



**LE RÉSEAU DE CRÉATION  
ET D'ACCOMPAGNEMENT PÉDAGOGIQUES**

**Ce document a été mis en ligne par le Canopé de l'académie de Bordeaux  
pour la Base Nationale des Sujets d'Examens de l'enseignement professionnel.**

Ce fichier numérique ne peut être reproduit, représenté, adapté ou traduit sans autorisation.

# CORRIGE

**Ces éléments de correction n'ont qu'une valeur indicative. Ils ne peuvent en aucun cas engager la responsabilité des autorités académiques, chaque jury est souverain.**

**BREVET DE TECHNICIEN SUPÉRIEUR  
INFORMATIQUE ET RÉSEAUX  
POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES**

**ÉTUDE D'UN SYSTÈME INFORMATISÉ**

Session 2011

Durée : 6 heures  
Coefficient 5

**DOCUMENT RÉPONSES  
AVEC CORRIGÉ**

<i>Repère</i>	<i>Partie</i>	<i>Page</i>	<i>Points</i>
Partie A	Analyse	2/12	15 points
Partie B	Conception	4/12	25 points
Partie C	Communication	8/12	15 points
Partie D	Base de données	9/12	8 points
Partie E	Réseaux	10/12	37 points

## Partie A – Analyse

### Q-1 Le transpondeur

Justifier le choix du boîtier en verre de 32 mm par rapport au boîtier en verre de 23 mm.

La détection du transpondeur doit se faire jusqu'à une hauteur de 70 centimètres par rapport à l'antenne. Le boîtier 32 mm peut être détecté jusqu'à 1 mètre, alors que le boîtier 23 mm ne peut être détecté que jusqu'à 60 cm.

### Q-2 Lecture d'un transpondeur

Justifier par le calcul que la lecture d'un transpondeur est possible sur un tapis d'un mètre de longueur.

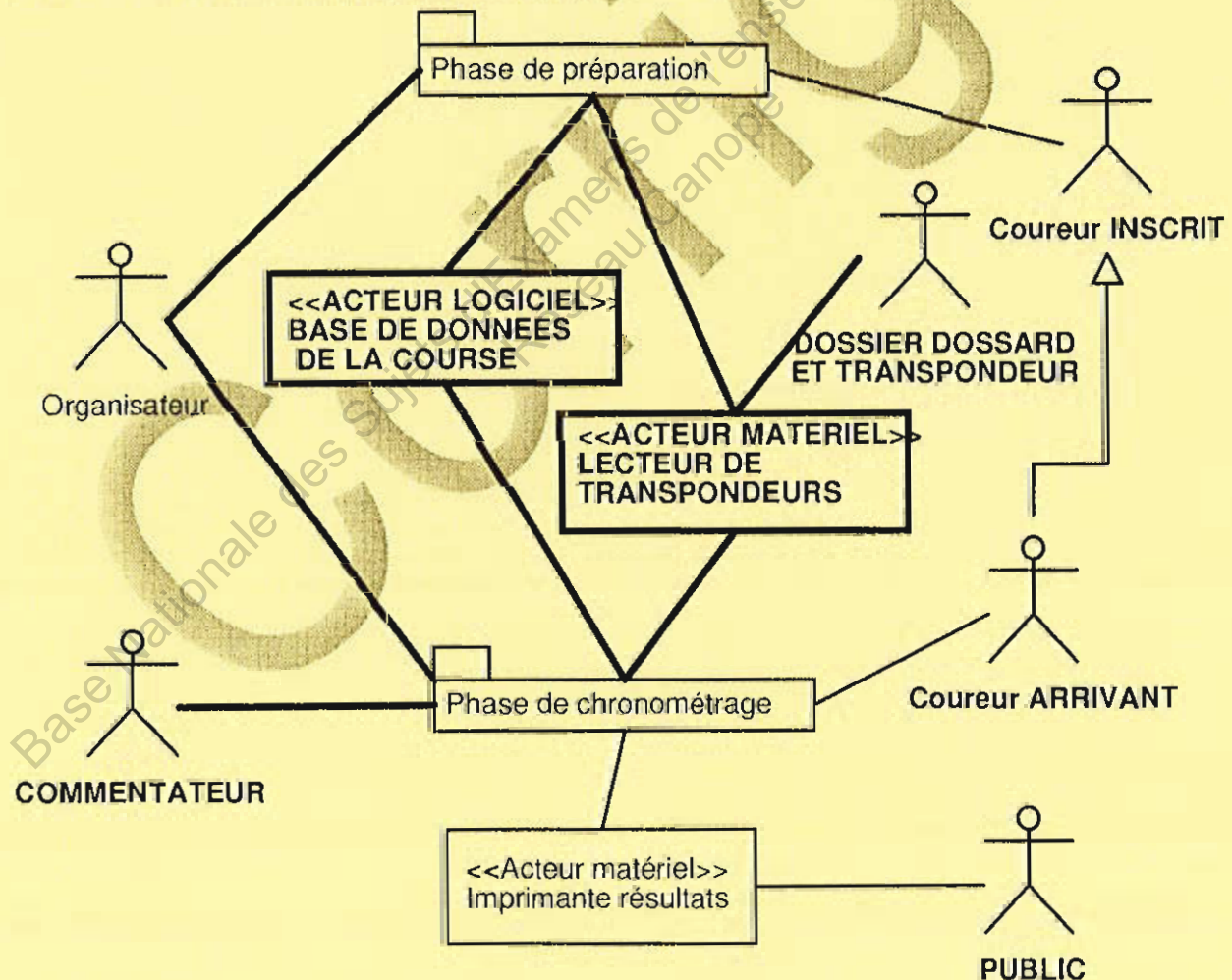
La vitesse de 10 km/h correspond à un déplacement d'1m/360 ms.

