

TP 16

Réseaux de Télécommunications

Mathex - Licence CC BY NC SA 4.0

<https://www.mathexien.com>

Table des matières:

Introduction:	2
Objectifs:	2
Instructions:	2
1. Présentation des réseaux	3
2. Simulation de réseaux	6
3. Topologie de réseaux	9
4. Protocoles et Couches Réseaux	11
5. Protocole du bit alterné	13
6. Et sur Python?	15

Introduction:

Nous avons déjà vu que toute information numérique est binaire et que des 0 et des 1 sont traités par les éléments de l'architecture d'une machine.

Nous allons ici étudier comment ces informations sont communiquées entre les machines à travers des réseaux de télécommunications.

Objectifs:

L'objectif de ce TP est d'étudier :

- la transmission des données dans un réseau
- les protocoles de communication
- les architectures réseaux

Instructions:

On va y aller progressivement, à travers plusieurs devoirs où je rajouterai à chaque fois la suite dans le même document.

Vous avez chacun un exemplaire de ce document que vous devez compléter par vos réponses aux questions.

1. Présentation des réseaux

Visualisez les capsules ci-dessous et répondez aux questions:

<https://youtu.be/5AVY6E-7yCc>

<https://youtu.be/JqweQE6TDTc>

Question 1: Pour chaque type de réseau, vous donnerez:

- A. la définition de son acronyme
- B. une explication de la nature du réseau
- C. les équipements réseaux constitutifs
- D. un schéma réalisé à la main (prenez une photo du schéma que vous avez réalisé sur une feuille et collez la dans ce document)

Renseignez votre réponse ici

LAN:

A.

B.

C.

D.

Renseignez votre réponse ici

MAN:

A.

B.

C.

D.

Renseignez votre réponse ici

WAN:

A.

B.

C.

D.

Renseignez votre réponse ici

Internet:

A.

B.

C.

D.

2. Simulation de réseaux

Nous allons travailler avec logiciel Filius qui permet de simuler assez simplement des réseaux, en intégrant des ordinateurs, des câbles , des équipements, des configurations, des installations logiciels et des exécutions de commandes réseaux.

La prise en main est proposée à travers le lien ci-dessous du site pixees:

https://pixees.fr/informatiquelycee/n_site/snt_internet_sim1.html

Pour les utilisateurs Linux et Mac OS, ci-dessous une aide pour l'installation:

<http://128mots.com/index.php/2019/11/10/filius-simulateur-reseau/>

Vous devez installer le logiciel puis réaliser les activités proposées et joindre vos fichiers filius à votre devoir classroom et répondre aux questions ci-dessous.

Question 2: réseau 0: relier 2 ordinateurs portables

- A. Réaliser le réseau 0 sur filius
- B. Joindre à votre devoir classroom le fichier filius sous le nom : “ votre nom reseau 0.flis”
- C. décrire les opérations réalisées et le réseau simulé
- D. coller une prise d'écran de votre schéma

Renseignez votre réponse ici

C.

D.

Question 3: réseau 1: relier 4 ordinateurs portables (*À faire vous-même 1*)

- A. Réaliser le réseau 1 sur filius
- B. Joindre à votre devoir classroom le fichier filius sous le nom : “ votre nom reseau 1.flis”
- C. décrire les opérations réalisées et le réseau simulé
- D. coller une prise d'écran de votre schéma

Renseignez votre réponse ici

C.

D.

Question 4: réseau 2: relier 2 réseaux de 4 ordinateurs portables

- A. Réaliser le réseau 2 sur filius
- B. Joindre à votre devoir classroom le fichier filius sous le nom : “ votre nom reseau 2.flis”
- C. décrire les opérations réalisées et le réseau simulé
- D. coller une prise d’écran de votre schéma

Renseignez votre réponse ici

C.

D.

Question 5: réseau 3: relier 3 réseaux de 2 ordinateurs portables (*À faire vous-même 2*)

- A. Réaliser le réseau 3 sur filius
- B. Joindre à votre devoir classroom le fichier filius sous le nom : “ votre nom reseau 3.flis”
- C. décrire les opérations réalisées et le réseau simulé
- D. coller une prise d’écran de votre schéma

Renseignez votre réponse ici

C.

D.

Question 6: réseau 4: un réseau un peu plus complexe (*À faire vous-même 3*)

- A. Réaliser le réseau 4 sur filius
- B. Joindre à votre devoir classroom le fichier filius sous le nom : “ votre nom reseau 4.fls”
- C. décrire les opérations réalisées et le réseau simulé
- D. coller une prise d'écran de votre schéma

Renseignez votre réponse ici

C.

D.

