

SYSTEME et RESEAUX

AVERTISSEMENT : Ceci est un sujet commun pour les matières : système et réseaux.

Vous devrez, cependant, rédiger sur deux copies différentes :

- une copie pour la partie réseau (1^{ère} Partie – Questions 1,2,3,4)
- une copie pour la partie système (2^{ème} Partie – Questions 1,2,3,4,5,6,7)

1 - Présentation du contexte

Le centre commercial "Les portes de Montpellier", souhaite mettre en place un système de mesures modernes pour mieux gérer les flux de clients.

Pour cela il va mettre en place une multitude d'objets connectés, qui seront des capteurs pour compter le nombre de passages. Ces objets seront répartis en 4 groupes (cf Annexe 1 - Plan du centre) :

- Le premier groupe , est un ensemble de capteurs, disposés aux 3 entrées des parkings , qui comptera les véhicules qui entrent et sortent.
- Le deuxième groupe, disposés aux 3 entrées du Centre, comptera les entrées et sorties de personnes dans les locaux du centre commercial.
- Le troisième groupe, disposé aux deux entrées du magasin comptera les personnes qui entrent.
- Le quatrième, disposés aux caisses, permettra de compter les passages.

Ces objets qui utilisent la technologie (IoT) sont de simples capteurs lumineux qui incrémentent un compteur dès qu'il y a un passage. Leur particularités, c'est qu'ils utilisent un réseau de transmission sans fil de faible débit (LoRa).

2 - Fonctionnement général

Le fonctionnement général de cette installation est décrite ci-dessous et schématisé en ANNEXE 2.

Possédant de faibles capacités de stockage , un objet transmet un message à une machine située à l'Accueil Client du centre commercial toutes les 5 minutes. La structure des données transmises est décrite dans l'annexe 3.

La machine de l'accueil est en charge de mettre en forme les données pour les transmettre à une base de données qui centralise toutes les mesures . En fait cette machine totalise par type de capteur le nombre total de mouvements enregistrés et envoie toutes les 10 minutes un message ayant la structure ci-dessous :

15:12:2107:10:10/Park:IN:20/Park:OUT:10/Center:IN:50/Center:OUT:30/Entree:IN:30/Caisse:OUT:30

Interprétation : Le 15/12/2017 à 10h10 entre 10h et 10h10 il a été observé :

20 voitures qui entrent et 10 qui sortent dans le parking

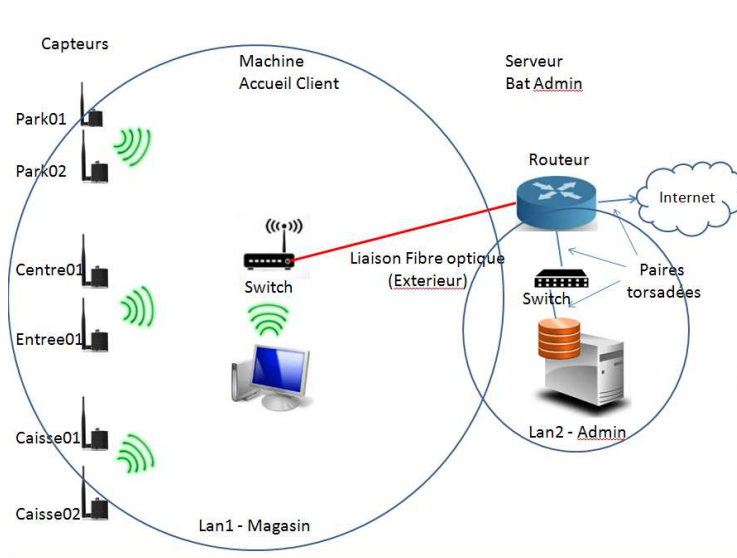
50 personnes qui entrent et 30 qui sortent du centre commercial,

...

Cette base se situe sur un serveur, dans le bâtiment administratif, situé à quelques dizaines de mètres du centre commercial.

1ère Partie – Réseaux

1 - Concevoir le schéma du réseau de cet établissement. Vous préciserez les équipements nécessaires et leur type, comment ils sont connectés, ainsi que les types de supports (câbles) à utiliser. (6 points)



L'idée est de créer 2 réseaux :

-L'un au niveau du magasin (Lan1), dans lequel les objets transmettent les données à la machine de l'accueil. Dans ce réseau on utilisera bien sûr des liaison sans fil et les machines communiquent via un switch compatible LoRa (2 points).

- L'autre au niveau de l'administration (Lan2), dans lequel on trouve le serveur, mais aussi d'autres machines qui peuvent servir à l'administration. Ici deux cas possibles : des liaisons sans fil ou des liaison par câbles à paires torsadées (solution à privilégier) quand on veut accéder à un serveur, à cause du volume des données (2 points).

