

Module de réseaux

Pédagogie inversée

All you need is autonomy and ethics



<https://www.placegrenet.fr/2017/01/13/nouveau-episode-4-clinatec-en-six-polemiques/114612>

Origin ?? Copyright ?? Prohibited reproduction ??

Christian Toinard

Professeur des Universités en informatique

What else ? (education, professional skills, teaching and research experiences,
...)

Objectifs

Before technics is autonomy and ethics !! (*could said H.G. Thoreau*)

- Permettre aux **élèves** d'être **autonomes** et au final de retenir et comprendre
- Permettre aux élèves d'avoir une **culture suffisante**
- Partager via chaque cours un **savoir collectif**
- Le savoir définitif n'existe pas : il faut acquérir **méthode et rigueur**
- Attitude d'un futur ingénieur : autonome, responsable, modeste, respectueux et citoyen

Thèmes

- **Programmation réseau** au moyen de l'API socket (application en mode connecté et non connecté, limites et formats des messages, ruptures de connexion, gestion des activités concurrentes)
- **Transmissions réseau et architecture OSI** (transmission asynchrone et synchrone, temps de transmission, débits et taux d'utilisation, transmission fiable, modèle OSI, notion de protocole et service, rôle des 7 niveaux, enveloppe et encapsulation, études de cas FTP/HTTP, protocoles fiables liaison et transport)
- **Programmation d'applications réparties** (résultat d'impossibilité, notion d'état, différence avec le mode centralisé, appel de procédure à distance, traitement de l'hétérogénéité, gestion des pannes, études de cas RPC sous Unix et REST pour le Web)
- **Administration réseau** (l'Internet Protocole, interfaces réseau, switchs, routeurs, passerelles et parefeux, études de cas routage et translation d'adresse sous Unix et parefeu sous Unix)

Séances de pédagogie inversée

- Programmation réseau : 6 cours et 5 TDs
 - Transmissions réseau et architecture OSI : 5 cours et 5 TDs
 - Programmation d'applications réparties : 2 cours et 2 TDs
 - Administration réseau : 2 cours et 3 TDs
- Total : 15 cours et 15 TDs

Répartition

- Cours :
 - 5 étudiants pour les 5 premiers cours. 4 étudiants pour les 10 cours suivants. Soit 65 étudiants.
- TDs :
 - Pour les 5 premières séances (programmation réseau) : 2 étudiants par TD soit $5 \times 2 \times 3(\text{groupes}) = 30$
 - Pour les 8 séances suivantes : 1 étudiant par TD soit $8 \times 1 \times 3(\text{groupes}) = 24$
 - Pour les 2 dernières séances (administration) : 2 personnes par TD soit $2 \times 2 \times 3 - 1 = 11$ étudiants + 1 enseignant

Méthode

- **Objectifs définis par les enseignants** via le présent document (le lire avec attention...).
- Chacune des 15 séances de cours ou de TD propose une **équipe** avec un **coordinateur**.
- **Fournitures** par le coordinateur aux enseignants des supports nécessaires avant la séance. L'enseignant mets les supports disponibles à tous sur Celene.
- Chaque **étudiant d'une équipe doit s'investir** dans la séance et rendre compte aux élèves coordinateurs et aux enseignants de leurs contributions (l'éthique paiera !!).
- **Livraisons sur Celene aux échéances** faites par les **coordinateurs** qui collecte, harmonise et rend compte des contributions de chaque membre (le respect des échéances paiera !!).
- **Chaque équipe dialogue et coopère** avec les enseignants pour fournir une intervention de qualité.
- **Appropriation par tous des supports** (transparents de cours, sujets de TD, codes, documentations, consignes, ...) avant la séance.

Evaluation bipartie continue

- Chaque cours et TD donne lieu à une note
- La note comprends une **évaluation par les pairs** et une **évaluation par l'enseignant**
- **L'évaluation par les pairs** est faite de façon **anonyme** mais l'enseignant a entière visibilité sur le travail d'évaluation
- **La note de l'enseignant** tient compte de la qualité de **l'intervention** et de **l'évaluation** mais aussi de **l'intégrité !!**

Evaluation par les pairs

- Chaque **cours** donne lieu à un **quizz noté** par l'équipe.
- **Chaque TD est noté** par l'équipe selon le travail rendu sur Célène.
- Chaque **élève note** anonymement **les membres de l'équipe** qui sont intervenus.
- Chaque membre note **une proportion** du travail rendu par les élèves (un rendu par élève).
- Toute **absence de rendu** de quizz ou de TD à l'échéance vaut un **zéro**.