3. Topologie de réseaux

Vous avez vu lors de votre manipulation de Filius, qu'à partir du moment où vous avez beaucoup d'équipements, il y a plusieurs "manière" de les relier entre eux. Cela s'appelle une topologie de réseau et nous allons les étudier dans cette partie.

Il existe des typologies types qui permettent de définir des architectures réseaux en tirant profit, selon les parties du réseau, de leurs avantages/inconvénients en terme de coût, de complexité, de robustesse, de sécurité,...

Question 7: Pour chaque topologie type de réseau:

- A. Coller une prise d'écran d'un schéma
- B. Décrire la topologie
- C. Préciser les avantages
- D. Préciser les inconvénients

Renseignez votre réponse ici

Réseau en anneau:

- A.
- B.
- C.
- D.

Réseau hiérarchique:

- A.
- B.
- C.
- D.

Réseau en bus:

- A.

B.

C.

D.

Réseau en étoile:

A.

B.

C.

D.

Réseau maillé:

A.

B.

C.

D.

Question 8: Décrire la topologie du réseau internet

Renseignez votre réponse ici

4. Protocoles et Couches Réseaux

Nous avons maintenant une idée des architectures réseaux qui vont permettre de transmettre des informations.

Mais concrètement, comment les données sont transmises depuis l'expéditeur jusqu'au destinataire?

Si j'envoie un mail par exemple, est ce que toutes les données du mail sont envoyées dans un seul "message" sur le réseau?

Comment le réseau va retrouver le bon destinataire?

Comment est organisé le "dialogue" entre les deux machines?

Comment est-on assuré que toutes les machines se comprennent?

Question 8: Répondre aux questions précédentes après avoir visionné les vidéos ci-après:

<https://youtu.be/s18KtOLpCg4>

<https://youtu.be/5AVY6E-7yCc>

<https://www.youtube.com/watch?v=qzWdzAvfBoo>

Renseignez votre réponse ici

Question 9: après avoir visionné la vidéo ci-après, répondre aux questions:

<https://youtu.be/t3NZsApAfQA>

Renseignez votre réponse ici

- A. Représenter les couches du modèle OSI

- B. Représenter les couches du modèle TCP/IP

- C. Indiquer la correspondance des couches entre les deux modèles
- D. Expliquer l'encapsulation et la décapsulation .
- E. Définir "protocole réseau"

- F. Lister des protocoles réseaux en indiquant leur rôle ainsi que la couche associée dans le modèle TCP/IP.

Question 10: simuler sur Filius des protocoles de couches basses et décrire les opérations réalisées ainsi que vos observations (vous pouvez consulter la vidéo <https://youtu.be/JpntUqJB4Fc> ainsi que le guide...)

Renseignez votre réponse ici

Question 11: simuler sur Filius des protocoles de couches hautes et décrire les opérations réalisées ainsi que vos observations (vous pouvez consulter la vidéo <https://youtu.be/JpntUqJB4Fc> ainsi que le guide...)

Renseignez votre réponse ici

5. Protocole du bit alterné

Différents protocoles, sur différentes couches réseaux, existent pour détecter et corriger des erreurs dans la transmission de données. Ceci est nécessaire car il n'y a aucune ressource matérielle ou logicielle fiable à 100%, même si les taux d'erreurs des systèmes actuels sont très bas.

Le programme officiel de NSI indique de traiter le protocole du bit alterné.

C'est un protocole qui intervient au niveau de la couche liaison de données (ou accès réseau), qui n'est pas utilisé dans la réalité mais qui est assez simple.

Il intervient pour fiabiliser des échanges après établissement d'une connexion entre deux machines.

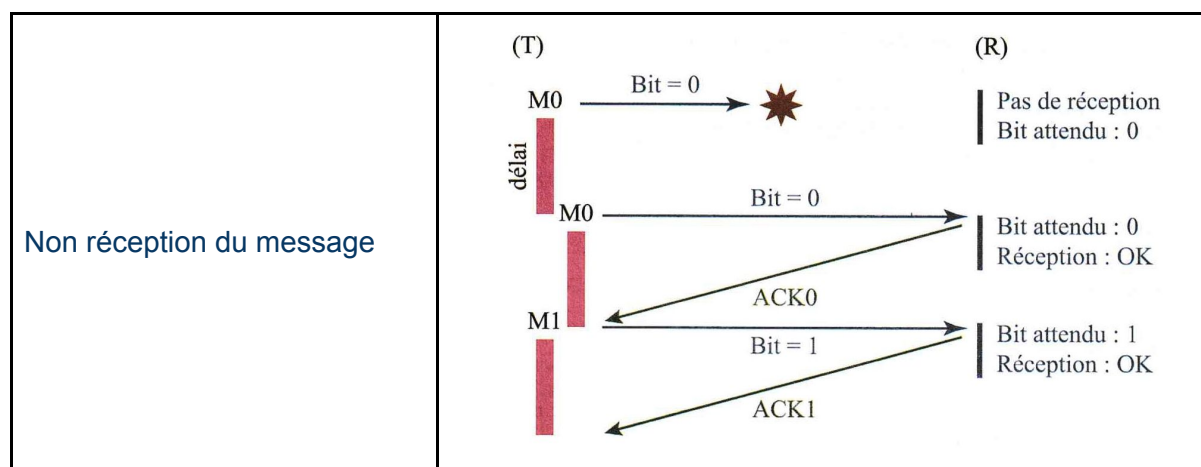
Le principe est d'accompagner chaque message par un bit de contrôle qui va alterner entre 0 et 1.

