Réseaux

Franck Pommereau

http://goo.gl/zLKg
pommereau@ibisc.fr

Licence 3, Université d'Évry Val d'Essonne

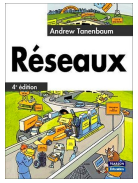
Support de cours et accompagnement

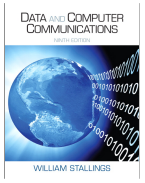
http://goo.gl/zLKg

Objectifs

- principes de fonctionnement des réseaux
- organisation en couches
- principaux protocoles
- fonctionnement d'Internet
- aspects pratiques

Références



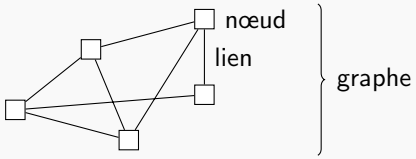


- A. S. Tanenbaum. *Réseaux*. Pearson Education, 2003.
- W. Stallings. *Data and Computer Communications*. Prentice Hall, 2011.
- The Internet Engineering Task Force. *Request for comments*. http://www.ietf.org/rfc.html

De quoi parle-t-on ?

Réseau

Ensemble de machines interconnectées et échangeant des données



Protocole

Règles de communication permettant l'échange de données

- mode connecté lien logique maintenu pendant la communication
  - ☞ *appel téléphonique*
- mode déconnecté envoi et réception d'un message en différé
  - ☞ *courrier postal*

Topologies

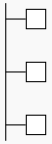
Organisation des nœuds et des liens — logique ou physique

Point à point

- limité à deux nœuds
- extensible si les nœuds le permettent

Bus

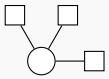
- lien partagé
- facilement extensible
- conflits possibles



Topologies
. . . suite (1/2)

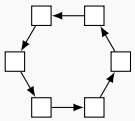
Étoile

- utilise un organe central d'interconnexion
- extensible dans la limite de ce nœud



Anneau

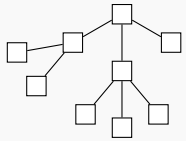
- liens orientés
- extensible au prix d'une coupure temporaire
- topologie logique le plus souvent
- anneau à jeton : circulation d'un "droit de parole"



Topologies
. . . suite (2/2)

Arbre

- topologie logique le plus souvent
- hiérarchique ou non
- simplifie le routage



Maillage

- topologie quelconque

- petits réseaux ⇒ applicables physiquement
- sinon ⇒ topologies logiques superposées au réseau physique
  - ☞ *arbre couvrant : arbre logique reliant tous les nœuds d'un maillage*

Familles de réseaux

LAN : *Local Area Network*

- réseau de petite taille, zone géographique définie — bâtiment
- généralement privé, appartenant à une même organisation
- généralement limité à une seule technologie
- souvent assez rapide, volumes limités

MAN : *Metropolitan Area Network*

- zone géographique plus vaste — ville
- généralement limité à une seule technologie
- interconnecte des LAN de technologies hétérogènes
- souvent très rapide, gros volumes

Familles de réseaux
. . . suite (1/3)

WAN : *Wide Area Network*

- très grande échelle — pays, continents, monde
- interconnecte des LAN et des MAN
- rapidités variables, très gros volumes
- WAN et MAN tendent à converger

Internet

- interconnexion mondiale de réseaux
- utilisant des protocoles standards — basés sur IP (*Internet Protocol*)
- d'autres réseaux mondiaux existent
  - ☞ *réseaux militaires*

Familles de réseaux
. . . suite (2/3)

Intranet

- réseau privé
- utilise les technologies et services typiques d'Internet
  - ☞ *site web, email, etc., internes à une entreprise*

Extranet

- partie d'un intranet ouverte sur l'extérieur
- nécessite un point d'entrée
  - ☞ *site web de réservation de ressources*

Familles de réseaux
. . . suite (3/3)

VPN : *Virtual Private Network*

- réseau logique utilisant un réseau physique comme support
- technologies cryptographiques ⇒ échanges privés
  - ☞ *différentes filiales d'une entreprise, reliées par Internet, mais partageant leur Intranet*

Commutation de paquets ou de circuits

- commutation de circuits — établissement préalable d'un lien
- commutation de paquets — acheminement indépendant de fragments
- utilisations différentes — voix *vs* informatique
- convergence vers la commutation de paquets — voix sur IP

Modèles en couches

Modèle OSI (*Open Systems Interconnection*)

- 7 : application — interface avec l'utilisateur
- 6 : présentation — format d'échange des données
- 5 : session — séquencement des communications
- 4 : transport — découpage des données et fiabilisation
- 3 : réseau — acheminement des données — adressage, routage
- 2 : liaison de données — partage et fiabilisation du médium
- 1 : physique — transmission effective d'un signal — médium

- ▸ chaque couche s'appuie sur le service rendu par la précédente
- ✘ modèle assez théorique
- ✔ le concept est le bon

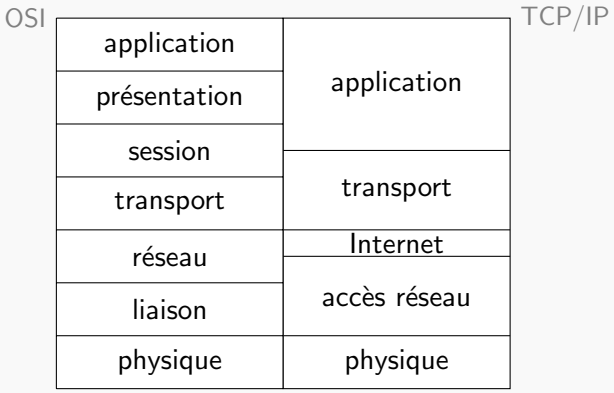
Exemple

Envoi d'un email dans le modèle OSI

- 7 : application — l'utilisateur tape son message et ajoute des pièces jointes dans Thunderbird, puis clique sur *envoyer*
- 6 : présentation — Thunderbird formate le message (texte et pièces jointes) selon le standard MIME
- 5 : session — le protocole SMTP est utilisé pour transmettre ces données formatées au serveur d'envoi
- 4 : transport — le protocole TCP permet le dialogue entre Thunderbird et le serveur SMTP
- 3 : réseau — le protocole IP achemine les fragments du flux TCP
- 2 : liaison — le standard MAC et le protocole ARP identifient les machines qui communiquent avec IP
- 1 : physique — le protocole Ethernet réalise la communication sur un câble électrique

Des couches plus réalistes

Modèle TCP/IP

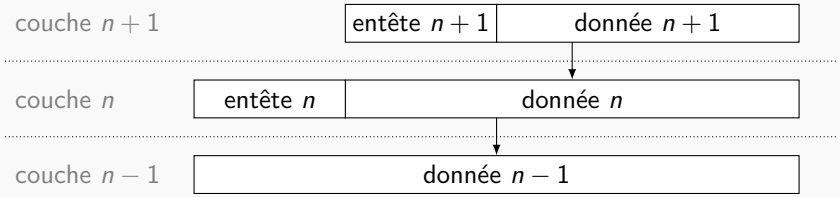


- ▸ plus proche de la réalité des systèmes existants
- ▸ le modèle OSI reste la référence comme guide de conception

Encapsulation des protocoles

- ▸ unité d'échange : entête + données — *header + payload*
  - ☞ *frames Ethernet, cellules ATM, paquets IP, datagrammes UDP, ...*
- ▸ (entête + données) à la couche $n+1$
  ⇒ passées comme données à la couche n
- ▸ données reçues de la couche n
  ⇒ décomposées comme (entête + données) par la couche $n+1$

Encapsulation entre les couches



Request for comments (RFC)

- définissent l'ensemble des protocoles d'Internet
- librement accessibles
  - `http://www.ietf.org`
  - `http://www.rfc-editor.org`
- c'est **la** référence en matière de protocoles, standards, etc.

