Internet : Architecture, adressage, IP

Session de Remise à Niveau, Télécom Paris

Août-septembre 2021

Ken CHEN

ken.chen@univ-paris13.fr

Plan

+ Architecture
+ Adressage
+ ARP
+ IP
+ ICMP

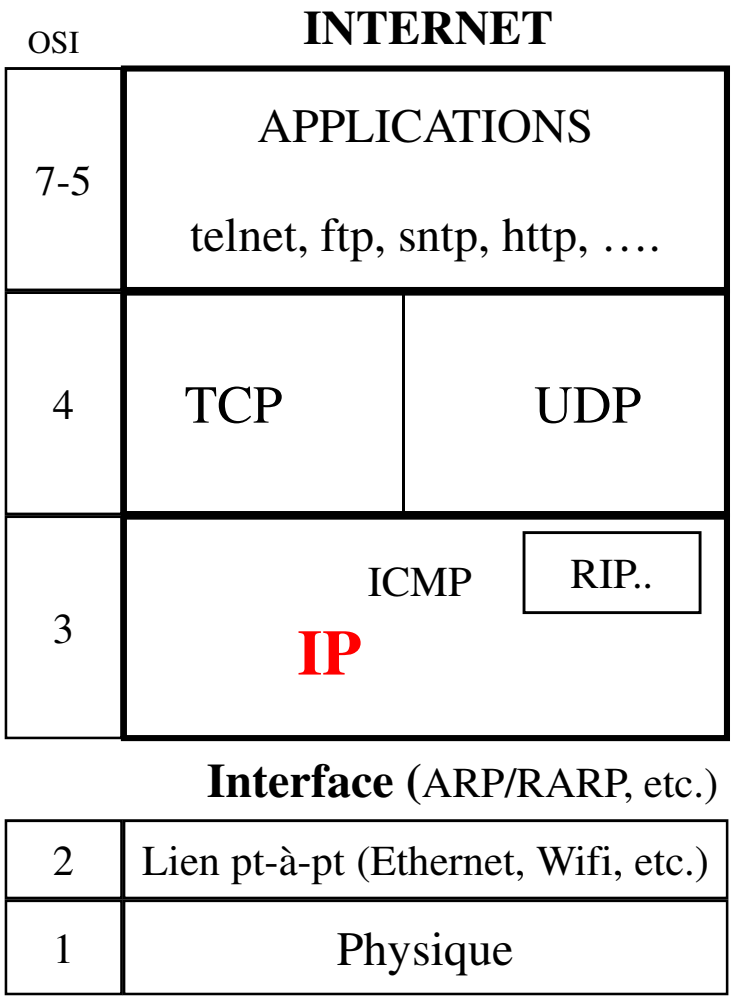
Contexte : Internet

+ Une **architecture** et un **empilement de protocoles** (*profil*) pour construire des réseaux étendues
+ L'infrastructure WAN la plus étendue
+ Solution et architecture **ouverte**

+ Base technologique fondée en début 1970
  - Expérience ARPANET débuté fin 1969
+ Evolution permanente (IETF)

+ Format de base : **IP** (Internet Protocol)
  - Tous les paquets Internet sont codés en IP

Modèle Architecture (1/3)



Modèle Architecture (2/3)

+ Internet
  - Infrastructure définie **à partir de la couche 3** (OSI)
  - Une pile de protocoles (***protocol suite***) entièrement **logicielle**
+ **Indépendance** vis-à-vis des moyens physiques de transfert de données (**liaisons**)
  - Tout type de lien peut être candidat
    + Ethernet, 802.11, Fibre optique, ADSL, etc.
  - Toute machine peut s'insérer dans Internet avec
    + la pile de protocoles Internet et
    + un lien avec une station déjà existante dans Internet
+ Un **seul** protocole couche 3 : **IP**
+ Un seul service couche 3 : **Datagram** (Best-Effort)

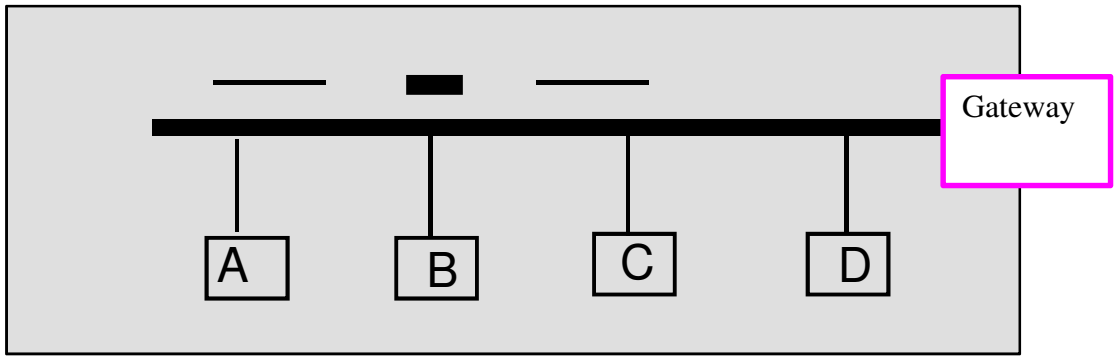
Modèle Architecture (3/3)

+ Deux protocoles de la couche 4 : TCP, UDP
  - TCP : service orienté connexion : **transfert fiable de bout-en-bout**
  - UDP : service ***datagram*** (même qualité que IP, mais au niveau 4)
  - TCP et UDP: même rôle d'interfaçage avec la couche appli
+ Couche application
  - HTTP, HTTPS, SMTP, etc.
  - Pas de couches 5 et 6 explicites
+ Aussi, d'autres protocoles utilitaires, par exemple
  - RIP, OSPF, BGP pour le routage
  - SSH, SSL, TLS pour la sécurité
  - DSN pour le serveur de nom, DHCP pour configuration de host