Sachant que la lecture d'un transpondeur se fait en 70 ms, la lecture est possible.

### Q-3 Contexte global des cas d'utilisation

Complétez le diagramme : complétez les noms des acteurs déjà donnés, ajoutez si nécessaire des acteurs humains ou matériels en précisant leurs noms et leurs liens avec les paquetages.

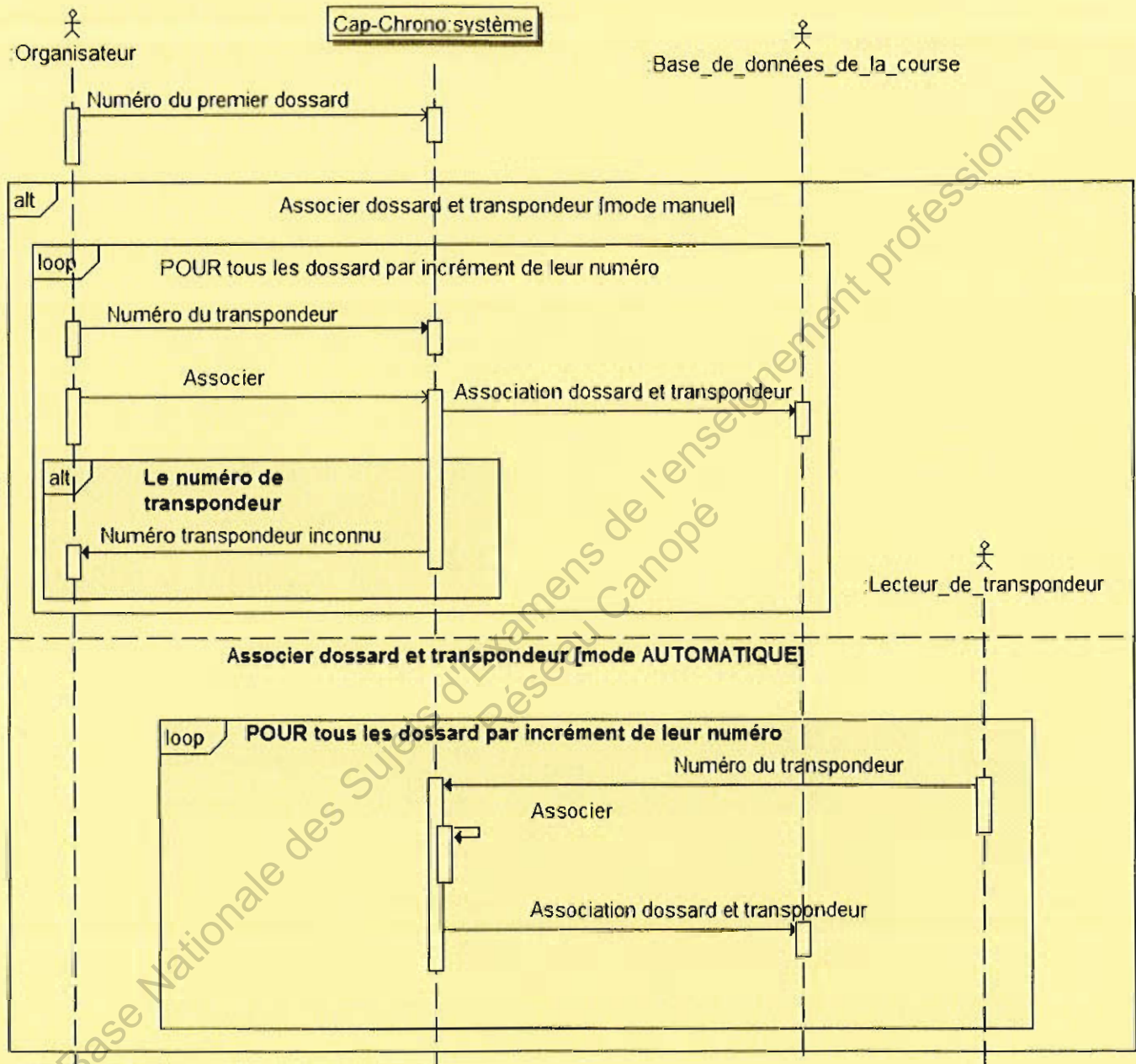
Réponses attendues : EN MAJUSCULES et en gras