Exemples de RFC

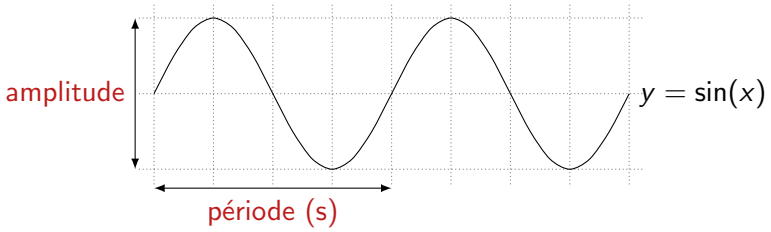
- RFC 791 *Internet Protocol* (IP)
- RFC 793 *Transmission Control Protocol* (TCP)
- RFC 1177 FAQ pour les utilisateurs débutants
- RFC 1207 FAQ pour les utilisateurs expérimentés
- RFC 1855 "Netiquette"

La couche physique

- différents supports de communication
  - transmission guidée (filaire)
  - ou non guidée (sans fil)
- le support propage des signaux
  - électromagnétiques : électriques, radio, lumineux, ...
  - peuvent être analogiques ou numériques (digitaux)
- la communication se fait dans un ou plusieurs sens
  - simplex uni-directionnel
  - half-dupex bi-directionel, chacun à son tour
  - full-duplex bi-directionel simultané
- les participants peuvent être reliés
  - en point-à-point ou en multipoint
  - de façon directe ou indirecte avec un dispositif intermédiaire (amplificateur, répéteur, ...)

Signaux périodiques

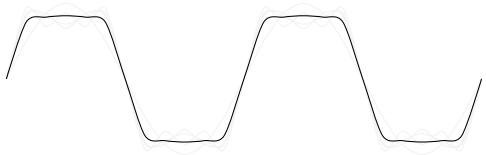
- un signal périodique se transporte plus efficacement
- tout signal peut être approché en composant des signaux sinusoïdes



- fréquence (Hz) 1/période
- phase (s) décalage par rapport à l'origine
- longueur d'onde distance parcourue pendant une période
  - vélocité × période

Signaux composites

$$y = \sin(x) + \sin(3x)/3 + \sin(5x)/5 + \sin(7x)/7 + \sin(9x)/9$$



- capacité du support ⇔ fidélité du signal
  - ✔ beaucoup de fréquences disponibles ⇒ reproduction fidèle
  - ✘ peu de fréquences disponibles ⇒ reproduction peu fidèle
- spectre plage de fréquences admissibles
- spectre efficace plage de fréquences fidèlement admises
- bande passante largeur du spectre

Transmission d'information

- débit (b/s) quantité d'information par seconde
- latence (s) délai entre l'envoi et la réception du premier bit
- taux d'erreurs (%) pourcentage de bits altérés

- bande passante ↗ ⇒ fréquences admissibles ↗ ⇒ débit ↗
  - ☞ *1 b/s nécessite 2 Hz*
- taux d'erreurs ↗ ⇒ débit ↘
  - ☞ *il faut retransmettre (et donc contrôler la qualité de la transmission)*

Dégradations du signal

- atténuation avec la distance
- distorsions — hautes amplitudes écrêtées
- distorsions de délai — vélocité différente selon la fréquence
- bruits divers — thermique, inter-modulation, interférences, perturbations, ...

Multiplexage

⇒ faire passer plusieurs canaux sur un seul

- division du spectre
  - plages de fréquences réservées au préalable
  - ✔ radio, ASDL
- multiplexage temporel synchrone
  - plages de temps réservées au préalable
  - nécessite une horloge globale
  - ✔ ATM
- multiplexage temporel statistique
  - division du temps selon le besoin
  - nécessite des adresses
  - n'utilise pas toute la capacité

Supports filaires

- paire torsadée deux fils de cuivre enroulés en tresse
  - ↘ interférences entre paires adjacentes
  - répartition uniforme des bruits
  - différentes qualités existent — blindées ou pas, ...
  - ✔ réseau téléphonique, ADSL, LAN
- câble coaxial un fil au sein d'un gaine, isolés l'un de l'autre
  - blindage automatique
  - meilleurs débits
  - plus grandes distances
  - ✔ télévision, LAN
- fibre optique matériaux transparent gainé, signal lumineux
  - léger
  - pas de bruit, faible atténuation
  - large bande passante
  - complexe à connecter
  - ✔ MAN, WAN

Transmissions sans fil

- infrarouge lumière juste sous le spectre visible
  - faible portée, sensible aux bruits
  - lent
  - ✔ souris, PDA, télécommandes, ...
- micro-ondes terrestres ou satellites, onde radio directionnelle
  - parabole pour l'émission et la réception
  - ✔ télévision par satellite, GPS, radars, ...
- radio diffusion onde radio non directionnelle
  - antennes simples
  - ✔ radio, télévision terrestre, souris, PDA, Wifi, Bluetooth, ...

Ethernet

- sur paire torsadée ou câble coaxial
- standard majoritaire pour les LAN
- partage du support par CSMA/CD

Carrier Sense Multiple Access with Collision Detection

- avant d'émettre une trame : attendre le silence
- pendant l'émission : écouter s'il y a une perturbation (collision)
- en cas de collision : attente aléatoire avant ré-émission
- garantie de détection des collisions
  - ☞ *calibrage de la longueur des trames et des câbles selon la vélocité du signal sur le support choisi*

- pour les transmissions non guidées : CSMA/CA *Collision Avoidance*
  - détection de collision pas toujours possible — portées limitées
  - RTS → CTS → données → ACK *Request/Clear To Send, acknowledge*

D'autres protocoles sur transmissions guidées

Asynchronous Transfert Mode (ATM)

- multiplexage temporel synchrone
- cellules de 53 octets — 5 d'entête, 48 de données
- réservation du support au préalable
- haut débit pour les MAN et WAN
- bonne prise en charge de la qualité de service

(*Asymetric*) *Digital Suscriber Line* (ADSL/DSL)

- sur paire torsadée, point à point — adapté aux lignes téléphoniques
- multiplexage par division de spectre
- asymétrique ⇒ bande passage montante plus étroite
  - favorise la réception par rapport à l'émission

Protocoles sur transmissions non guidées

Bluetooth

- radio diffusion de faible portée (< 30m)
- faible consommation
- ✔ appareil portables — PDA, téléphones, souris, ...

Wifi

- radio diffusion à courte moyenne (< 100m)
- bon débit (54Mb/s) — adapté à l'utilisation d'Internet
- ✔ ordinateurs, *smartphones*, ...

