

Reseaux general

Fiche Architecture des réseaux et des ordinateurs

1 Introduction aux réseaux et modèles des réseaux en couches

Les réseaux sont constitués de plusieurs machines distantes: Problématique d'interopérabilité : normalisation des communications (standards, RFC, etc.)
Les communications réseaux impliquent beaucoup de mécanismes
Création de nombreux protocoles. Ils sont regroupés par fonctionnalités dans des couches
Les données sont encapsulées (ex : les données HTML sont encapsulées dans des données HTTP, qui sont encapsulées dans un paquet TCP, qui sont encapsulées dans un paquet IP, qui sont encapsulées dans une trame Ethernet)

Modèle OSI: Sept couches

● OSI = Open Systems Interconnection : Proposition en 1978, normalisation en 1984, Normalisé par l'ISO (International Standard Organization)

Couches	Protocoles	Utilité
Physique (manipule des bits) (basse)		● Modulation = transformation d'un signal binaire en un signal adapté au médium (ex : sinusoïde paramétrée avec amplitude / phase / fréquence) ; ● Codage = manière dont les bits sont encodés (ex : codage de Manchester, codage de Manchester différentiel, codage NRZ, etc.)
Liaison de données (manipule des trames) (basse)		Adressage physique et partage du temps de parole : ● Détection et correction d'erreur, retransmission ● Tour de rôle, accès multiple avec contention
Réseau (manipule des paquets) (intermédiaire)		Adressage logique, routage et acheminement ● Routage = calculer la route à suivre de la source à la destination ● Acheminement = suivre la route calculée
Transport (manipule des segments) (intermédiaires)		Couche transport : communication de bout en bout : ● Multiplexage
Session (manipule des données) (haute)		gère la reconnexion automatique
Présentation (manipule des données) (haute)		Encodage des données : ● Codage ● Compression ● Chiffrement
Application (manipule des données) (haute)		Non spécifiée

Modèle Internet (ou TCP/IP) : Conçu en 1976, mais non normalisé. Quatre couches :

- 1● Couche accès réseau (= couche physique + couche liaison de données)
- 2● Couche Internet (= couche réseau)
- 3● Couche transport (= couche transport)
- 4● Couche application (= c session + c présentation + c application)