### Q-4 Scénario du cas d'utilisation « Associer dossard et transpondeur »

Compléter le diagramme de séquence correspondant à ce cas d'utilisation, actuellement partiellement établi dans le document réponse. Compléter la description du mode « manuel » et décrivez le mode « automatique ».

Réponses attendues : SUR FOND GRIS et partiellement en gras



## Partie B – Conception

### Q-5 Codage en C++ de la déclaration de la classe RSCom

```
class RSCom{
private:
    HANDLE hdl;
    string NomCom;

public:
    RSCom(string Nom);
    ~RSCom();

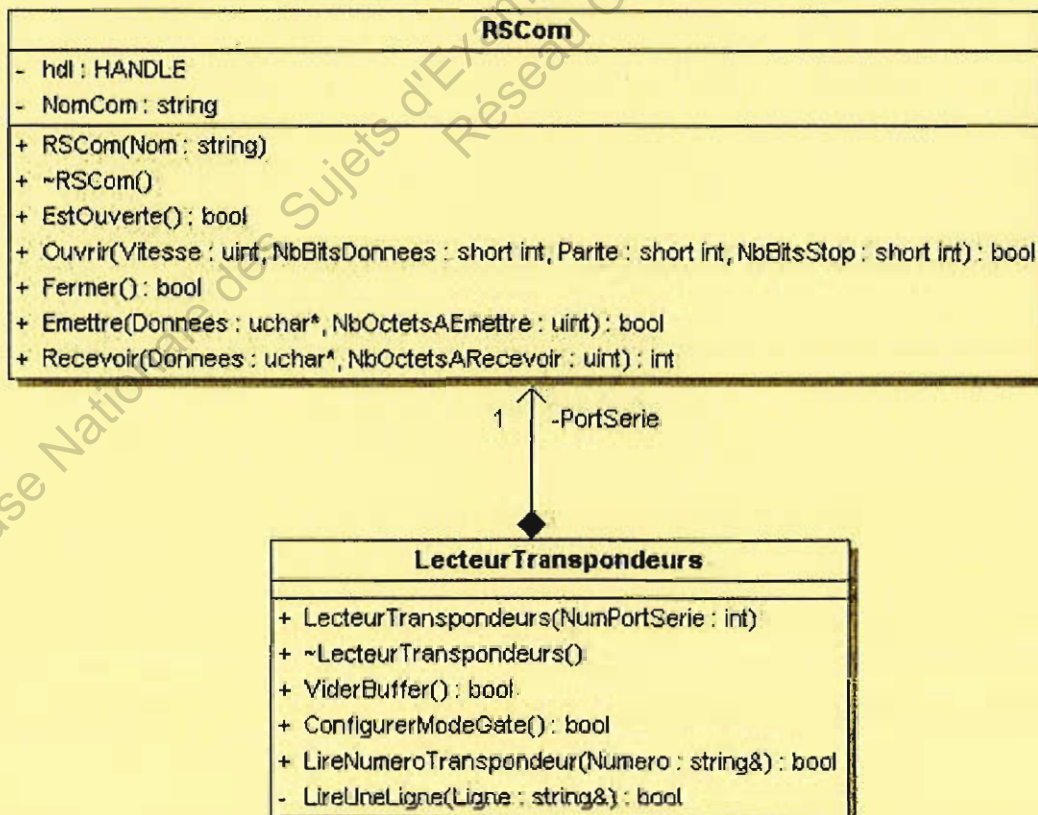
    bool EstOuverte();

    bool Ouvrir(uint Vitesse, short int NbBitsDonnees,
               short int Parite, short int NbBitsStop);
    bool Fermer();

    bool Emettre(uchar* Donnees, uint NbOctetsAEmettre);
    int Recevoir(uchar* Donnees, uint NbOctetsARecevoir);
};
```

