au support et le délai d'émission complète d'une trame ? Justifier la réponse.

Non car :

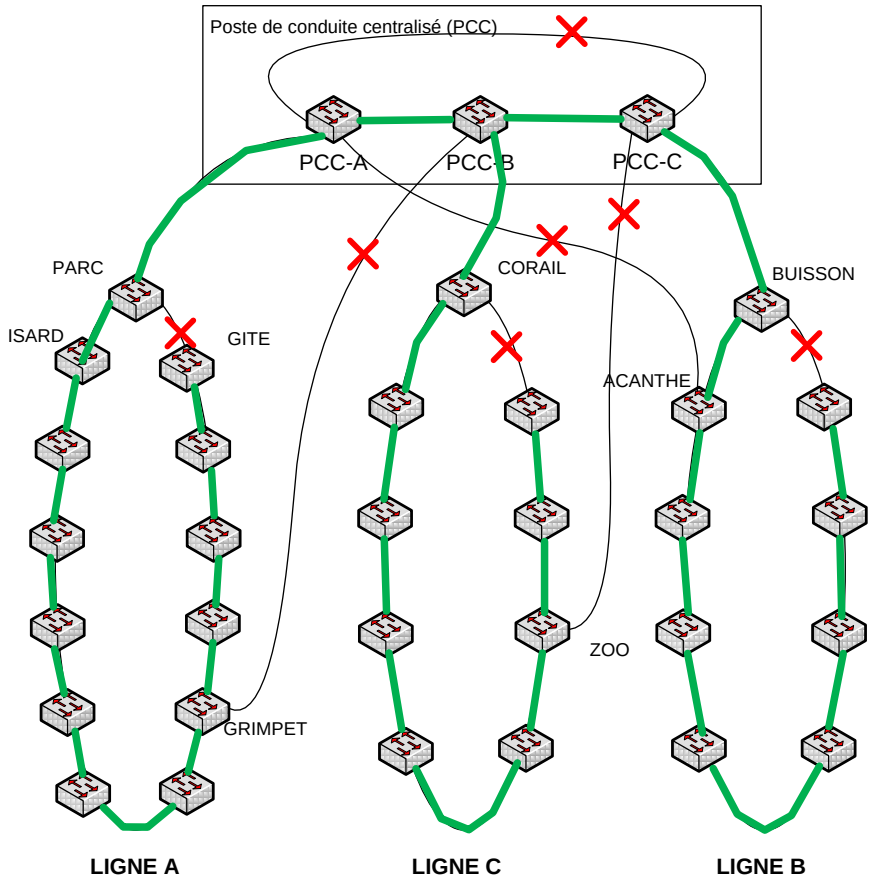
- le support peut être occupé par n'importe quel autre équipement (accès multiple)
- il peut se produire des **collisions**
- la résolution des collisions est probabiliste

G.1.2. Quelle est la topologie physique :
a) d'un réseau Ethernet sur câble coaxial ?
b) d'un réseau Ethernet sur paires torsadées et fibres optiques, comportant plusieurs commutateurs ?

a) Bus
b) Arbre (ou étoile étendue)

G.1.3. Dans la figure ci-dessous :
- Marquer d'une croix (X) les liaisons redondantes à désactiver en fonctionnement normal (la topologie résultante doit être compatible avec un réseau Ethernet commuté normal).
- Indiquer le nombre total de liaisons désactivées

Nb de liaisons désactivées : 7



Il faut recréer une topologie en arbre (figurée ici en trait vert épais).
La solution proposée consiste à couper chacune des 7 boucles identifiées en 1 point, soit 7 coupures au total.

G.2. Etude de la volumétrie

G.2.1. En se reportant à **« Annexe 9** : Réseau Multi Service (RMS) », Compléter le tableau ci-après.

CODEC	**Taille trame complète** avec préambule et IFG **(en bits)**	**Nb de trames par seconde**	**Bande passante ethernet (en kilobits/s)**
G.711 (PCM)	240o+78o = 318o = 2544 bits	33	83,9kbit/s
G.729a (CS-CELP)	(2x10)o+78o = 98o = 784 bits	50	39,2kbit/s

G.2.2. Combien de caméras cette liaison permet-elle de traiter au maximum (détailler le calcul) ?

Etant en full duplex, le flux entrant des caméras n'a pas d'incidence.
Nombre de caméras = (100Mbit/s x 0,75) / 250kbit/s = 300

G.2.3. Justifier le choix du multicast par rapport à l'unicast pour l'application « tv tram ».

En multicast, un même flux de données est partagé par l'ensemble des clients. Le besoin en bande passante est constant quel que soit le nb de clients.

En unicast, il faudrait un flux dirigé vers chacun des clients, ce qui impliquerait un besoin en bande passante proportionnel au nombre de clients.

G.3. Câblage – choix du support

G.3.2. Choisir les types de câbles et les normes Ethernet adaptés à chaque liaison, et au meilleur coût.
Compléter le tableau ci-dessous (consulter « **Annexe 9** : Réseau Multi Service (RMS) »).

Liaison	**Longueur**	**Besoin en bande passante**	**Câble**	**Norme Ethernet**	**Débit nominal**
BUISSON - ACANTHE	950m	10Mbit/s	Multimode Optical Fiber	100BASE-FX	100Mbit
PARC - PCC-A	1500m	30Mbit/s	Multimode Optical Fiber	100BASE-FX	100Mbit
GRIMPET - PCC-C	8km	30Mbit/s	Singlemode Optical Fiber	100BASE-FX	100Mbit
PCC-A – PCC-B	10m	60Mbit/s	Twisted Pair (cat 5)	100BASE-TX Ou 1000BASE-T(*)	100Mbit/s Ou 1Gbit/s
Switch station – caméra IP	50m	300kbit/s	Twisted Pair (cat 5)	100BASE-TX	100Mbit/s

(*) le coût restant faible, ce choix peut être justifié pour garder une marge d'évolutivité du réseau.

G.4. Plan d'adressage IP

G.4.1. Avec le masque 255.255.0.0, combien peut-on créer de sous-réseaux dans le bloc d'adresse 10.0.0.0/255.0.0.0 (justifier la réponse) ?
Combien y a-t-il d'adresses IP utilisables dans chaque sous-réseau ?

Bloc d'adresses à découper : 255.0.0.0 => 8 bits de netmask
Le découpage avec le masque 255.255.0.0 utilise 8 bits de sous-réseau, soit 2^8 = 256 sous-réseaux.

Dans chaque sous-réseau il y a 16 bits de hostid. Il y a donc $2^{16} – 2 = 65534$ adresses utilisables

G.4.2. Compléter le tableau ci après de façon à respecter le plan d'adressage préconisé.

Sous-réseau	**Adresse sous-réseau**	**Adresse de diffusion sur le sous-réseau**	**Adresse d'hôte la plus basse**	**Adresse d'hôte la plus haute**
Données ligne A	10.0.0.0	10.0.255.255	10.0.0.1	10.0.255.254
Vidéosurveillance ligne A	10.1.0.0	10.1.255.255	10.1.0.1	10.1.255.254
Signalisation ferroviaire ligne A	10.2.0.0	10.2.255.255	10.2.0.1	10.2.255.254
Billettique ligne A	10.3.0.0	10.3.255.255	10.3.0.1	10.3.255.254