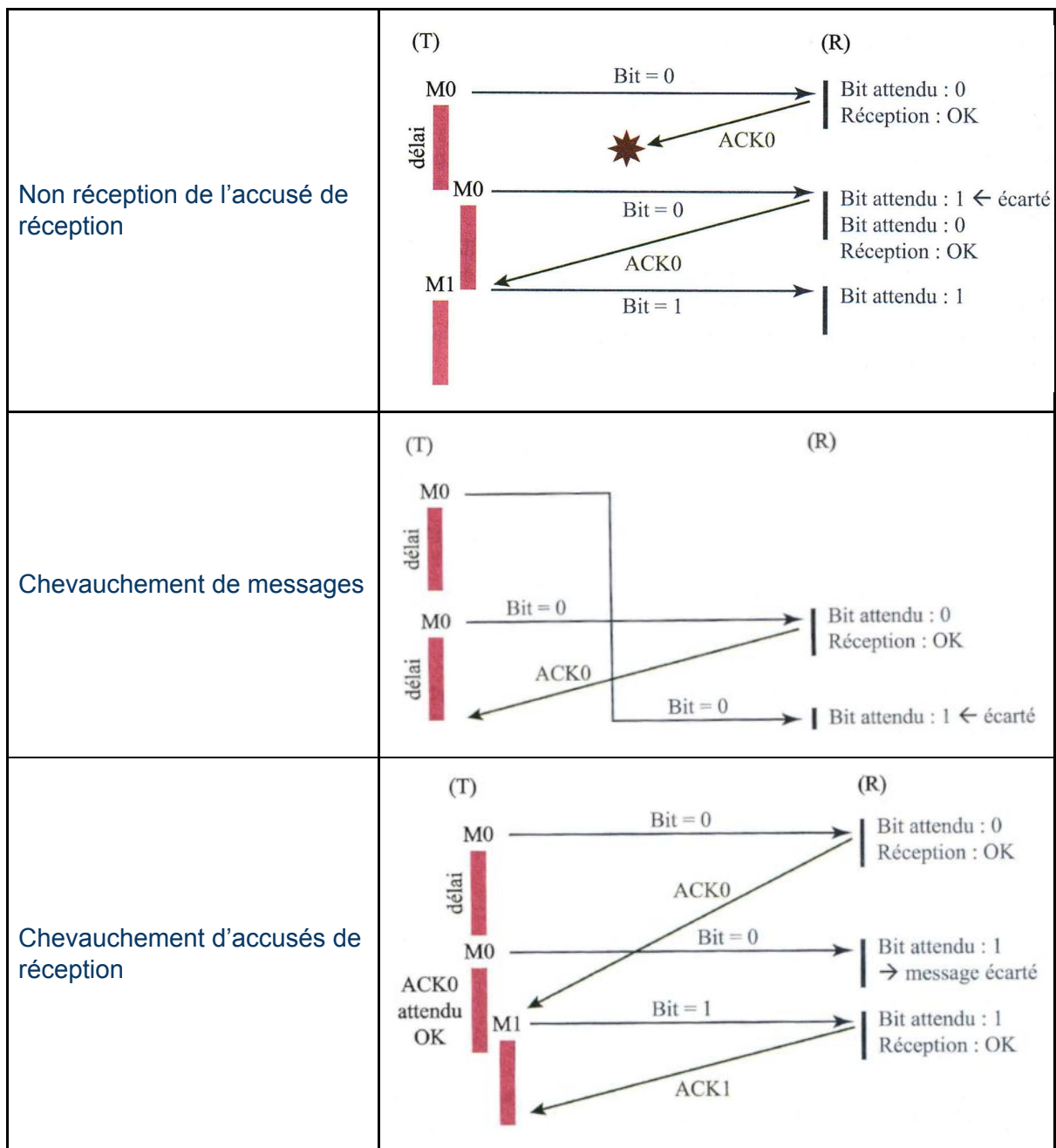
Si le message est reçu, le destinataire répond à l'expéditeur avec un accusé de réception (ACK: acknowledgment) avec le même bit de contrôle.