### Q-6 Classe LecteurTranspondeurs

Dessiner la relation entre les classes LecteurTranspondeurs et RSCom. Faire apparaître le nom de rôle, la visibilité, la navigabilité et la cardinalité de la relation.



**Q-7 Déclaration en C++ de la classe Chronometre**

```
class Chronometre : public Thread {
private:
    LecteurTranspondeurs* LesLecteurs[4];

    bool InitialiserLecteurs();

public:
    Chronometre(int NumPremierPort);
    ~Chronometre();

protected:
    void Run();
};
```

**Q-8 Codage en C++ du constructeur de la classe Chronometre**

```
Chronometre::Chronometre(int NumPremierPort) : Thread() {
// Création dynamique des 4 lecteurs de transpondeurs
for (int lecteur = 0; lecteur < 4; lecteur++) {
    LesLecteurs[lecteur] = new
LecteurTranspondeurs(NumPremierPort+lecteur);
}
}
```

**Q-9 Codage en C++ du destructeur de la classe Chronometre**

```
Chronometre::~Chronometre() {
// Destruction des 4 lecteurs de transpondeurs
for (int lecteur = 0; lecteur < 4; lecteur++) {
    delete LesLecteurs[lecteur];
}
}
```

**Q-10 Codage en C++ de la méthode *InitialiserLecteurs()* de la classe Chronometre**

```
bool Chronometre::InitialiserLecteurs() {  
    // Pour chacun des 4 lecteurs  
    for (int lecteur = 0; lecteur < 4; lecteur++){  
        if (!LesLecteurs[lecteur]->ViderBuffer())  
            return false ;  
  
        if (!LesLecteurs[lecteur]->ConfigurerModeGate())  
            return false ;  
    }  
  
    return true ;  
}
```

**Q-11 Accessibilité de l'attribut *ConnexionEtablie* à partir de la classe JournalLumineux**

Réponse : NON

Explication :

L'attribut *ConnexionEtablie* a une visibilité privée dans la classe *TCPsocketClient* ; il n'est accessible qu'à partir de la classe *TCPsocketClient*.

**Q-12 Méthode à ajouter à la classe *TCPsocketClient***

Prototype :

```
bool GetConnexionEtablie() const;
```

Justification de sa visibilité :

Cette méthode doit avoir une visibilité publique (*public*) pour être accessible à partir de la classe *JournalLumineux*.

Codage en C++ de cette méthode :

```
bool TCPsocketClient::GetConnexionEtablie() const {  
    return ConnexionEtablie;  
}
```



**Q-13 Codage en C++ de la méthode *Connecter()* de la classe TCP SocketClient**

```
bool TCP SocketClient::Connecter() {
// Création du socket
IdSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
if (IdSocket == INVALID_SOCKET) return false;

// Connexion au journal lumineux
struct sockaddr_in infosServeur;

infosServeur.sin_family = AF_INET;
infosServeur.sin_addr.s_addr = inet_addr(adresseServeur.c_str());
infosServeur.sin_port = htons(PortServeur);

if (connect(IdSocket, (const struct sockaddr*) &infosServeur,
sizeof(infosServeur)) == SOCKET_ERROR) // ou != 0
return false;

ConnexionEtablie = true ;
return true ;
}
```

**Q-14 Codage de la méthode *Effacer()* de la classe JournalLumineux**

```
bool JournalLumineux::Effacer() {
// Connexion
if (!LeSocketClient->Connecter()) return false;

// Envoi de la commande d'effacement
if (!LeSocketClient->Envoyer("$E\r", 3)) return false;

// Déconnexion
if (!LeSocketClient->Deconnecter()) return false;

return true;
}
```

## Partie C – Communication

### Q-15 Octet de « start »

Valeur en hexadécimal : 7E.