- Wimax : radio diffusion à longue portée — 50km, 70Mb/s
- GSM (2G), GPRS (EDGE/2.5G), UMTS (3G) : pour la téléphonie mobile, adaptés aux communications informatiques

Matériels d'interconnexion de réseaux

- répéteur régénère un signal entre deux supports
  - aucune interprétation du contenu ⇒ couche 1
- amplificateur répéteur qui augmente l'amplitude
  - compense l'atténuation
- concentrateur (*hub*) répéteur multiport
  - pour la création de réseaux en étoile
- pont (*bridge*) segmentation des LAN
  - réseaux de même technologie
  - filtrage sur les adresses de niveau 2
  - réduit le trafic sur chaque segment
- commutateur (*switch*) réseau en étoile avec segmentation
  - utilité du concentrateur
  - caractéristiques du pont

Matériels d'interconnexion de réseaux

. . . suite

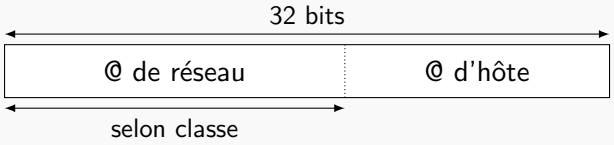
- passerelle (*gateway*) interconnexion des LAN
  - technologies différentes
  - existe à différents niveaux (≥ 3)
- routeur passerelle de niveau 3
  - acheminement à travers différents réseaux
- proxy serveur mandataire de niveau ≥ 4
  - réalise une requête à la place d'un client
- firewall routeur filtrant
  - tous ces matériels peuvent filtrer, chacun à son niveau

Le réseau des réseaux

- interconnexion de réseaux au niveau mondial
  - utilise le protocole IP (*Internet Protocol*)
- chaque participant possède une adresse IP
  - adresse sur 32 bits (4 octets)
  - 4 nombres décimaux séparés par des points ex : 192.168.1.5
- une adresse IP définit un hôte (*host*)
  - plusieurs adresses IP par machine ex : 192.168.1.5 et 127.0.0.1
  - une adresse IP peut cacher plusieurs machines voir NAT plus tard
- sur un même hôte, différents systèmes utilisent TCP ou UDP
  - chacun avec son propre numéro de port ex : port 80 ⇒ web
  - multiplexage de TCP et UDP sur IP
  - TCP et UDP sont des protocoles de la couche 4

Adresses IP

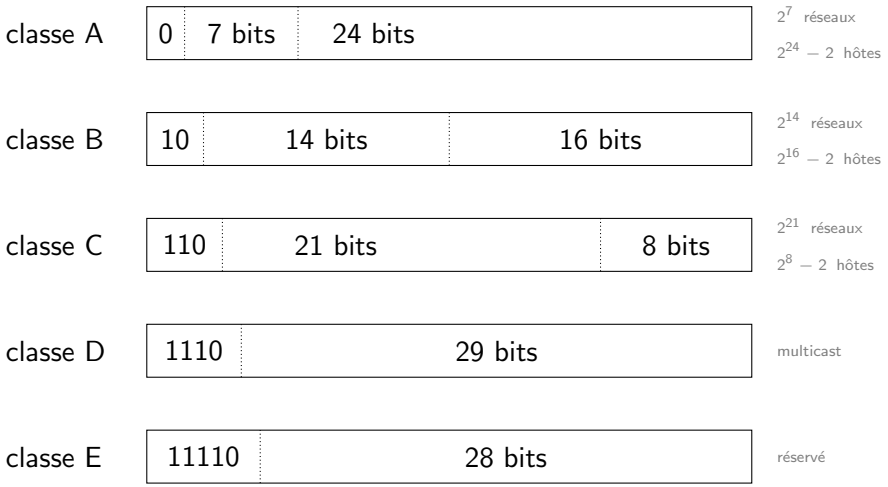
Deux sous-adresses se partageant 32 bits



- @ d'hôte $= 0 \cdots 0$ adresse IP d'un réseau
- @ d'hôte $= 1 \cdots 1$ adresse de diffusion du réseau broadcast

- adresses IP notées comme quatre octets en décimal
  ☞ *192.168.12.25*

Classes d'adresses



Classless Inter-Domain Routing (CIDR)

RFC 4632 (1992)

- rend obsolète et généralise la notion de classes
- découpage réseau/hôte à une position arbitraire

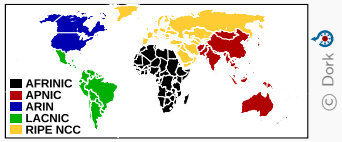
Découpage des adresses

- notation *a*/*n* adresse *a* avec *n* bits pour l'adresse de réseau
  - ☞ *10.10.1.32/27 ⇒ @ de réseau sur 27 bits*
    00001010.00001010.00000001.00100000
- attention à la notion décimale !
  - 10.10.1.32/27 00001010.00001010.00000001.00100000
  - ✔ 10.10.1.44 00001010.00001010.00000001.00101100
  - ✘ 10.10.1.90 00001010.00001010.00000001.01011010
- adresse de réseau ⇒ masque de sous-réseau

Adresses réservées

Internet Corporation for Assigned Names and Numbers (ICANN)

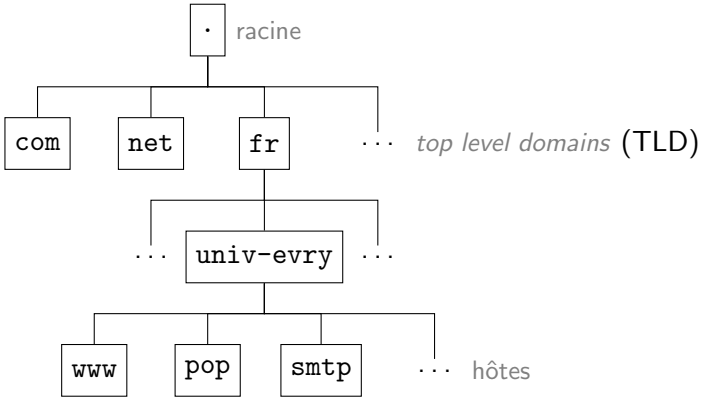
- gère l'attribution des adresses IP
- délègue par zone géographique aux registres Internet régionaux (RIR)



- 10.0.0.0/8 réseaux privés de classe A
- 172.16.0.0/12 réseaux privés de classe B
- 192.168.0.0/16 réseaux privés de classe C
- 127.0.0.0/8 adresses de rebouclage *loopback*
- 192.0.2.0/24 documentation, exemples, ...
- 192.88.99.0/24 relais IPv6 vers IPv4

Domain Name System (DNS)

☞ *système permettant d'associer des noms aux adresses IP*



- `www.univ-evry.fr.` est le FQDN *Fully Qualified Domain Name*
  - ☞ *au plus 127 niveaux de 63 caractères, au plus 255 caractères au total*

Serveurs de noms

- 13 serveurs racines prennent en charge les TLD
  - `a.root-servers.net`, ..., `m.root-servers.net`
  - gérés par l'ICANN
- chaque (sous-)domaine est géré par deux serveurs
  - enregistrés dans le serveur du niveau supérieur
  - serveur primaire + serveur secondaire
- chaque client interroge le serveur de son domaine
  - un parcours de l'arborescence permet la résolution
  - par le client (itératif) ou par les serveurs (récursif) voir plus tard
- le DNS est une base de donnée répartie
  - les enregistrements sont des *Resource Records* (RR)

Resource Records (RR)

- FQDN — terminé par . ⇒ complet, sinon : suffixe
- *Time To Live* (TTL) durée de mise en cache
- type de l'enregistrement
  - A — adresse — type le plus courant
  - CNAME — pour les alias — *Canonical Name*
  - MX — serveur de mail du domaine — *Mail eXchange*
  - NS — serveur DNS ayant autorité sur le domaine — *Name Serveur*
  - ...
- classe de l'information
  - IN — Internet — public
  - CH — chaotique — privé
- RDATA données associées au RR, selon son type :
  - A — adresse IP sur 32 bits
  - CNAME — nom de domaine
  - MX — priorité et nom d'hôte
  - NS — nom d'hôte
  - ...