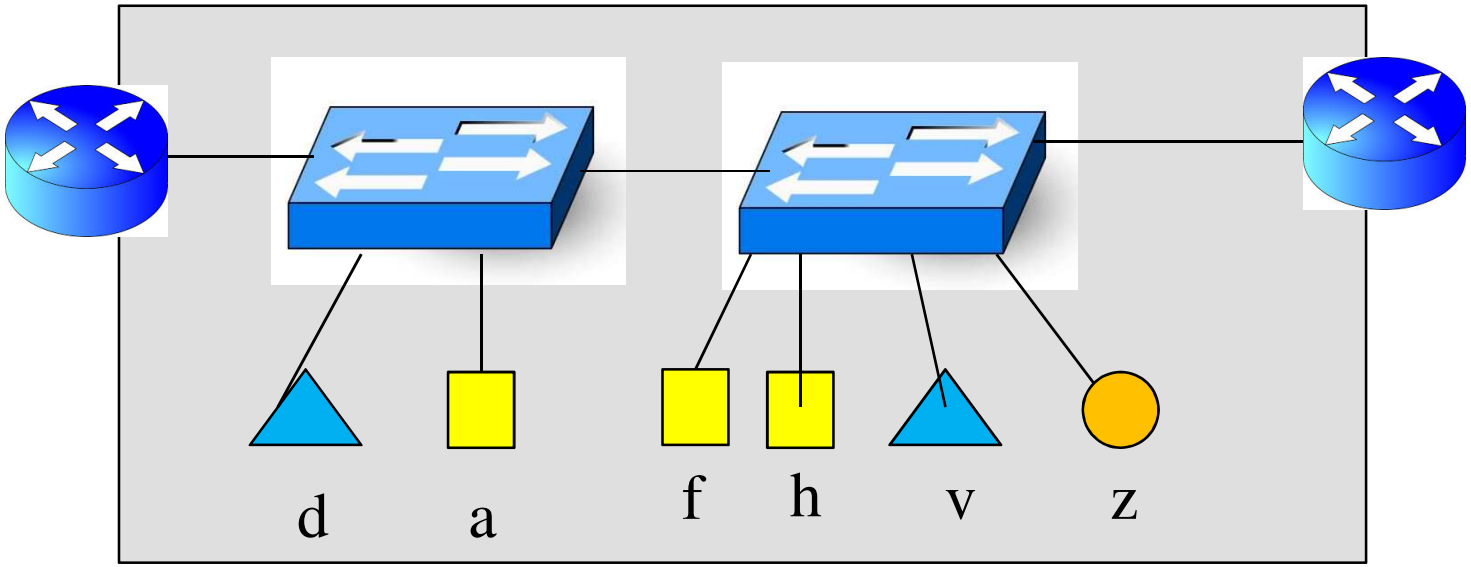
Composition

+ Internet = réunion des sites internet (IP)
+ Un site IP
  - Au moins une **passerelle** (***gateway***)
  - Possibilités de passerelles multiples
  - Un certain nombre de machine **hôte** (***host***)
  - Liens entre passerelle et hôtes : libre
    + Ethernet, Wifi, etc.
+ Interconnexion entre sites
  - Liens entre passerelles
+ Rôle de passerelle : routage
  - Passerelle == **routeur** (***router***)