Ces 2 réseaux communiquent via un serveur installé, au bâtiment administratif, pour des raison évidentes de sécurité. La liaison entre le serveur et le magasin se fera par une fibre optique, car le câble doit relier 2 bâtiments. (1 point)

Un liaison vers l'extérieur est à prévoir au niveau du serveur. (1 point)

Remarques:

- on peut aussi accepter une solution dans laquelle il y aura 2 routeurs : un au magasin et un à l'administration. Mais la solution n'étant optimale, car trop lourde, on enlèvera 1 a 2 points.

- Il faut bien faire apparaitre routeur et switch, une solution avec seulement des switchs n'est pas correcte.

2 - Dans ce type de réseau, les problèmes de sécurité s'articulent essentiellement autour de la disponibilité du système. En effet , les objets transmettant exclusivement des valeurs correspondant à un comptage, leur interception ne pose pas de problème. Par contre , si un ou plusieurs composants (machines serveur incluses) venaient à ne plus fonctionner, cela va impacter l'organisation de l'établissement. (4 points)

- Quelles peuvent être les causes de blocage de ce système ?
- Quelles mesures de protection peut-on envisager ?

Causes de blocage (2 points : Au moins 4 items : 4x0,5)	Mesures de protection (2 points : Au moins 4 items : 4x0,5)
1 - Physique	
- Panne des capteurs	- Avoir du matériel de rechange en stock
- Panne matériel informatique et/ou transmission	- Idem

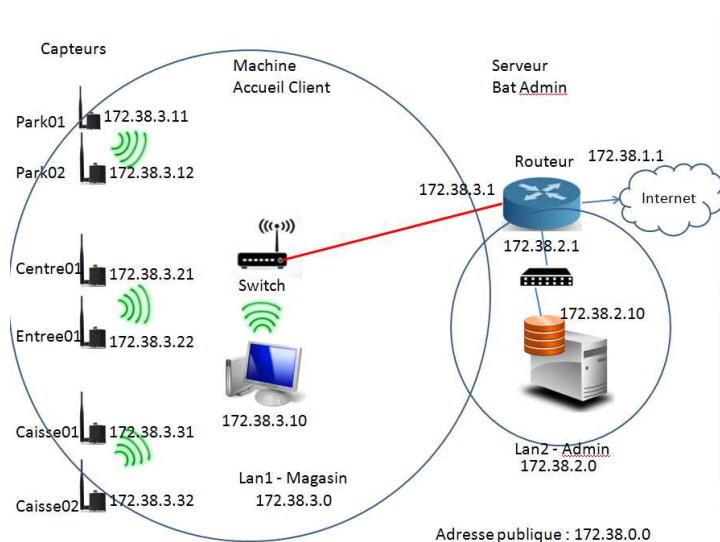
(switch) - Coupure alimentation électrique - Dégradation ou vols	- Mise en place d'alimentation de secours (batteries ...) - Protéger les capteurs dans des boîtiers, mettre sous alarme les ordinateurs.
2 - Système /réseau	
- Dysfonctionnement pour cause de virus - Blocage du système à cause de DoS	- Mettre en place des antivirus et de pare-feu - Sensibiliser le personnel pour limiter les risques d'intrusions

Toute réponse cohérente doit être prise en compte.

3 - Sachant que l'établissement possède l'adresse réseau : 172.38.0.0, proposez un plan d'adressage de ce réseau, en prenant en compte les objectifs de sécurité de la question précédente (4 points)

Deux possibilités :

1 - On garde l'adresse publique 172.38.0.0 pour numéroté les machines et composants



Comme nous avons une classe B, on va utiliser le 3ème octet pour identifier le réseau.

Par exemple:

172.38.1.0 pour la liaison avec l'extérieur

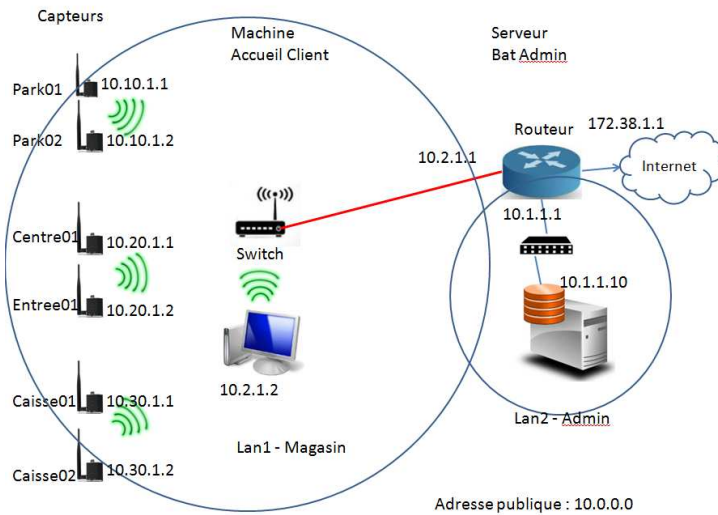
172.38.2.0 pour le réseau de l'admin

173.38.3.0 pour le magasin et les capteurs

Attention le masque à utiliser pour les tables de routage sera 255.255.255.0

Avec cette solution , les adresses publiques peuvent poser un problème, car les capteurs seront accessibles de l'extérieur, si un pare-feu n'est pas installé et bien paramétré sur le routeur. Un DMZ serait la bienvenue.