Top Level Domains

- noms génériques :
  - `.arpa` — machine issues du réseau ARPANET
  - `.com` — commercial — fourre-tout
  - `.edu` — éducation — États-Unis
  - `.gov` — gouvernement — États-Unis
  - `.mil` — militaire — États-Unis
  - `.net` — Internet — fourre-tout
  - `.org` — non lucratif
- noms génériques depuis 2000 :
  - `.museum` — musées
  - `.name` — personnes ou personnages
  - `.aero` — aéronautique
- nationaux :
  - `.fr` — France
  - `.uk` — Royaume Uni
  - `.de` — Allemagne

Uniform Resource Identifier (URI)

URL (... *Locator*) et URN (... *Name*)

☞ *adresse d'une ressource sur Internet*

PROTO://LOGIN:PASSWD@HOST:PORT/PATH;PARAM?QUERY#FRAG

- PROTO — protocole à utiliser — `http, ftp, mailto, ...`
- LOGIN — pour s'identifier auprès de l'hôte
- PASSWD — pour s'authentifier — toujours avec LOGIN
- HOST — nom ou adresse IP de l'hôte
- PORT — numéro de port explicite — sinon, déduit du protocole
- PATH — chemin d'accès, peut être vide — / est le séparateur
- PARAMS — paramètres d'accès — non utilisé
- QUERY — requête pour la ressource — `arg=val&...`
- FRAG — position à l'intérieur de la ressource — `#section2`

Encodage des URI

- caractères réservés : espace, /, &, ?, #, @, ...
- encodage par %XY où XY est le code ASCII du caractères — hexadécimal
  - %26 ⇒ &
  - %20 ⇒ espace

Caractères ASCII imprimables

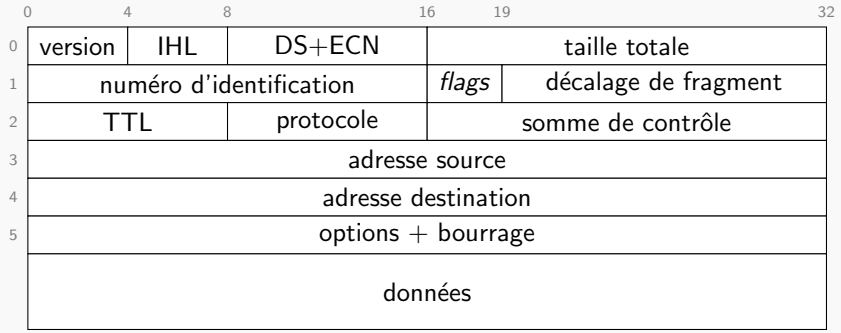
20	*sp*	21	!	22	"	23	#	24	$	25	%	26	&	27	’
28	(	29	)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[	5c	\	5d	]	5e	^	5f	_
60	‘	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c	|	7d	}	7e	~	7f	*nbsp*

Internet Protocol (IP)

RFC 791

- fait transiter des paquets entre deux hôtes (@IP)
- n'assure ni la livraison (pertes) ni l'intégrité (altérations)

Format des paquets IP



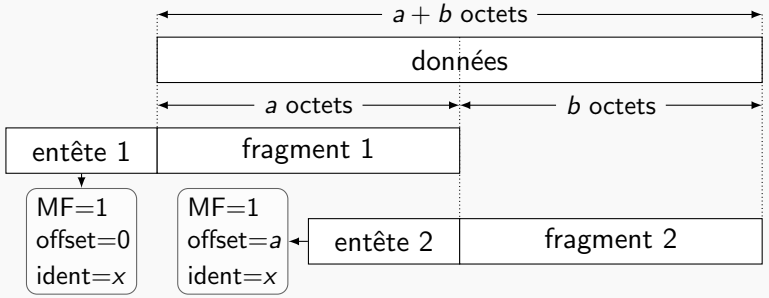
Entêtes IP

- version numéro de version : 4 (ici) ou 6 (plus tard)
- *internet header length* en mots de 32 bits
- *differential service + explicit congestion notification* pour la qualité de service
- taille totale en octets
- flags (drapeaux) pour la fragmentation :
  1. inutilisé
  2. DF : *don't fragment*
  3. MF : *more fragments*
- décalage de fragment (*fragment offset*) en octets
- time to live (durée de vie) décrémenté à chaque routeur
- protocole numéro du protocole transporté par la paquet
  - ☞ *ICMP=1, TCP=6, UDP=17, ...*
- somme de contrôle recalculé à chaque routeur

Fragmentation

- un paquet IP peut occuper 65536 octets au plus
- un routeur peut décider de fragmenter un paquet — si DF=0
  - MTU *maximum transmission unit*
  - ☞ *Ethernet : MTU=1500 octets ; Wifi : MTU=2272 octets*

Découpage des données

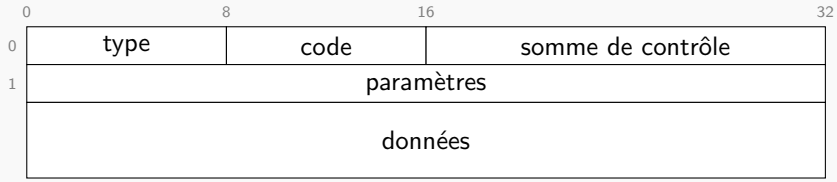


Internet Control Message Protocol (ICMP)

RFC 792

- message des contrôle (pas de données) :
  - ☞ *signalement des erreurs, test de connectivité, ...*
- paquets ICMP encapsulés dans des paquets IP — couche 3 quand même

Format des paquets ICMP



Entêtes ICMP

- type nature du message
  - 3 destinataire non accessible
  - 11 durée de vie écoulée — TTL arrivé à zéro
  - 5 redirection
  - 8 *echo request* — ping
  - 0 *echo reply* — pong
- code détails sur le type
  - 3.0 réseaux inaccessible
  - 3.1 hôte inaccessible
  - 3.2 protocole non disponible
  - 3.4 fragmentation nécessaire mais interdite — DF=1
- paramètres information supplémentaires suivant le type
  - ☞ *echo* ⇒ *16 bits de n° d'identification + 16 bits de n° de séquence*
- données selon le type
  - echo données arbitraires et optionnelles
  - 3/11 entête IP + 64 octets de données du paquet détruit

Address Resolution Protocol (ARP)

RFC826

- ARP adresse logique (IP) ↦ adresse physique
- RARP adresse physique ↦ adresse logique
  - *Reverse Address Resolution Protocol* (RFC903)
  - beaucoup moins utilisé

Adresses MAC (*Medium Access Control*)

- adresse physique utilisée sur Ethernet, Wifi, Bluetooth, ATM, ...
- adresse sur 6 octets
  - 3 pour le constructeur de la carte réseau
  - 3 pour le numéro de série
    - ☞ `00:15:C5:1A:2B:3C` ⇒ *Dell*
      `00:18:DE:4D:5E:6F` ⇒ *Intel*
  - notées comme 6 nombres hexadécimaux séparés par ":"
- doit être unique sur un même réseau physique — re-programmables