Site

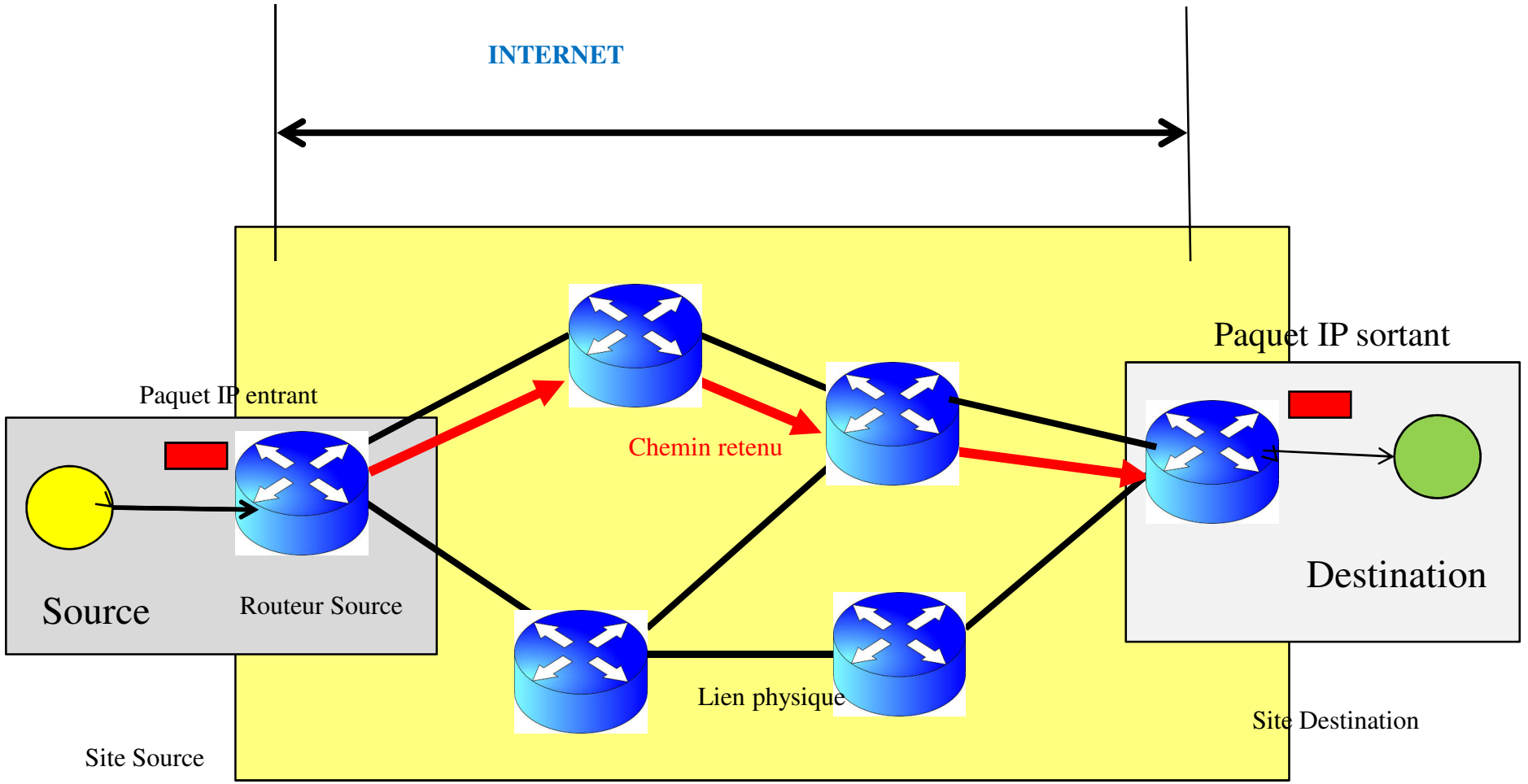


- Schéma « classique » d'un site avec un seul gateway



•Schéma d'un site avec deux gateways et avec les symboles usuels (Switch, router)

Vision réduite aux routeurs



Routage (L3)

Adresse IP

Format

Classes

CIDR

Adressage IP

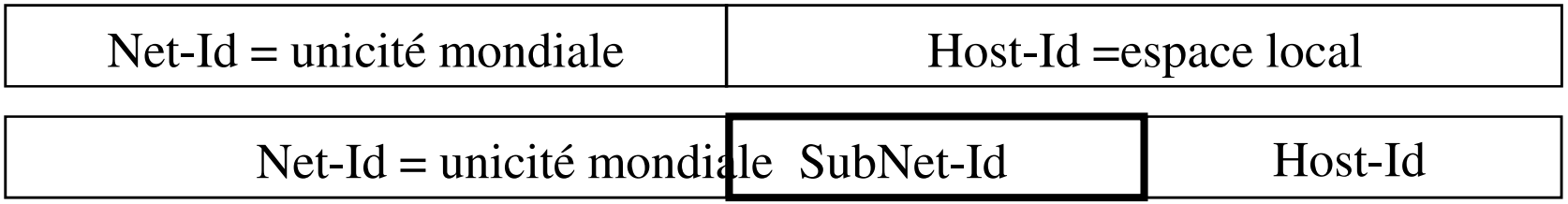
+ Format :
  - 4 octets (32 bits) à notation décimal pointé A.B.C.D.
  - Exemple : 137.194.52.18
+ Unicité à travers le Monde
+ Donne l'identité du site auquel elle appartient
  - Exemple **137.194**.x.y désigne une adresse de Télécom Paris
  - L'information de localisation facilite le routage
+ Associé à chaque **Interface** réseau (lien physique)
+ Conséquence :
  - un routeur (plusieurs liens par définition) possède autant d'adresses IP que d'interfaces

Structure (1/3)

+ Principe : découpage en deux parties
  - Une première partie d'identification globale (**Net-Id**)
  - Une seconde partie d'identification locale (**Host-Id**)
  - **Structure : Net-Id (global) + Host-Id (locale)**
+ Exemple
  - Dans **137.194**.52.18 :
    + 137.194 désigne le réseau ENST (Télécom Paris),
    + 52.18 désigne une station au sein de ce réseau
+ Conséquence pour le routage
  - Tous les paquets ayant comme destination **137.194**.x.y vont dans la même direction
  - Une **seule entrée** dans la table de routage pour toutes les **137.194**.x.y
    + À comparer avec la gestion des adresses MAC (adresses « ***plates*** ») dans des commutateurs

Structure (2/3)

+ Application itérative du principe de découpage :
  - Au sein d'un **réseau donné**, découpage de l'espace « identification locale »
  - Structure : Id. Réseau + [Id. Sous-réseau + Id. Hôte]
  - Exemple :
    + le réseau 137.194… possède un espace local sur deux octets (64 K !)
    + Division, ***hypothétique***, en 256 sous-réseaux (3e octet)
    + Adresse 137.194.52.18 = Hôte #18 du sous-réseau # 52