2 - On utilise des adresses privées de type 10.0.0.0 pour numéroté les machines et composants et on garde l'adresse publique pour la communication vers l'extérieur (exemple 172.38.1.1)



L'utilisation d'une adresse privée présente au moins deux avantages:

- plus grande facilité de numérotation, on peut prévoir des plages de numérotation par groupe de capteurs.

Par exemple:

10.10.0.0 pour les capteurs des parkings

10.20.0.0 pour les capteurs de l'entrée du centre

10.30.0.0 pour les capteurs des caisses

etc ...

- meilleure protection des équipements du centre, car les adresses privées ne sont pas accessibles de l'extérieur. Cela, limite voire supprime les risques d'attaques par DoS.

Remarques :

- 2 points si la réponse est une solution cohérente (une des 2 solutions)

- 1 point si les choix sont justifiés

- 1 point si des références aux problèmes de sécurité sont évoqués

- +1 point bonus, si l'étudiant évoque les 2 possibilités d'adressage

4 - Les objets communiquant via le protocole TCP/IP, ils envoient les données à une application (serveur) située sur machine de l'accueil. Cette application serveur est accessible à l'adresse 10.20.1.10 et le port 1111. La logique d'échange est simple, le capteur se connecte au serveur, il envoie sa mesure, le serveur renvoie un accusé de réception : la chaîne "ACK", si la valeur est bien reçue et se déconnecte. (6 points)

S'il y a eu un problème le serveur retourne "NACK". Dans ce cas le capteur retransmet la mesure. En cas de nouvelle erreur il réessaye 2 fois, puis se déconnecte.

En utilisant les signatures des fonctions de l'annexe 4, proposer le code (ou pseudo-code) de l'application cliente, située sur les capteurs. Des explications dans ce code seront très appréciées du correcteur.

Remarque : Pour simplifier l'application, on met à votre disposition une fonction : `set_addr(Addr_IP4, Port)` qui va mettre en forme les différents champs de `sock-addr`.

// ces définitions ne sont pas attendues

```
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h> // chaînes de caractères
#include <sys/socket.h> // interface socket
#include <netinet/in.h> // gestion adresses ip
#include <sys/types.h>
```

```

#define SERV "10.20.1.10"    // adresse IP serveur
#define PORT 1111           // port d'ecoute serveur
int port,sock;              // n°port et socket
char mesure[80];            // chaine de caracteres représentant la mesure
char reponse[80];           // chaine de caracteres représentant la reponse

struct sockaddr_in  serv_addr;    // zone adresse

void set_addr(char * Addr_IP4, int Port)
// Cette partie n'est pas demandée aux étudiants
{ struct hostent *server; // nom serveur
server = gethostbyname(Addr_IP4); // verification existence adresse
if (!server){ fprintf(stderr, "Problème serveur \"%s\"\n", SERV);exit(1);}
// creation socket locale
bzero(&serv_addr, sizeof(serv_addr)); // preparation champs entete
serv_addr.sin_family = AF_INET; // Type d'adresses
bcopy(server->h_addr, &serv_addr.sin_addr.s_addr,server->h_length);
serv_addr.sin_port = htons(port); // port de connexion du serveur
}

void main()
{ // creation socket (1 point si fonction avec bons paramètres)
sock = socket(AF_INET, SOCK_STREAM, 0); // AF_INET=famille adresse internet
// SOCK_STREAM= mode connecte-TCP      creer_socket();
set_addr(SERV,PORT); // 0,5 points si pas oublié l'appel de cette fonction
// connexion au serveur 1 point si fonction correcte
if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
{ perror("Connexion impossible:");exit(1);}
printf ("connexion avec serveur ok\n");

// Envoie de la mesure 1 point pour l'envoi et 1 point pour la gestion de la réponse retour
// donc 1 point pour ces 2 lignes
write(sock,mesure,80); // ou  send(sock,mesure,80,0); --> envoi mesure
read (sock,reponse,80); // ou  recv(sock,reponse,80,0) --> attente réponse
int nb=1;
// et 1 point pour cette partie
while ((strcmp(reponse,"ACK",3)!=0) && nb <3)
{ write(sock,mesure,80); // ou  send(sock,mesure,80,0);
read (sock,reponse,80); // ou  recv(sock,reponse,80,0);
nb++;
}

close (sock); // 0,5 points si pas oublié
}

// on se garde 1 point pour la structure générale

```