Rôle : Il sert à synchroniser la lecture des données du transpondeur.

### Q-16 BCC

Les 16 bits du BCC contrôlent la validité des 64 bits d'identification du transpondeur.

### Q-17 Configuration de la liaison série

Vitesse : 9600 baud.

Format de la transmission : 8 bits de donnée, pas de parité et 1 bit de stop.

### Q-18 Transmission d'un message

Caractères	Signification
GR	Mode GATE et transpondeur de type read only
022	Nombre d'enregistrement
4095	Code d'application
4000000627370400	Code d'identification du transpondeur
<CR><LF>	Fin du message

### Q-19 Charge du transpondeur

150dec = 96hex

Messages :

```
PC --> Lecteur
Z
Lecteur --> PC
Z
PC --> Lecteur
96
Lecteur --> PC
<CR><LF>
```

### Q-20 Choix du concentrateur

Référence : QSU-100

Raisons : 4 Ports RS232

## Partie D – Base de données

Ne pas pénaliser si les apostrophes ou guillemets sont manquants dans les requêtes.

**Q-21** Requête d'obtention du numéro et de l'identifiant de tous les transpondeurs déclarés perdus

```
SELECT Numero, Identifiant FROM Transpondeur WHERE Perdu = TRUE
```

**Q-22** Requête de mise à jour du champ « HeureDepart » avec la valeur « 14:25:32 » pour la course ayant un champ « idCourse » de valeur « 23 »

```
UPDATE Course SET HeureDepart = "14:25:32" WHERE idCourse = "23"
```

**Q-23** Requête d'obtention des informations nécessaires au calcul du temps et à l'ajout dans la table « Arrivee » de l'enregistrement correspondant à l'arrivée du coureur ayant le transpondeur d'identifiant « 0000000139319757 »

```
SELECT idInscrit, HeureDepart FROM Inscrit, Course, Transpondeur  
WHERE idCourse = Course_idCourse  
AND idTranspondeur = Transpondeur_idTranspondeur  
AND Identifiant = "0000000139319757"
```

La jointure entre les tables « Course », « Inscrit » et « Transpondeur » peut aussi être réalisée à l'aide de l'instruction JOIN.

## Partie E – Réseaux

**Q-24 Choix du commutateur PoE et justification si le matériel ne convient pas**  
Il faut au moins 5 ports 100M et 2 ports 100M avec PoE.

Commutateur	Convient (cocher)	Ne convient pas ( <u>justifier</u> )
N°1		Pas assez de ports 100M
N°2		Pas de PoE
N°3		Pas assez de ports 100M
N°4	X	

**Q-25 Taille des classes d'adresses**

Nombre d'adresses IP utilisables :

Classe d'adresses	Nombre d'adresses IP utilisables dans une classe
Classe A	$2^{24} - 2 = 16777214$
Classe B	$2^{16} - 2 = 65534$
Classe C	$2^8 - 2 = 254$

Justification du choix d'une adresse réseau de classe C :

10 adresses IP sont nécessaires donc une classe C suffit amplement.

**Q-26 Possibilité de restriction de la plage d'adresses IP**

Masque : 255.255.255.240 ou /28

Justification :

10 adresses IP sont nécessaires donc on laisse seulement 4 bits pour la partie « machine » de l'adresse IP car  $2^4 - 2 = 14$  adresses IP disponibles.

**Q-27 Caractéristiques du réseau privé choisi**

Adresses IP utilisables	
Première adresse	Dernière adresse
192.168.1.1	192.168.1.254

Réseau	
Masque	Adresse de diffusion
255.255.255.0 ou /24	192.168.1.255

**Q-28 Adresses IP statiques ou dynamiques**

Pour chaque matériel, cocher la case correspondante.