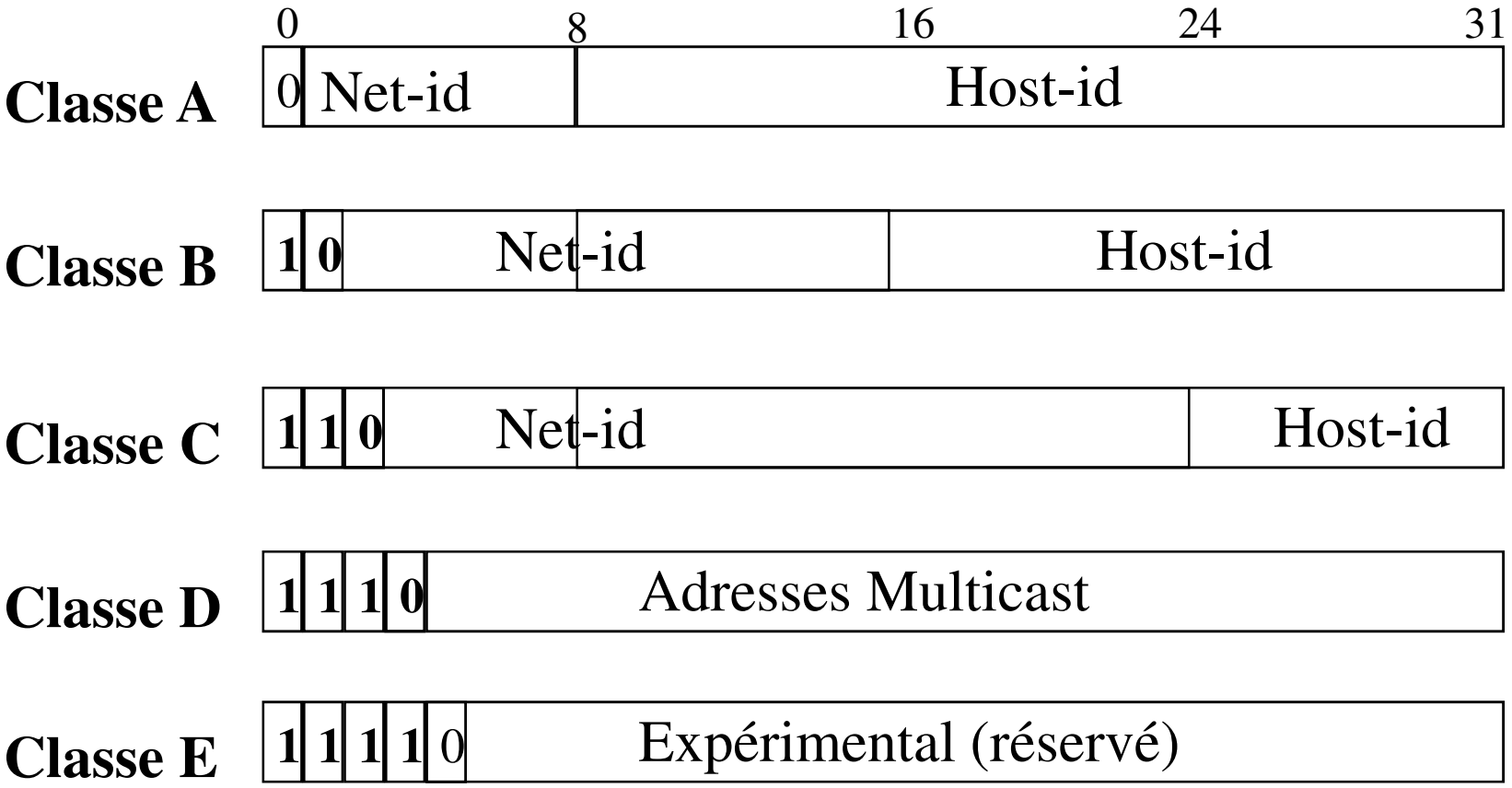


Structure (3/3)

+ Intérêt :
  - Faciliter l'organisation (découpage administratif)
  - Facilite le routage
+ Principe applicable itérativement
+ Précision : espace des identificateur = contigu
  - Interdiction d'avoir Subnet-Id entouré de Host-Id ou vis versa

+ Reconnaissance de Net-Id/Subnet-id :
  - utilisation de **Masque** (cf plus loin)

Les classes (1/3)



Les classes (2/3)

+ Définition initiale :
  - 3 classes d'adresses individuelles
  - 1 classe d'adresses multicast
  - 1 classe (espace) réservé
+ Classe A :
  - 126 réseaux au total (127.x.y.z = usage spécial)
  - De 1.0.0.0 à 126.0.0.0
  - Exemples : 17.0.0.0 (Apple), 18.0.0.0 (MIT)
  - TRES grands réseaux (16 M stations /réseaux)

Les classes (3/3)

+ Classe B :
  - 16382 (2^{14}) réseau
  - De 128.0.0.0 à 191.254.0.0
  - Exemple : 137.194.0.0. (ENST)
  - Réseaux de taille « raisonnable » : 64 K stations/réseaux
+ Classe C :
  - Beaucoup de réseux (2 Millons, 2^{21} - 2 = 2097150)
  - De 192.0.0.0.0 à 223.255.254.0
  - Réseau de taille faible : max. 254 stations
+ Classe D : multicast
+ Classe E : réservé

Adresse multicast

+ De 224.0.0.1 à 239.255.255.255
+ Utilisé par les applications multicast
  - Multicast = l'envoi d'un seul paquet pour destinataires multiples
+ Ne respecte pas la structure **Net-Id+Host-Id**
  - Une adresse Multicast = un label regroupant une liste d'adresses IP individuelles
  - Cf. RFC 1700 (10/94) pour une liste des adresses
+ Exemples
  - « application » Routage RIP v2
  - Adresse multicast : 224.0.0.9

Fin des classes

+ Phénomène
  - Classe B : classe d'adresses préférée
  - Classe C : trop petite
+ Remède : Remise en question des classes
  - Remise en question des C uniquement
  - Compatibilité
+ Procédé :
  - « casser » la barrière sur un octet entier
  - Champ Host-Id = 9, 10, 11 … bits
  - Précision : Net-Id = bits contigu
+ Réseau « C » de tailles raisonnable grandes possible

CIDR

+ Classless Internet Domain Routing (RFC 1338)
+ Abolition des classes (classe C surtout)
  - Résoudre le problème de « pénurie »
+ Notation pour la longueur de le Net-Id
  - Exemple : 137.194.0.0/**16**
+ Règles d'attribution des blocs contigus d'adresses par localisation géographiques
  - Diminuer le nombre d'entrée dans des (grands) routeurs
  - Situation avant :
    + attribution non contrôlée
    + Deux réseaux « voisins » dans l'espace des adresses peuvent se retrouver sur deux continents

Masque (1/2)

+ Nécessité de reconnaître la partie Net-Id (y compris SubNet-Id) dans une adresse
+ Technique : utilisation d'un masque
+ Masque : séquence binaire de 32 bits dont les N premiers bits valent 1 et le reste 0
+ N : longueur de Net-Id
  - Notation : adresse réseau / N
  - Exemple : 137.194.0.0 /16
+ Masque = toujours associé à un réseau
  - M = 1111….. 111 000000
  - Idée : par une opération **AND** : faire apparaître uniquement la partie Net-Id