Evaluation par les enseignants

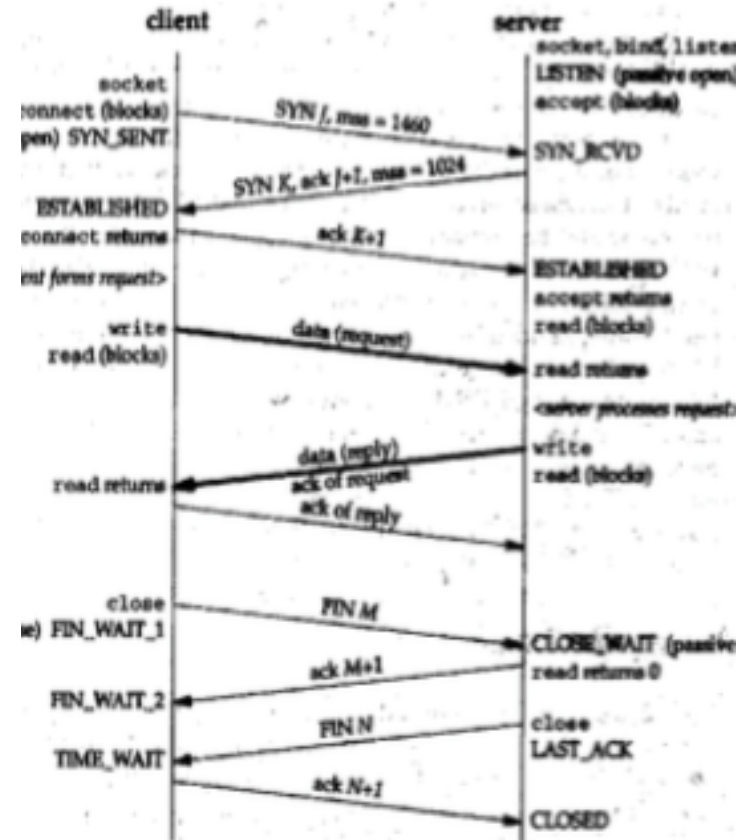
- L'enseignant qui suit le cours ou le TD note à la fois la **qualité de l'intervention** mais aussi **la qualité de l'évaluation rendue**.
- La **qualité de l'intervention** tient compte de l'implication et des résultats fournis (qualité des recherches, des supports, du dialogue, ...).
- La **qualité de l'évaluation rendue** tient compte de la **moyenne et de l'écart type** (une moyenne dépassant 15 n'est pas sérieuse) mais aussi des **précisions données** (barème, commentaires pour les élèves, commentaires pour les enseignants...)
- Tout **manque d'intégrité ou d'honnêteté** diminuera significativement la note : **faire peu mais bien !!**

<http://lionnedatlas.over-blog.com/article-15756296.html>



Programmation réseau

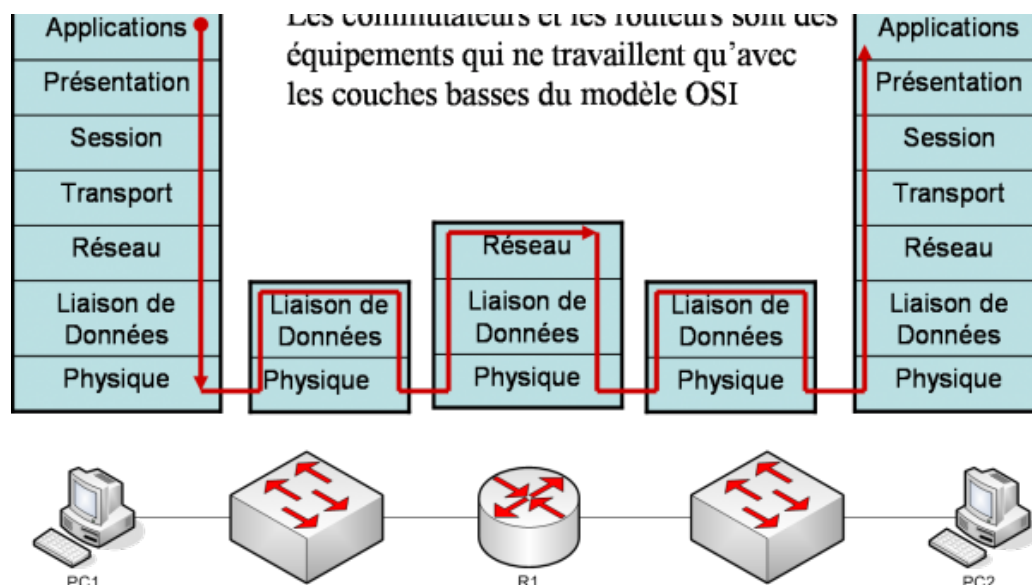
- Pourquoi des appels système et rien d'autre ?
- Est ce simple de faire les échanges réseau (SYN, SYN +ACK, PUSH, ...) ?
- Quelles sont les différentes versions des "piles" réseau existantes ?
- Comment et où est implantée une pile réseau ?
- Un programme informatique est il sûr ?
- Est ce qu'on peut faire quelque chose de déterministe en réseau ?