Si l'expéditeur reçoit le bon ACK, il considère que le message a bien été reçu, et il envoie le message suivant.

Si l'expéditeur ne reçoit pas le bon ACK, ou s'il ne reçoit pas d'ACK au bout d'un délai, il renvoie le message.

Voici quelques scénarii:





Question 12: Schématiser, comme ci-dessus, le scénario où les deux premiers messages sont reçus correctement, puis la connexion est rompue (juste après l'envoi du troisième message) pendant 2 délais d'attente, puis la connexion est rétablie.

Renseignez votre réponse ici

6. Et sur Python?

Nous avons vu que le système d'exploitation (SE) permettait aux applications des utilisateurs d'accéder aux ressources de la machine à travers le shell, et que les langages de programmation peuvent faire appel à des services du SE.

Donc, de manière similaire, on peut accéder aux services réseaux à travers le shell du SE et à travers un langage de programmation:

- couche réseau: ping, ifconfig, arp, traceroute, netstat, nslookup, ...
- couche transport: TCP, UDP, ...
- couche application: telnet, ssh, ftp, sftp, smtp, http, https, ...

Nous pouvons maintenant faire de la programmation réseau et créer des applications qui communiquent (message, email, partage, jeu en ligne, réseau social, plateforme collaborative, ...)

Voilà comment créer des connexions TCP (mode connecté) et UDP (mode non connecté) sous Python en mode Client/serveur et échanger des messages sur ces connexions:

Import de la bibliothèque	<pre>from socket import *</pre>
Création d'une socket	<pre><i># création d'une socket en général</i> mySocket = socket (socket_family, type_of_socket, protocol=value) <i># création d'un socket de type TCP</i> myTCPSocket=socket(AF_INET, SOCK_STREAM) <i># création d'un socket de type UDP</i> myUDPSocket=socket(AF_INET, SOCK_DGRAM)</pre>
Création et lancement du serveur	<pre><i># crée la socket du serveur puis la lie (bind) à l'ip de notre machine et à un port de notre choix</i> HOST_NAME="localhost" <i># ip de la machine (ou nom résolvable)</i> PORT_NUMBER= 1111 <i># port de la connexion (entre 1024 à 49151)</i> serveur=socket(AF_INET, SOCK_STREAM) <i># création d'une socket TCP</i> serveur.bind((HOST_NAME, PORT_NUMBER)) <i># permet au serveur d'accepter des connexions clients</i> serveur.listen() <i># on peut ajouter en argument le nombre max de clients</i> """" <i>se met en attente d'un client et renvoie un tuple:</i> - conn : socket de connexion entre le serveur et le client</pre>

	<pre>- adresse: l'adresse du client """" conn , adresse = serveur.accept()</pre>
Création et connexion d'un client	<pre># crée la socket du client puis se connecte au serveur SERVEUR_NAME="localhost" # ip du serveur (ou nom résolvable) PORT_NUMBER= 1111 # port de la connexion (entre 1024 à 49151) client=socket(AF_INET, SOCK_STREAM) # création d'une socket TCP client.connect((SERVEUR_NAME, PORT_NUMBER)) #</pre>
Envoi et réception de messages	<ul style="list-style-type: none"> • Sur le programme client: <pre># envoie un message au serveur client.send("je suis un client".encode()) # on envoie des bytes d'où la méthode encode()</pre> • Sur le programme serveur: <pre>BUF_SIZE=1024 # la taille maximale pour la réception d'un message (en bytes) # réception du message par le serveur message=conn.recv(BUF_SIZE).decode() print(message) # envoie d'un message au client conn.send("Réponse du serveur:\n j'ai bien reçu ton message".encode())</pre>

Question 13: On commence par créer deux programme "serveur.py" et "client.py" qui établissent une connexion TCP entre eux et affichent "connexion établie" avec les références de la connexion.

Renseignez votre réponse ici

NB: les deux programmes tournent sur la même machine, éventuellement vous pouvez les faire tourner sur deux machines du même sous réseau, mais pour les faire tourner sur des machines de sous réseaux différents, il faudrait configurer les routeurs (voir NAT)

Question 14: On ajoute un envoi de message par le client, sa réception par le serveur et son affichage

Renseignez votre réponse ici

Question 15: On ajoute une communication en boucle entre des clients et le serveur

Renseignez votre réponse ici

En option:

Question 16: vous pouvez maintenant créer une application en réseau, par exemple un pendu, un jeu de puissance 4, une messagerie instantanée, une application collaborative, un jeu en réseau, ...

Renseignez votre réponse ici

Vous trouverez une vidéo explicative au lien ci-dessous:

<https://youtu.be/wUnamYAv8TE>