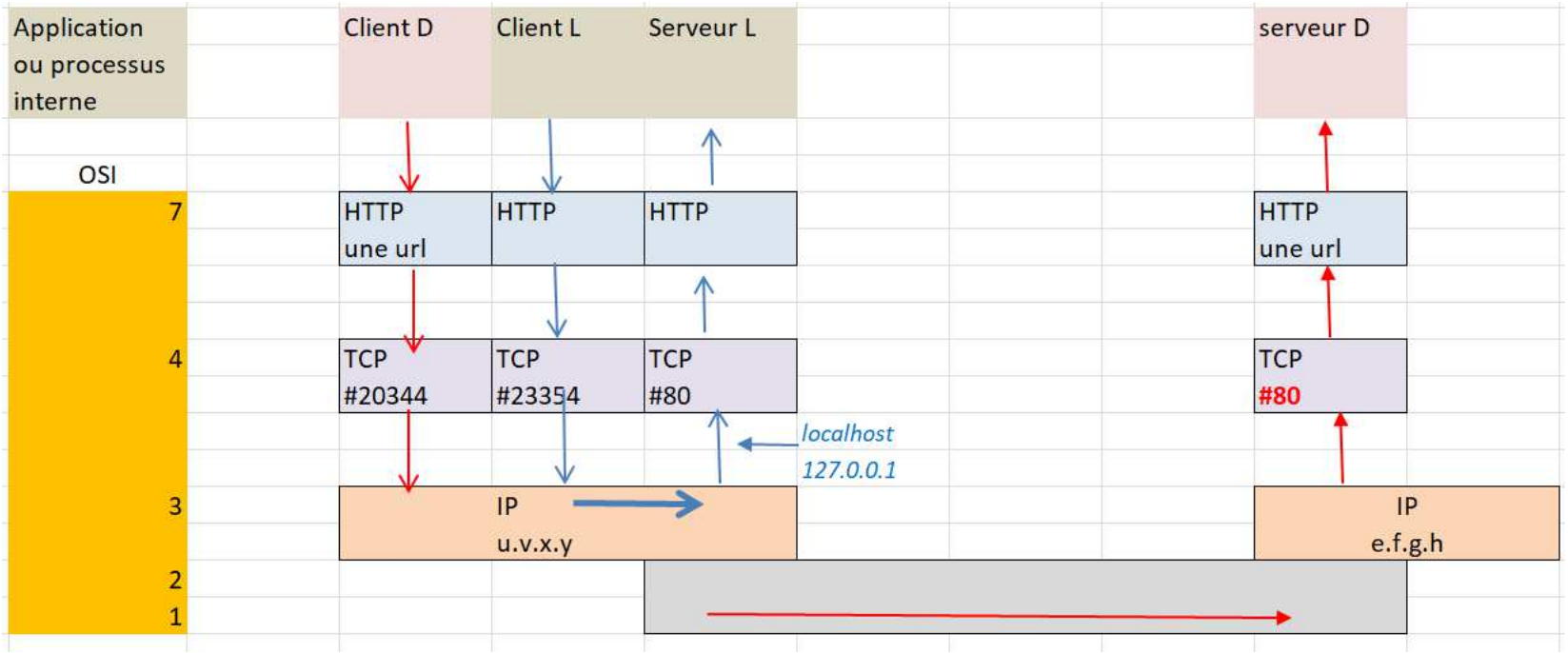
Masque (2/2)

+ Procédé
  - Objectif : vérifier si un paquet avec DA=*A* (ici : 137.195.3.4) appartient au réseau de l'ENST (PF=137.194.0.0/16)
  - M = 255.255.0.0
+ Opérations
  - M (ENST)=255.255.0.0
  - A **AND** M donne R=137.195.0.0
  - **Compare** R avec PF (Net-Id de l'ENST)
  - Résultat : Négatif
+ Conclusion
  - A n'appartient pas au réseau ENST
  - Le paquet n'est pas à être dirigé vers le réseau ENST

Adresses particulières (1/2)

+ 127.0.0.1 (en général) : loopback, localhost
  - Retourné dans sa propre entité
  - communication inter-processus au sein d'une machine
    + Application : test de logiciels de communication (sur la même machine): client et serveur Web
+ Adresse Réseaux (Net-Id)
  - Partie « Host » => 0
  - Exemple : 137.194.**0.0**.
+ Tous les bits partie machine à 1 : **diffusion**
  - Exemple : 137.194.255.255
    + Ce serait pour tout l'ENST (!)
+ Un réseau avec 2**n adresses peut donc avoir 2**n-2 adresses *utiles* (dont celle(s) de routeur(s))

LocalHost


Application ou processus interne
Client D
Client L
Serveur L
serveur D
OSI
7
HTTP une url
HTTP
HTTP
HTTP une url
4
TCP #20344
TCP #23354
TCP #80
TCP #80
localhost
127.0.0.1
3
IP u.v.x.y
IP e.f.g.h
2
1

Adresses particulières (2/2)

+ 0.0.0.0 : route par défaut
  - *0.0.0.0 peut aussi représenter une machine sans adresse*
    + *Station sans disque qui utilise RARP*
+ Adresses réservées aux réseaux privés (RFC1918)
  - Classe A : 10.0.0.0
  - Classe B : 172.16.0.0 à 172.32.0.0
  - Classe C : 192.168.0.0 à 192.168.254.0
+ Le trafic d'un réseau privé (***Intranet***) ne se mélange pas par définition avec celui d'**Internet** (public)
  - Une adresse de réseau privé n'est pas « *routable* » : elle n'est pas censée être traitée par les routeurs d'Internet

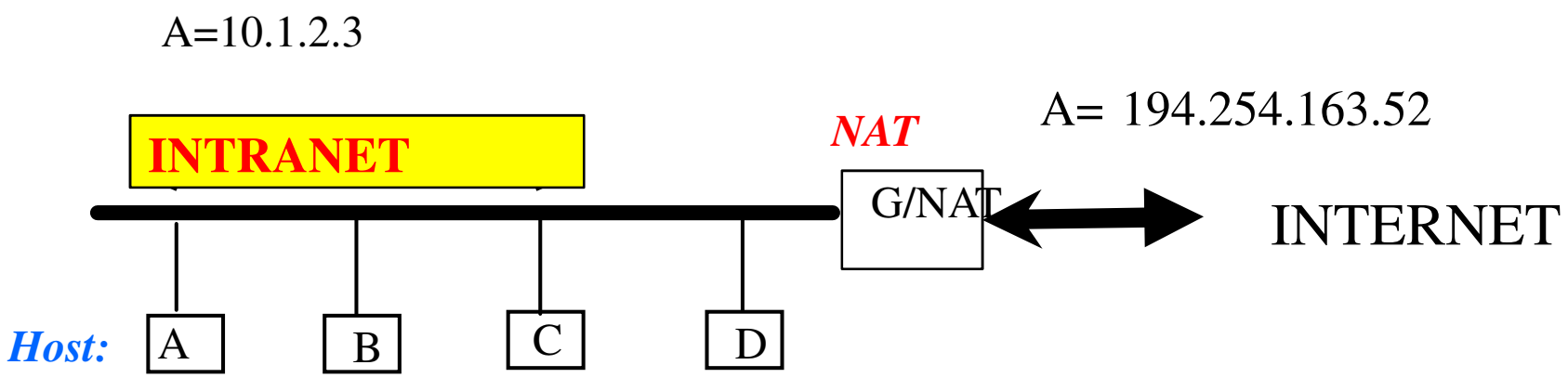
NAT et INTRANET (succinct)