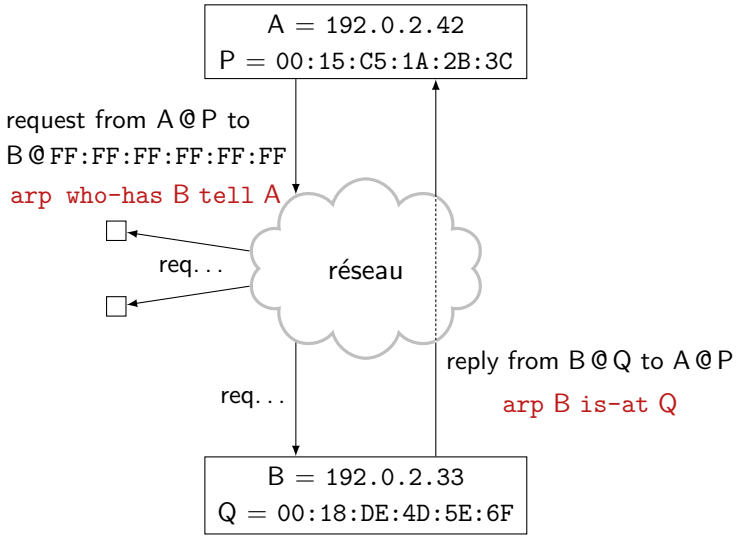
Paquets ARP

Format des paquets ARP et RARP

	0	16	32
0	type du réseau physique		type de l'adresse demandée
1	taille @ physique	taille @ logique	opération
	adresse physique de l'émetteur (taille variable)		
	adresse logique de l'émetteur (taille variable)		
	adresse physique du destinataire (taille variable)		
	adresse logique du destinataire (taille variable)		

- type de réseau 1 ⇒ Ethernet
- type d'adresse 0x800 ⇒ IP
- tailles d'adresses en octets — MAC ⇒ 6, IP ⇒ 4
- opération 1 ⇒ *request*, 2 ⇒ *reply*

Déroulement du protocole ARP



Le routage

Objectifs

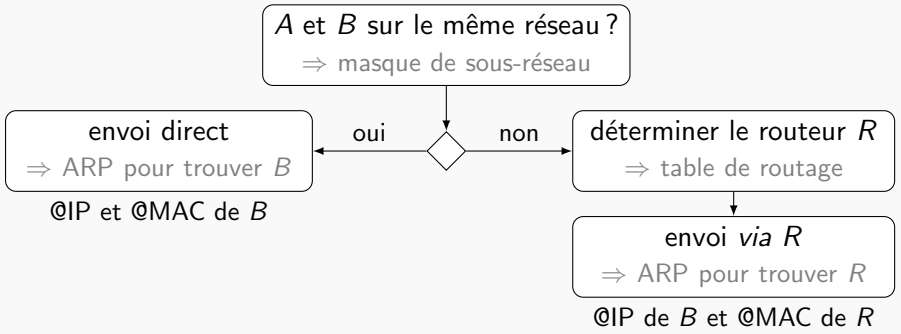
- interconnecter des réseaux
- organiser de grands réseaux en les divisant
- minimiser les coûts de transmission — temps, énergie, prix, ...

Enjeux

- chaque station doit savoir quand et comment s'adresser à un routeur
- chaque routeur doit savoir :
  - déterminer un chemin — acheminement
  - quand et comment délivrer l'information aux stations
- chaque système connecté à routeur doit avoir une adresse discriminante — hôte ou réseau

Point de vue des stations

Envoi d'un paquet IP : $A \rightarrow B$



- table de routage : @ réseau $\mapsto$ @ routeur, interface
  - une entrée est le routeur par défaut
- l'adresse IP du routeur n'est utilisée que pour ARP

Point de vue des routeurs

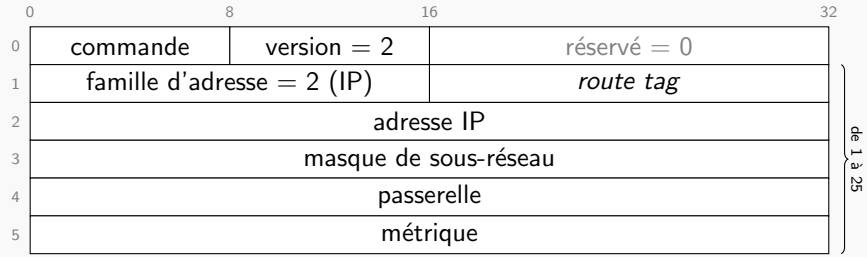
- utilisation des tables de routage
  - en tant que stations
  - pour transmettre les paquets des stations
  - enrichies ⇒ métriques et horodatages
- utilisation d'un protocole de routage
  - échange d'information entre les routeurs
  - mise à jour des tables de routage
- détermination des meilleurs chemins
  - minimisation du nombre de sauts
  - minimisation des coûts
  - maximisation de la vitesse — latence et débit
  - gestion des congestions et des pannes
  - minimisation du trafic dû au protocole de routage
  - évitement des boucles et des erreurs de routage
- connaissance globale ou locale du réseau

Routing Information Protocol (RIP)

version 2, RFC 2453

- métriques en nombre de sauts
- 15 sauts maximum
- convergence lente — cycles de 30 s ⇒ 7 min
- communications sur UDP entre voisins — multicast 224.0.0.9

Format des datagrammes RIP



Routing Information Protocol (RIP)

suite

- commande demande (1) ou diffusion (2)
- *route tag* routes internes (RIP) *vs* externes (autre protocole)
  - externes ⇒ invalidées après 6 cycles, puis supprimées après 2 cycles
- adresse, masque, . . . ⇒ une ligne de la table de routage

Fonctionnement

- démarrage ⇒ demande à tous les voisins famille=0, métrique=16
  - voisins connus par la configuration
- chaque 30 s ⇒ diffusion de toute ou partie de la table
- changement de métrique ⇒ diffusion aux voisin *triggered updates*
- réception d'une diffusion ⇒ mise à jour de la table
- permanence des meilleures routes jusqu'à nouvel ordre
- optimisations : *split horizon* ou *poisoned reverse*

IP version 6

Moderniser IPv4

- augmenter l'espace d'adressage
  - 32 bits ↦ 128 bits
    - ☞ *667 millions de milliards d'adresses par mm^2 sur Terre*
  - chaque hôte peut posséder une adresse pour chaque usage
- simplifier le routage
  - 64 premiers octets ⇒ adresses de réseaux, organisées hiérarchiquement
  - protocole simplifié ⇒ ↗ performances des routeurs
- améliorer la sécurité
  - intégration des extension IP-sec optionnelles en IPv4
- faciliter les extensions futures
  - entêtes d'options multiples liste chaînée, classée
- mécanismes de compatibilité et de transition
  - nouvelles versions des protocoles de niveau 3
  - mise à jour des protocoles utilisant des adresses IP
  - compatibilité du champ de version
  - encapsulation des adresses IPv4 dans une classe réservée
  - mécanisme IPv6 *over* IPv4

Outils pour la couche réseau

- ping requêtes ICMP *echo request*
  - ☞ *ping -t15 -c 2 www.google.com*
- arp consultation et manipulation de la table ARP
- ifconfig configuration des interfaces réseau
  - ☞ *ifconfig eth0*
  - ☞ *ifconfig eth0 192.16.8.0.1*
- route affichage et manipulation de la table de routage
- traceroute recherche des routeurs intermédiaires utilise le TTL
- tcpdump capture de trames *sniffer*
- wireshark analyseur de trame *sniffer* + disséqueur

Deux modes de transports

User Datagram Protocol (UDP)

- mode déconnecté ⇒ orienté paquet
- pas de contrôle de la livraison

Transmission Control Protocol (TCP)