2 Notions de base, différents protocoles Internet :

IP = Internet Protocol (1981) ● Fonctionnalités principales : adressage et acheminement ● Pas de routage : rôle des protocoles de routage (RIP, OSPF, BGP, etc.) ● Adressage : ● IPv4 = 32 bits, IPv6 = 128 bits ● Notation à points ● Adressage IPv4 ● Classes A, B et C (et D et E) ● Masques de réseaux, adresse de réseaux, notation " /x " pour les x premiers bits à 1 dans le masque ● CIDR (= Classless Inter-Domain Routing) Acheminement = longest common prefix match : ● Notion de route par défaut ● Notion de passerelle (plutôt pour les équipements terminaux) Autres fonctionnalités : fragmentation, détection d'erreurs						
Schema TAB						
ICMP = Internet Control Message Protocol (1981); Fonctionnalité : gestion des messages de contrôle d'IP Types de messages ICMP : ● Echo request et echo reply (ping) ● Destination unreachable ● TTL exceeded ● Timestamp request et timestamp reply ● Etc.						
ARP = Address Resolution Protocol (1982) Fonctionnalité : donner l'adresse MAC d'une machine du réseau dont l'adresse IP est connue : Utilisé pour l'acheminement de proche en proche						
UDP = User Datagram Protocol (1980) ● Fonctionnalités principales : Multiplexage, au travers de ports (= numéros servant à identifier le service (côté serveur) ou l'application source (côté client))						
● Format de l'entête UDP	<table><tr><th>Port source (16 bits)</th><th>Port destination(16 bits)</th></tr><tr><td>Longueur</td><td>Somme de contrôle</td></tr></table>	Port source (16 bits)	Port destination(16 bits)	Longueur	Somme de contrôle	
Port source (16 bits)	Port destination(16 bits)					
Longueur	Somme de contrôle					
Avantage/Inconvénient: La création d'un client/serveur UDP est facile a priori, car il y a peu d'appels systèmes. Mais, il faut gérer les pertes éventuelles de paquets, la fragmentation, etc. UDP est un protocole rapide (autant qu'IP) mais non fiable (pas plus qu'IP). UDP est adapté aux communications temps-réel (visio-conférence, audioconférence, jeux, etc.)						
TCP = Transmission Control Protocol (1980) Fonctionnalités principales : ● Multiplexage, au travers de ports ● Mode connecté : garantie de livraison + non duplication + ordonnancement ● Contrôle de flux, contrôle de congestion						
SCHEMA FORMAT						
● La création d'un client/serveur TCP est un peu plus compliqué a priori qu'avec UDP. Mais, le développement du reste de l'application est beaucoup plus simple ● TCP est un protocole lent (par rapport à IP et UDP) mais fiable (contrairement à IP ou UDP). TCP est adapté aux communications fiables (transfert de fichier, transfert de texte)						
DNS = Domain Name System (1983). Utilise le port 53 (en TCP et en UDP) Fonctionnalité principale : traduction de noms en adresses IP.Base de données distribuée (de manière hiérarchique) fournissant un service d'annuaire Enregistrements DNS : ● Adresses IPv4 (A) et IPv6 (AAAA), alias (CNAME) ● Serveurs de noms (NS) ● Serveurs SMTP (MX) ● Etc.						
<table><tr><td>Espace de noms = structure arborescente ● Chaque nœud a un nom et des enregistrements associés ● La descente dans l'arbre se fait en ajoutant un " . " ● Un nœud peut déléguer la responsabilité d'une zone à un serveur de noms ● Un domaine est un chemin, ou un nom lorsqu'il n'y a pas d'ambiguïté</td><td colspan="2" rowspan="2">Mécanisme de résolution d'adresses ● Un résolveur détermine le serveur de noms responsable d'un domaine en suivant le chemin dans l'arbre, en partant de la racine ● Des serveurs caches stockent ces informations temporairement ● Les serveurs caches implémentent souvent eux-mêmes les requêtes récursives</td></tr></table>			Espace de noms = structure arborescente ● Chaque nœud a un nom et des enregistrements associés ● La descente dans l'arbre se fait en ajoutant un " . " ● Un nœud peut déléguer la responsabilité d'une zone à un serveur de noms ● Un domaine est un chemin, ou un nom lorsqu'il n'y a pas d'ambiguïté	Mécanisme de résolution d'adresses ● Un résolveur détermine le serveur de noms responsable d'un domaine en suivant le chemin dans l'arbre, en partant de la racine ● Des serveurs caches stockent ces informations temporairement ● Les serveurs caches implémentent souvent eux-mêmes les requêtes récursives		
Espace de noms = structure arborescente ● Chaque nœud a un nom et des enregistrements associés ● La descente dans l'arbre se fait en ajoutant un " . " ● Un nœud peut déléguer la responsabilité d'une zone à un serveur de noms ● Un domaine est un chemin, ou un nom lorsqu'il n'y a pas d'ambiguïté	Mécanisme de résolution d'adresses ● Un résolveur détermine le serveur de noms responsable d'un domaine en suivant le chemin dans l'arbre, en partant de la racine ● Des serveurs caches stockent ces informations temporairement ● Les serveurs caches implémentent souvent eux-mêmes les requêtes récursives					
Gestion des dépendances circulaires : enregistrements " glue " (glue records) ● Exemple : Le serveur de noms du domaine " isima.fr. " est " ns.isima.fr. ". Pour connaître l'adresse IP de " www.isima.fr. ", un utilisateur obtient le nom " ns.isima.fr. ", qu'il doit résoudre. Mais pour résoudre " ns.isima.fr. ", qui est dans la zone " isima.fr. ", il faut résoudre " ns.isima.fr. ". On a une dépendance circulaire. ● Les dépendances circulaires sont évitées par les enregistrements glue : l'adresse IP du serveur de noms est donnée dans la réponse.						