+ INTRANET :
  - Un site protocolairement « Internet »
  - Avec son **propre plan d'adresse**
+ Avantage :
  - Propre organisation
  - Sécurité
    + Partie exposée à (accessible par) Internet = DMZ
  - Contournement de la pénurie des adresses
+ Problème : comment se connecter à Internet
+ Solution : NAT (network adresse translation)
  - NAT: passerelle entre Intranet et Internet

NAT et INTRANET (succinct)

+ NAT = Passerelle entre Intranet et Internet
  - Correspondance <a (adrs Intra), b (adrs INTERNET)>
+ Exemple :
  - Un site Intranet disposant de la plage d'adresse [194.254.163.46, 194.254.163.58]
  - Un sender intra A avec adrs = 10.1.2.3
  - NAT établit la correspondace < 10.1.2.3, 194.254.163.52>
    + Tout paquet issu de A vers Internet aura le champ SA remplacé par 194.254.163.52
    + Tout paquet reçu avec DA=194.254.163.52 sera remplacé par 10.1.2.3
+ En pratique : beaucoup de communications en parallèle
  - Etablissement dynamique des correspondances
  - Translation combinée "adresse ET port" (NAPT)
  - Adresse = niveau 3(IP), port=niveau 4 (TCP/UDP)

NAT



G

G: vers Internet:
je prend
194.254.163.52

A la réception d'un paquet IP pour A
194.254.163.52 **traduite** en 10.1.2.3

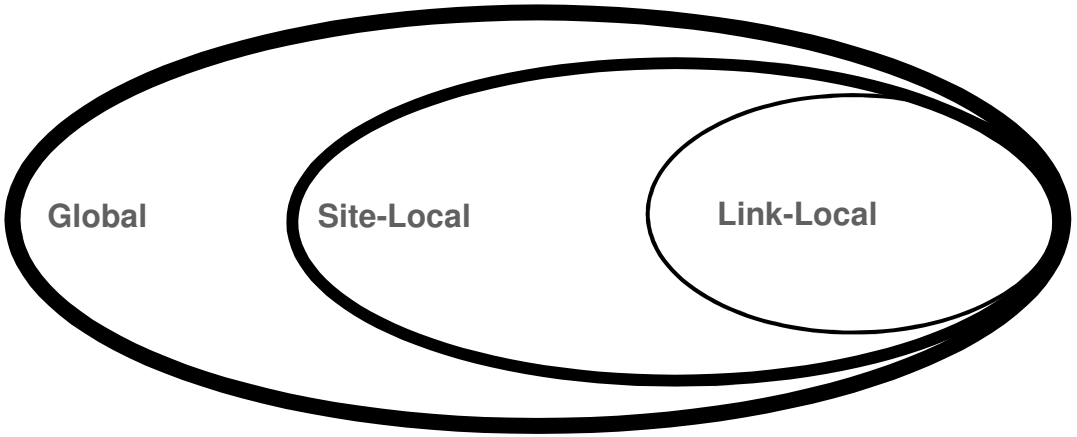
Adresses IPv6

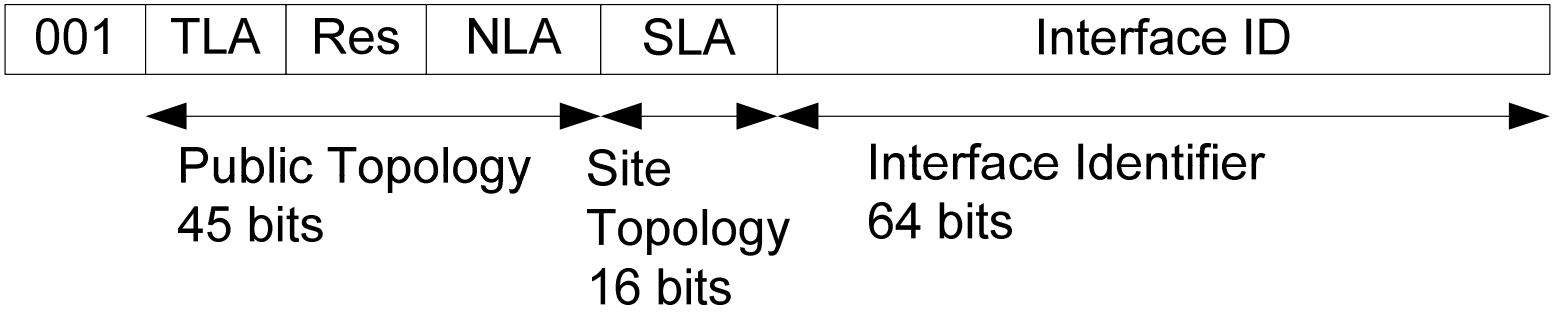
+ Specification : RFC 2373, RFC2374
+ Codé sur 128 bits (16 octets)
  - (Rappel: Adrs IPv4: 32bits = 4 Giga)
  - 2**128=3,4 E38 = 665E21/m2 (surface de la Terre)
    + Visionnaire: goodnew pour IoT!
+ Constitué de deux parties :
  - Préfix (pour routage)
  - Identificateur d ’interface (IID)
+ Hiérarchie
  - Global, puis site, puis local