- mode connecté ⇒ orienté flux bidirectionnel
- contrôle de la livraison transmission fiabilisée
- contrôle du flux

Dans les deux cas :

- communication point-à-point entre applications
- mode client/serveur

Ports réseaux et *sockets*

- adresse IP ⇒ identifie un hôte
- numéro de port ⇒ identifie une application sur un hôte
- socket = @IP + n°port
  - ☞ *195.221.162.126 :80 ⇒ serveur web sur www.ibisc.univ-evry.fr*

Clients et serveurs

- serveur — attend les connexions (écoute) sur un port défini
- client — initie la connexion depuis un port quelconque

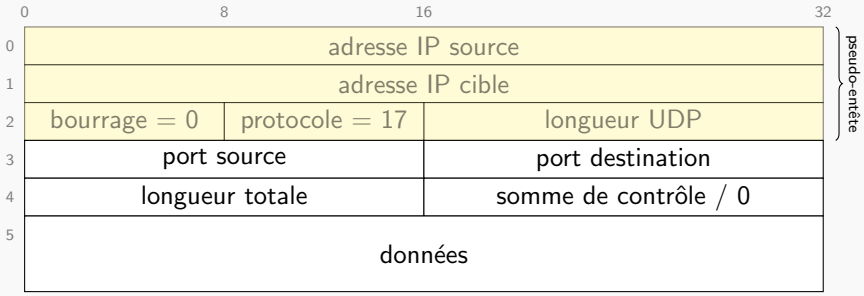
Assignation des ports

- 0 → 1023 — ports reconnus, réservés et privilégiés
  - ☞ *80 ⇒ HTTP, 110 ⇒ POP, 22 ⇒ SSH, 52 ⇒ DNS, ...*
- 1024 → 49151 — ports enregistrés — applications particulières
- 49152 → 65535 — ports privés — utilisables à discrétion

User Datagram Protocol (UDP)

RFC 768

Format des datagrammes UDP

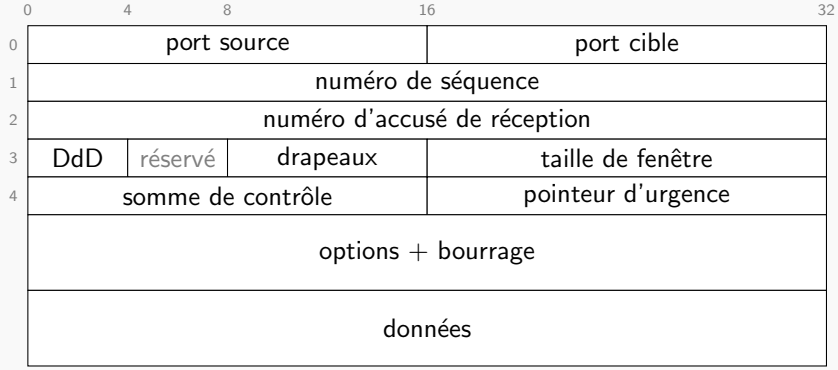


- pseudo-entête pour le calcul de la somme de contrôle — 0 sinon
- permet d'éviter les erreurs de routage

Transmission Control Protocol (TCP)

RFC 793

Format des segments TCP



Entêtes TCP

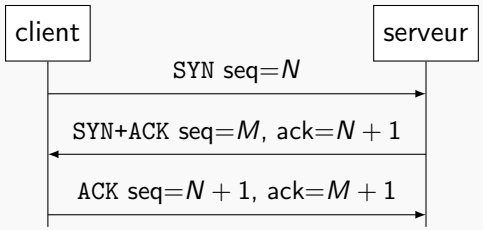
- numéro de séquence pour la remise en ordre à la réception
- accusé de réception numéro du prochain segment attendu
- décalage de données taille de l'entête — / 32 bits
- drapeaux
  - CWR *congestion window reduced*
  - ECE *ECN-Echo* — congestions
  - URG pointeur d'urgence renseigné
  - ACK accusé de réception — obligatoire après la connexion
  - PSH *push* : envoi des données à l'application
  - RST *reset* : réinitialisation de la connexion
  - SYN synchronisation lors de la connexion
  - FIN fin des données et début de la déconnexion
- fenêtre nombre max d'octets attendus à partir du segment accusé
- urgence nombre d'octets urgents restant à transmettre

Fiabilisation de la transmission

- sommes de contrôle
- numérotation des segments
- connexion et déconnexion explicite
- accusés de réception
  - maintient de la connexion
  - accusé de réception du segment $n \Rightarrow$ segments $< n$ reçus
    ☞ $\searrow$ *nombre d'accusés de réception*
  - envois avant acquittement dans la limite de la fenêtre
    ☞ *fenêtre glissante ajustable*
  - délai maximal pré-programmé pour les accusés de réception

Poignée de mains à 3 temps

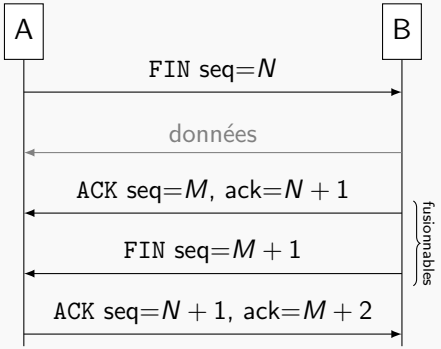
Connexion



- le premier numéro N est choisi aléatoirement
  ☞ *évite l'interposition d'un intrus : connection hijacking*
- jusqu'à la déconnexion : `ACK` dans tous les paquets