From **Richard Stevens** TCP/IP Book

Transmission réseau et OSI

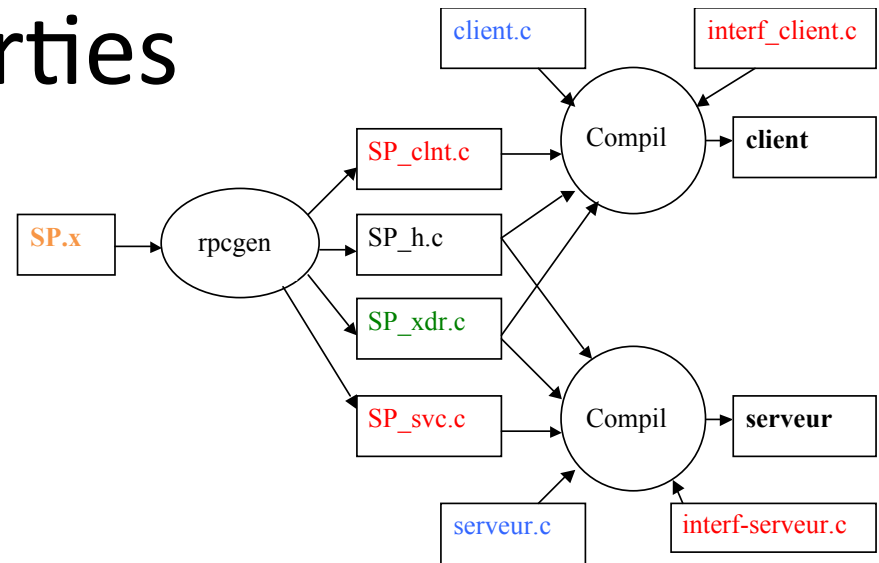
- A quoi servent ces différents niveaux ?
- Pourquoi est ce aussi compliqué ?
- Peut on le simplifier ?
- Comment faire pour que cela soit un peu fiable ?
- Quelles sont les parties du domaine de l'informatique ?
- Et Internet est il conforme à ce modèle ?



<http://reussirsonccna.fr/modele-osi/>

Programmation d'applications réparties

- Quel est le rapport avec le modèle OSI ?
- Est ce facile à utiliser ?
- Utilise t on cela encore de nos jours (nous vivons une époque formidable !!) ?
- Et les solutions modernes (REST, ...) ?
- Est ce applicable sur Internet ?
- Qu'est ce qu'il reste à faire (cela semble trop facile) ?



rdict.x:

```
const MAXSTRING=256;
const DICTSIZ=100;

struct DictRecord{
    int longueur;
    char donnee[MAXSTRING];
};

program RDICTPROG {
    version RDICTVERS {
        int    INITDICT() = 1;
        int    INSERTW(DictRecord) = 2;
        int    DELETEW(DictRecord) = 3;
        int    LOOKUPW(DictRecord) = 4;
    }=1;
} = 0x30090949;
```

Administration réseau

- Fait on de l'administration à tous les niveaux ?
- Est ce nécessaire pour des gens qui vont faire de la sécurité ?
- Y a t il de mauvais administrateurs ?
- Peut on être un expert réseau uniquement ?
- Que veut dire DevOps ?
- Quel est le lien entre l'administration, le développement et les usagers ?
- Pourquoi est ce important pour un ingénieur en informatique et encore plus pour quelqu'un qui veut faire de la sécurité (merci N. Grenèche) ?

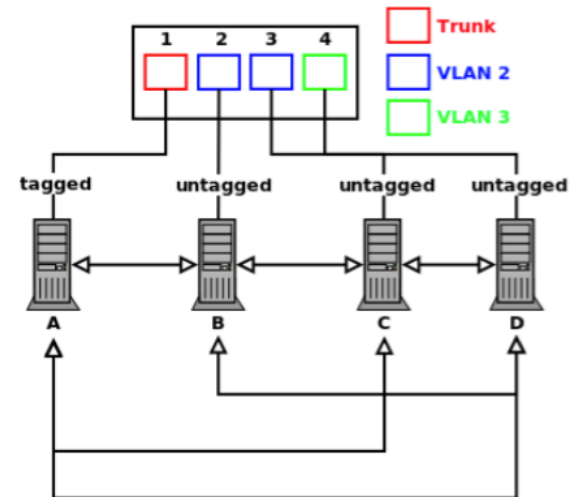


Table de routage IP du noyau

Destination	Passerelle	Genmask	Indic	MSS	Fenêtre	irtt	Iface
0.0.0.0	X.X.X.X	0.0.0.0	UG	0	0	0	eth1
192.168.3.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0.3
192.168.4.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0.4