Types d'adresses

+ Unicast
  - une et une seule interface physique (carte d'accès)
+ Multicast
  - un groupe d'interfaces (i.e. d'adresses unicast)
+ Anycast
  - une interface quelconque dans un groupe
  - Généralement pour les communications type « à l'inteface la plus proche) (au sens distance de routage)
+ Pas de broadcast
+ Spéciales
  - compatibles/mappée IPv4
  - loopback

Adresse IPv6





IPv6: Préfix

typed'adresse	Préfix
IPv4-compatible	0000...0 /96 (96 zero bits)
global unicast	**001**/3
link-local unicast	1111 1110 10/10
site-local unicast	1111 1110 11/10
multicast	1111 1111/8

+ Tous les autres PF sont reservés (7/8)
+ Addresse Anycast : assimilée à une unicast (du point de vue préfix)

ARP

Principe

Procédé

RARP

Position du problème

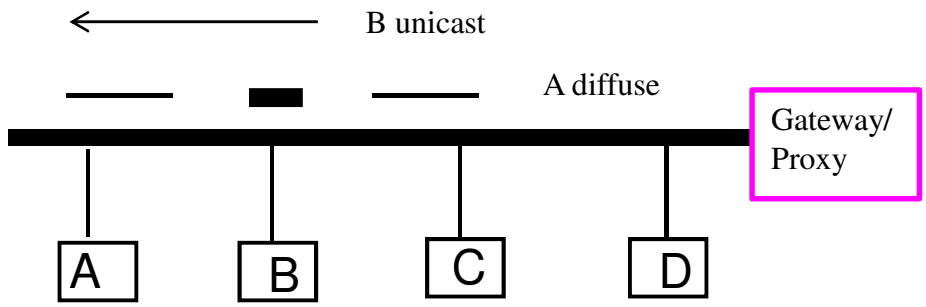
+ Une entité IP = identifiée par une adresse IP
  - Internet = un monde « virtuel » basé sur des liens physiques
+ Transmission sur un lien physique
  - Besoin d'identifier le récepteur physique correspondant
+ Nécessité d'une **résolution d'adresses**
  - établit une **correspondance** entre **Adrs IP et Adrs Physiqe**
+ Procédé général : liste de correspondance
+ Si **diffusion** possible (Ethernet en particulier) :
  - résolution automatisable
  - Le protocole **ARP**

ARP

+ Address Resolution Protocol (RFC826)
+ Développé initialement pour Ethernet
+ Objectif :
  - Trouver une adresse MAC en partant d'une adresse IP
+ Scenario :
  - Deux stations IP, A et B, sur le **même segment** Ethernet.
  - A veut envoyer un datagramme à B.
    + A connaît bien sûr Adrs IP de B, mais pas son adresse Ethernet
  - Problème : Comment l'obtenir **automatiquement** ?
+ Procédé ARP
  - A diffuse (**broadcast**, adresse DA = FFFFFF) une trame de réclamation (type = **0x0806**) qui contient l'adresse IP de B en particulier
  - Toutes les machines du réseau local reçoivent la requête.
  - Seul B répond à A (**unicast**) en lui donnant son adresse Ethernet.
  - Le problème est ainsi réglé

ARP: Schéma et format

+ Hardware type (HTYPE, 2) : 1 (Ethernet)
+ Protocol Type (PTYPE, 2): 0x0800 (IPv4)
+ Hardware adress length (HLEN, 1) : 6 (pour ethernet)
+ Protocole adress length (PLEN, 1) : 4 (pour IPv4)
+ Operation (OPER, 2): 1 (Request) ou 2 (Reply)
+ Sender HW adrs (SHA) : LSB en premier (1er octet d'aboord)
+ Sender Protocol adrs (SPA): idem
+ Target HW adrs (THA): idem
+ Target Protocol adrs (TPA): idem
+ (Total: : 28 octets)



ARP : Cache

+ ARP procède par diffusion
  - Procédure lourde
  - Délai supplémentaire lié au dialogue
+ Mise en **cache** des correspondances trouvées par ARP
  - Table de routage direct
    + Commande « **arp –a** »
+ Avantage
  - Moins de trafics
  - Moins d'accès (donc de risques de collision)
  - Délai plus court

ARP : Proxy

+ Proxy = Agent = (ici) un routeur
+ Scénario :
  - Idem précédent sauf A et B sont sur deux segments!
  - B ne pourra jamais répondre à la trame ARP
  - Le routeur doit répondre à la place de B
    + Critère : comparaison de l'appartenance de l'adresse IP de B à ce sous-réseau
+ Conséquence
  - Le routeur devient un agent (proxy) de B
  - **Il attire vers lui le trafic** vers B, et, d'une manière générale, le **trafic sortant**.
+ Problème potentiel de sécurité:
  - ARP prend toutes les réponses….

ARP particulier

+ Gratuitous (type Request)
  - SPA =TPA = adrs IP sender, THA=0
  - Initiative d'actualisation
+ ARP Probe (type Request)
  - SPA=0.0.0.0., TPA= adrs IP sender
  - Si conflit, l'autre entité IP avec la même adresse A répondra
    + Détection de conflit
  - Similaire à Gratuitous
    + Gratuitous: annonce, Probe: détection de conflit

RARP

+ *Reverse Address Resolution Protocol (RFC903)*
+ Problème dual de ARP
  - Obtention d'une adresse IP
  - Utile pour les machines n'ayant pas de mémoire non volatile
+ Même procédé que ARP
  - Diffusion d'une trame (type = **0x8035**)
  - Seul le Serveur RARP répond avec une adresse IP