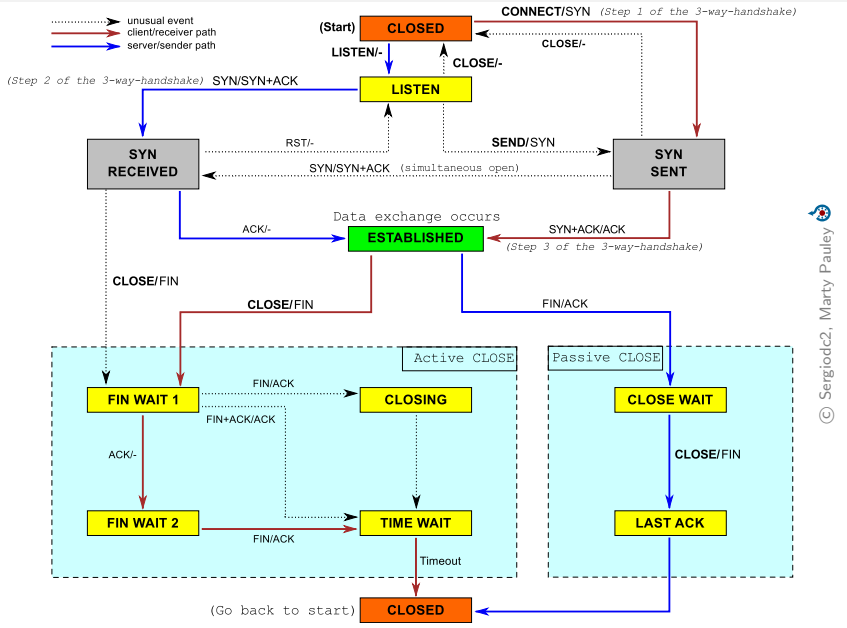
Poignée de mains à 4 (ou 3) temps

Déconnexion

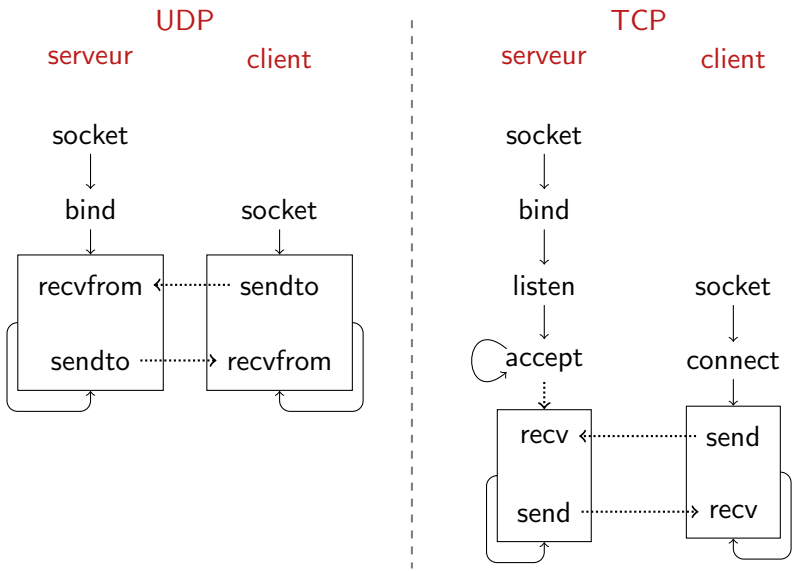


- la fusion `ACK+FIN` implique d'autres numéros de séquences

Diagramme des états TCP



Deux protocoles et deux modes



Client et serveur UDP

Serveur : socket → bind → (recvfrom | sendto)*

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(("", 1234))
data, address = sock.recvfrom(1024)
print "%s:%u says" % address, repr(data)
sock.sendto(data, address)
```

Client : socket → (recvfrom | sendto)*

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
data = raw_input("? ")
sock.sendto(data, ("127.0.0.1", 1234))
data, address = sock.recvfrom(1024)
print "%s:%u says" % address, repr(data)
```


Client et serveur TCP

Serveur : socket → bind → listen → accept → (recv | send)* → close

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(("", 1234))
sock.listen(1)
connection, address = sock.accept()
print "%s:%u connected" % address
data = connection.recv(1024)
print "client says:", repr(data)
connection.send(data)
connection.close()
```


Client et serveur TCP

. . . suite

Client : socket → connect → (recv | send)* → close

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(("127.0.0.1", 1234))
data = raw_input("? ")
sock.send(data)
data = sock.recv(1024)
print "server says:", repr(data)
sock.close()
```

Network Address Translation (NAT)

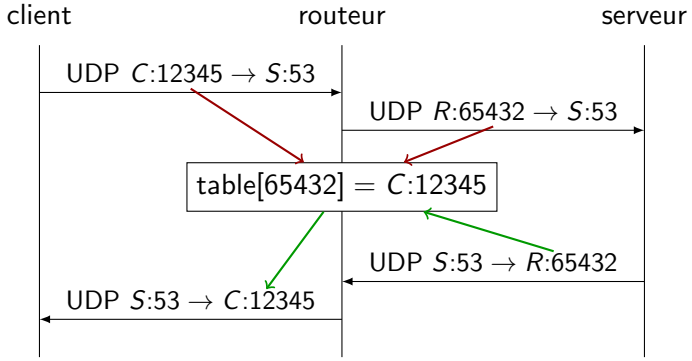
RCF 3022

- compense le manque d'adresses IPv4
- un réseau privé communique sur Internet via l'adresse du routeur
- pour cacher des machines ou répartir la charge
- utilisation des numéros de ports pour la cohérence
  - ☞ *routage de niveau 3 exploitant des informations de niveau 4*
- traçage des connections depuis le réseau interne

Port forwarding

- assignation statique d'un port à une adresse interne
- accès direct à un serveur

Exemple : NAT sur une requête DNS

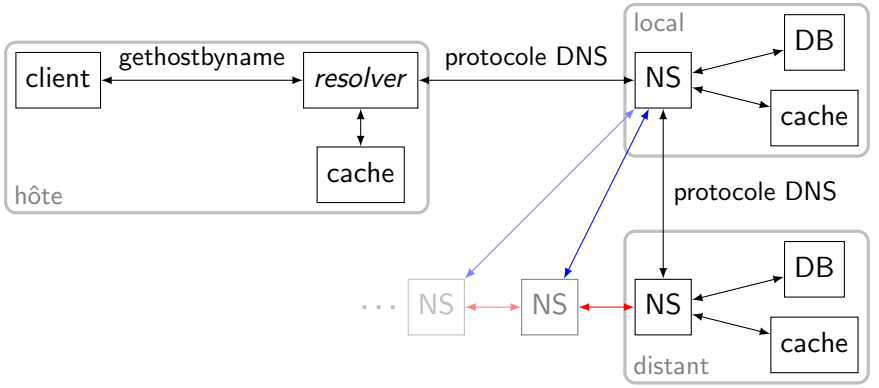


Outils pour la couche transport

- netstat liste les sockets ouverts et leur état
  - ☞ *netstat -t -u*
- netcat (nc) envoi/réception de données sur TCP ou UDP
  - ☞ *nc www.google.com 80*
- wireshark décortiquer des datagrammes, reconstituer des flux
- votre langage de script préféré et l'API socket

Domain Name System (DNS)

RFC 1034 et 1035



- **récursif** : transmission des requêtes à un autre NS
  - ☞ *mode par défaut d'un NS local*
- **itératif** : redirection vers un autre NS

Format des messages DNS

sur UDP (message courts) ou TCP (messages longs)

- identificateur (16 bits) pour lier la requête à la réponse — pour UDP
- requête/réponse (1 bit) type de message
- opération (4 bits)
  - standard — résolution de nom
  - inverse — résolution d'adresse
  - statut — état du serveur
- réponse d'autorité (1 bit) si le NS fait autorité sur le domaine résolu
- tronqué (1 bit) si la réponse est trop longue pour UDP
  - ☞ *recommencer en utilisant TCP*
- récursion demandée (1 bit) dans la requête
- récursion disponible (1 bit) dans la réponse
- réservé (3 bits)

Format des messages DNS

. . . suite (1/2)

- code de réponse (4 bits)
  - pas d'erreur
  - requête mal formée
  - erreur du serveur
  - domaine inexistant
  - requête non supportée
  - requête refusée
- nombre de requêtes (16 bits)
- nombre de RR dans la réponse (16 bits)
- nombre de RR dans la redirection (16 bits) pour le mode itératif
- nombre de RR dans les commentaires (16 bits) informations complémentaires disponibles

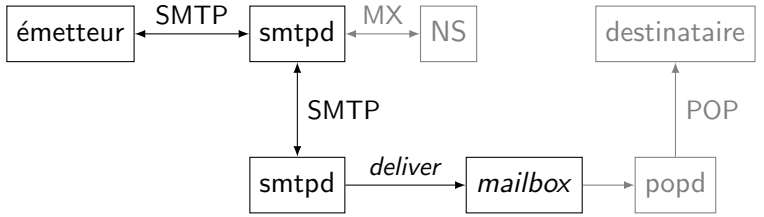
Format des messages DNS

. . . suite (1/2)

- section de requêtes chaque entrée contient :
  - nom de domaine — codage : (taille,nom)*,\0
  - type de requête — type de RR : A, CNAME, . . .
  - classe de requête — IN ou CH
- section de réponses : liste de RR
- section d'autorité : liste de RR de redirections
- section supplémentaire : liste de RR liés à la requête mais n'y répondant pas directement