Matériel	Adresse IP statique	Adresse IP dynamique
PC « Arrivées »		X
Webcam	X	
Journal lumineux	X	
PC « Commentateur »		X
Point d'accès Wifi	X	
PC « Passerelle »	X	
PC « Serveur Base de données »	X	
PC « Résultats »		X

**Q-29 Plan d'adressage des matériels recevant une adresse IP privée statique**

Matériel	Adresse IP privée statique
Point d'accès Wifi	192.168.1.1
PC « Serveur Base de données »	192.168.1.2
PC « Passerelle »	192.168.1.3
Journal lumineux	192.168.1.4
Webcam	192.168.1.5

Toutes les adresses entre 192.168.1.1 et 192.168.1.254 sont bien sûr valides.

**Q-30 Caractéristiques des adresses IP publiques**

De 200 à 203, 2 bits du 2<sup>ème</sup> octet sont utilisés pour la partie « machine » de l'adresse IP. Donc il reste  $8 + 6 = 14$  bits pour la partie « réseau » des adresses IP.

Nombre de bits de la partie « réseau » des adresses IP	Masque	Nombre d'adresses IP utilisables
14	255.252.0.0	$2^{18} - 2 = 262142$

**Q-31 Interface réseau « MS TCP Loopback interface »**

C'est l'interface de loopback ou de bouclage local. Elle permet de faire des tests locaux de la pile TCP/IP.

**Q-32 Nombre d'interfaces réseau physiques**

Il y a 2 interfaces physiques (lignes 4 et 5). Elles sont identifiables par la présence d'une adresse physique (ou adresse MAC) et par leur description.

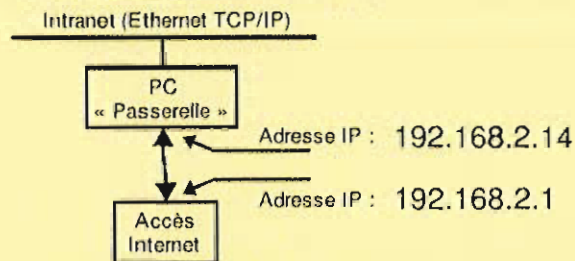
**Q-33 Ligne 9**

Destination = 0.0.0.0 et masque = 0.0.0.0 : c'est la route par défaut. C'est la route qu'emprunteront les trames qui n'ont pas d'entrée correspondante dans la table de routage.

### Q-34 Adresse IP des interfaces réseau pointées par les flèches

La ligne 11 signale que l'adresse 192.168.2.14 correspond à la machine elle-même (127.0.0.1).

Les lignes 13 et 9 indiquent que la passerelle par défaut (192.168.2.1) est accessible par l'interface 192.168.2.14. La passerelle par défaut correspond donc à l'interface de la « box » d'accès à Internet.



### Q-35 Information recherchée via les trames 2 et 3

Recherche de l'adresse MAC de l'interface d'adresse IP 192.168.2.1.

### Q-36 Information recherchée via les trames 4 et 5

Recherche de ou des adresses IP correspondant à l'URL [www.google.com](http://www.google.com).

### Q-37 Rôle des trames 6, 7 et 8

Protocole TCP = mode connecté. C'est la connexion du PC « Passerelle » (192.168.2.14) à un des serveurs web (port 80) de Google (209.85.227.104). SYN = demande de connexion et ACK = acquittement.

### Q-38 Rôle de la trame 9

C'est la requête HTTP demandant la page par défaut du serveur web de Google.

### Q-39 Nom de la couche du modèle TCP/IP pour ces protocoles

<b>Protocole</b>	<b>ARP</b>	<b>DHCP</b>	<b>ICMP</b>	<b>IP</b>
<b>Nom de la couche</b>	Accès réseau	Application	Internet	Internet
<b>Protocole</b>	<b>HTTP</b>	<b>TCP</b>	<b>Telnet</b>	<b>UDP</b>
<b>Nom de la couche</b>	Application	Transport	Application	Transport

### Q-40 Protocole de transport correspondant au mode connecté

Mode connecté = TCP.

### Q-41 Numéro du port d'écoute

*Justification* : les valeurs inférieures à 1024 sont réservées aux services « bien connus » et définis.