3 Protocoles web, applicatifs classiques

<p>HTTP = HyperText Transfer Protocol (1996):</p> <ul style="list-style-type: none">• Utilise le port 80 (parfois 8000 ou 8080)• Un hypertexte est un texte pouvant contenir des liens vers d'autres ressources (incluant d'autres textes) <p>Fonctionnalité principale : transférer des fichiers hypertextes</p> <ul style="list-style-type: none">• Protocole synchronisé de type requête-réponse• Un client demande un accès à des ressources HTTP, en utilisant des URL (Uniform Resource Locators). Les URL se décomposent en une adresse de serveur et une URI (Uniform Resource Identifier).• HTTP permet de transférer des fichiers textes (généralement au format HTML) ou des fichiers binaires (images, sons, vidéos)	
<p>les requêtes : Format des requêtes</p> <ul style="list-style-type: none">• Une ligne contenant la requête (avec la méthode) • Des lignes d'entête de requête (optionnelles) • Une ligne vide • Un corps éventuel (optionnel) <p>Méthodes de requêtes</p> <ul style="list-style-type: none">• GET : permet de demander une ressource • POST : pour soumettre une ressource (ou un formulaire) • OPTIONS : pour obtenir les méthodes supportées par le serveur pour une certaine URI • Etc. <p>Entêtes de requêtes : • Accept (exemple : text/html) • Accept-Encoding (exemple : gzip) • Cookie (voir plus loin) • Host (utile en cas de proxy, obligatoire depuis HTTP 1.1) • User-Agent</p>	<p>les réponses</p> <p>Format des réponses : • Une ligne contenant le code de réponse et un message • Des lignes d'entête de réponse (optionnelles) • Une ligne vide • Un corps de réponse (optionnel)</p> <p>Codes de réponse : • 1XX = Information • 2XX = Succès • 3XX = Redirection • 4XX = Erreur client • 5XX = Erreur serveur</p> <p>Entêtes de réponse : • Content-Encoding (exemple : gzip) • Content-Language (exemple : fr, en) • Content-Length • Content-Type (exemple : text/html; charset=utf-8) • Date • Expires • Location (en cas de redirection)</p>
<p>les cookies</p> <p>Certaines applications HTTP nécessitent de conserver un état • Exemple : une information d'authentification, un panier d'achats • Le serveur stocke associe les informations de connexion à une chaîne de caractères représentant la session de l'utilisateur • Le client stocke le numéro de la session dans un fichier texte, appelé cookie, et transfère ce numéro de session à chaque requête</p> <p>EXEMPLE HTTP</p>	
<p>FTP = File Transfer Protocol (1985)</p> <p>Fonctionnalité principale : transfert de fichiers:</p> <ul style="list-style-type: none">• Connection séparée pour les commandes (port 21) et les données (généralement port 20)• Dans le mode actif (par défaut), le client informe le serveur du port sur lequel il écoute pour recevoir les données (avec la commande PORT)• Dans le mode passif (déclenché avec la commande PASV), c'est le serveur qui informe le client du port choisi	
<p>Liste des commandes du client</p> <ul style="list-style-type: none">• USER et PASS pour le login et le password• CWD et CDUP pour changer de répertoire (descendre ou monter)• DELE pour effacer un fichier• HELP pour l'aide• LIST et NLST pour les informations sur le répertoire ou le fichier• PASV et LPSV pour entrer en mode passif (court ou long)• PORT pour informer de l'adresse et du port pour le transfert de données <ul style="list-style-type: none">• PWD pour connaître le répertoire actuel • RETR pour télécharger un fichier• TYPE pour changer le mode de transfert (ASCII ou binaire) <p>EXEMPLE</p>	<p>Liste des codes de réponse du serveur</p> <ul style="list-style-type: none">• 1XX : réponse positive (début + milieu)• 2XX : réponse positive (fin)• 3XX : réponse positive en cours (en attente d'une autre requête)• 4XX : réponse négative transitoire• 5XX : réponse négative permanente• 6XX : réponse protégée• X0X : syntaxe• X1X : information• X2X : connexion• X3X : authentification• X5X : système de fichiers

4 Protocoles email

<p>SMTP = Simple Mail Transfer Protocol (1982), port 25</p> <p>Fonctionnalité : transmission d'emails (mais pas la consultation d'emails)</p> <p>Commandes du client SMTP</p> <ul style="list-style-type: none">• Commande MAIL : informe de l'adresse de l'émetteur• Commande RCPT : informe de l'adresse du destinataire• Commande DATA : informe du contenu du message (terminé par une ligne contenant uniquement " . ") <p>Réponses du serveur SMTP</p> <ul style="list-style-type: none">• 2XX (réponse positive), 4XX (réponse négative transitoire) ou 5XX (réponse négative permanente) <p>EXEMPLE</p>	
<p>POP3 = Post Office Protocol version 3 (1988), port 110</p> <ul style="list-style-type: none">• Fonctionnalité : récupération/consultation d'emails• Protocole requête-réponse très simple• Les messages sont a priori transférés du serveur vers l'ordinateur de l'utilisateur <p>EXEMPLE</p>	<p>IMAP = Internet Message Access Protocol (2003), port 143</p> <ul style="list-style-type: none">• Fonctionnalité : consultation d'emails• Permet au même utilisateur de se connecter simultanément depuis plusieurs clients• Permet de télécharger des emails partiellement• Permet de gérer des répertoires• Permet de faire des recherches côté serveur <p>EXEMPLE</p>

5 Exemples Socket et manipulation réseaux