Simple Mail Transfer Protocol (SMTP)

RFC 821 (protocole) et 822 (codages)



Dialogue sur TCP

- $C \rightarrow S$ commande (paramètre)*\r\n
- $S \rightarrow C$ statut (commentaire) ?\r\n

Principales commandes SMTP

- `HELO nom`
  identification du client (hôte) — obligatoire à la connexion
- `MAIL FROM: <email>`
  expéditeur — aucune vérification
- `RCPT TO: <email>`
  destinataire — plusieurs possibles, peuvent être refusés
- `DATA`
  début du message, terminaison par `.\r\n` (seul sur la ligne)
- `QUIT`
  fin de la session

Principales réponses SMTP

2xx ⇒ OK, 3xx ⇒ suite attendue, 4xx ⇒ erreur temporaire, 5xx ⇒ erreur permanente

- 220 invite lors de la connexion ⇒ `HELLO` attendu
- 221 réponse à `QUIT`
- 250 message délivré localement
- 251 message délivré à un relais
- 354 début des données
- 452 espace disque saturé
- 500 commande non reconnue
- 502 commande non prise en charge
- 503 commande mal séquencée
- 550 destinataire inconnu

Format des données SMTP

RCF 822

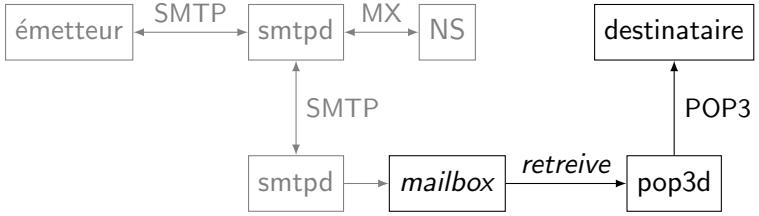
- `entête: valeur\r\n` — autant de fois que nécessaire
  ☞ `Date`, `From`, `To`, `CC`, `Subject`, ...
  sans aucun lien avec ce qui a été échangé dans les commandes SMTP
- `\r\n`
- texte sans ligne valant `.\r\n`
- `.\r\n` — comme spécifié précédemment

Multipurpose Internet Mail Extensions (MIME) — RFC 2045 à 2049

- format de messages étendu pour gérer :
  - les encodages — accents, alphabets non latins, données binaires
  - les message multi-parties — avec pièces attachées
  - les message multi-formats

Post Office Protocol v.3 (POP3)

RFC 1939



Dialogue sur TCP

- $C \rightarrow S$ commande (paramètre)*`\r\n`
- $S \rightarrow C$ `+OK` (commentaire) ?`\r\n`
- $S \rightarrow C$ `-ERR` (commentaire) ?`\r\n`

Principales commandes POP3

et les réponses associées

- `USER login`
  `PASS passwd` — authentification, en clair sur le réseau
- STAT — statistiques sur la boîte
  - `+OK nombre_de_messages taille_totale`
- LIST — liste les messages de la boîte
  - `+OK\r\n`
    `identificateur_1 taille_1\r\n`
    ...
    `.\r\n`
- RETR identificateur — récupération d'un message, terminé par `.\r\n`
- DELE identificateur — effacement d'un message
- TOP identificateur nombre — récupération du début d'un message
- QUIT — fin de la session

Limites de POP3

- une seule boîte, un seul client à la fois
- uniquement lecture et effacement des messages
- remplacé progressivement par IMAP

Internet Message Access Protocol (IMAP) — RFC 3501

- gestion de boîtes multiples en lecture/écriture/effacement
- gestion des états de messages
- recherche de messages sur le serveur
- multiples connexions simultanées — de longue durée
- ...

HyperText Transfer Protocol (HTTP)

RFC 2616 (v1.1)

Messages sur TCP

- requête ou statut — (1 ligne)
  - `méthode URL version\r\n` — pour une requête
  - `version code explication\r\n` — pour une réponse
- entêtes générales
  - ☞ `Date`, `Forwarded`, `Keep-Alive`, `MIME-version`, ...
- entêtes de requête ou de réponse
  - ☞ *pour les requêtes :* `Accept`, `Accept-Language`, `Authorization`, `If-Modified-Since`, `Range`, `Referrer`, `User-Agent`, ...
    pour les réponses : `Location`, `Retry-After`, `Server`, ...
- entêtes concernant le corps
  - ☞ `Content-Encoding`, `Content-Length`, `Content-Language`, `Content-Type`, `Expires`, `Last-Modified`, ...
- corps + `\r\n` — une ligne vide à la fin

Principales méthodes HTTP

- GET récupération d'une ressource, via le corps de la réponse
- HEAD comme `GET` mais seules les entêtes sont envoyées
- OPTIONS demande d'information sur une ressource
- POST comme `GET` mais des paramètres sont passés dans le corps de la requête
- PUT stockage d'un fichier
- PATCH modification d'un fichier
- COPY copie d'un fichier (à l'entête `URL-Header`)
- MOVE déplacement d'un fichier
- DELETE suppression d'un fichier

Principaux codes de retour HTTP

- 10x information
- 20x réussite
  - 200 OK
  - 204 pas de réponse à donner
- 30x redirection
  - 301 ressource déplacée de façon permanent
  - 302 ressource déplacée
  - 304 aucun changement avec `If-Modified-Since`
- 40x erreur du client
  - 400 mauvais requête
  - 401 non autorisé, il faut s'authentifier
  - 404 ressource non trouvée
- 50x erreur du serveur
  - 500 erreur interne
  - 501 méthode non supportée
  - 503 service non disponible

Exemple de requête HTTP

capturé par wireshark

```
GET / HTTP/1.1\r\n
Host: www.google.com\r\n
Connection: keep-alive\r\n
Accept: application/xml,text/html,text/plain,image/png,*/*\r\n
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US) AppleWebKit/534
Accept-Encoding: gzip,deflate,sdch\r\n
Accept-Language: en-GB,en-US,fr-FR\r\n
Accept-Charset: ISO-8859-1,utf-8,*\r\n
Cookie: rememberme=true; PREF=ID=a4d6dbe649bca4f1:U=41c3b238da53d532
\r\n
HTTP/1.1 302 Found\r\n
...
```


Exemple de requête HTTP

... suite

```
...
\r\n
HTTP/1.1 302 Found\r\n
Location: http://www.google.fr/\r\n
Cache-Control: private\r\n
Content-Type: text/html; charset=UTF-8\r\n
Date: Fri, 29 Oct 2010 12:19:03 GMT\r\n
Server: gws\r\n
Content-Length: 218\r\n
\r\n
<HTML><HEAD><meta http-equiv="content-type" content="text/html;chars
<TITLE>302 Moved</TITLE></HEAD><BODY>\n
<H1>302 Moved</H1>\n
The document has moved\n
<A HREF="http://www.google.fr/">here</A>.\r\n
</BODY></HTML>\r\n
```


Outils Unix pour les protocoles de haut niveau

- host requêtes DNS
  - ☞ *host -t MX univ-evry.com* — *type de RR*
    host -a google.com — *tous les RR*
    host www.google.com — *-t A*
- wget client HTTP et FTP
  - ☞ *wget http://www.univ-evry.fr*
    wget -m http://www.ibisc.univ-every.fr/~fpommereau/reseau-L3/
- mailx client SMTP
- getmail client IMAP et POP — alternative à fetchmail
- netcat (nc) client généraliste — il faut connaître le protocole
- votre langage de script préféré et sa bibliothèque standard