+ **Successeur: DHCP**

IP

Principe

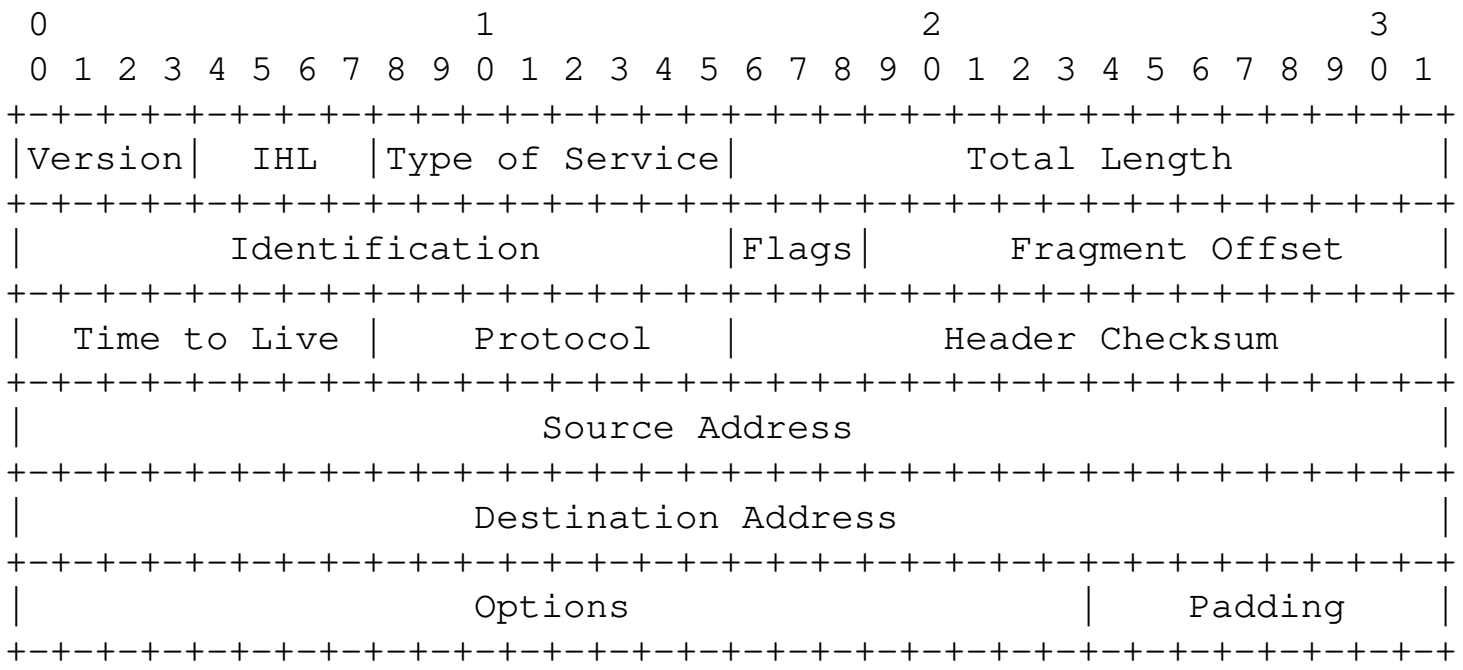
Format

Fonctionnalité

Généralités

+ IP = Internet Protocol (RFC 791)
+ Correspond à la couche 3 de l' OSI
+ Service = Datagram
  - Communication non fiable
+ « **Socle** » de Internet
  - **Unique format** pour les échanges de données dans Internet
+ Mode de fonctionnement : Best-Effort
  - Jusqu'à épuisement des ressources (BP, CPU, RAM)
  - Pas de réservation de ressources
  - Solution simple et économique
  - Pas de garantie de **QoS** (Qualité de service)

Format de l'en-tête IP



Format

+ Organisation par lot (ligne) de 32 bits
+ En-tête de taille variable (mais tjrs n*32 bits)
  - Min (la plus courante) = **20 octets**
  - Max. = 60 octets (15*32 bits)
+ Taille paquet Max. (**théorique**) : 64 Ko

Les champs (1/2)

+ Version du protocole IP (IPv4)
  - (la nouvelle génération, **IPv6**, utilise)
+ IHL (Longueur de l'en-tête) : en mots de 32 bits
  - Min : 5 (sans d'option), Max: 15
+ Type of service (TOS) => devenu **DSCP** (cf . TOS->DSCP)
+ Longueur totale du "fragment" sur 16 bits (en-tête + données)
  - Taille max. = 64 Ko
  - Champ permettant de **délimiter le paquet IP** dans Ethernet
    + distinction infos utiles / infos de bourrage

Les champs (2/2)

+ Identification+Flag+Offset : cf Fragmentation
+ TTL (Time to Live) : crédit de traitement (cf. TTL)
+ Protocole : Identifie le protocole de niveau supérieur
  - 1 : ICMP, 2 : IGMP, 6 : TCP, 17 : UDP
+ Somme de contrôle de l'en-tête : détection d'erreurs
+ Options : facultatif
  - longueur variable (rajout de bourrage pour avoir n*32 bits)
  - Détails dans RFC 1700

TOS -> DSCP

+ ToS : objectif initial : options de gestion (RFC1394)
  - 3 bits LSB : **Précédence** (priorité du datagramme)
    + Pas utilisé
  - 5 bits MSB (dernier non utilisé)
    + 4 drapeaux pour le choix du lien
    + délai court (telnet,ftp), débit élevé(ftp), fiabilité (snmp) et coût faible (nntp).
  - Défaut : 00000000
+ Devenu : DSCP = Diffentiated Service Code Point
  - Les 6 LSB : définir jusqu'à 64 classes de services
  - Gestion avec l'approche **Service différencié** (par classe)
  - Les 2 MSB : non utilisées

Fragmentation : principe

+ Fonctionnalité de base de la couche 3
  - Liens de « gabarits » différents a priori
+ Opération : Fragmentation /Restitution
  - à la source : préférentielle mais pas toujours prévisible
  - Dans le réseau : solution évidente mais pas très efficace
+ Restitution
  - Jamais dans le réseau
  - Toujours chez le récepteur final

MTU

+ MTU = Maximum Transmission Unit
+ Taille max. qu'un lien puisse transiter
+ MTU spécifique à chaque lien
+ **MTU Ethernet = 1500 octets**
+ **MTU minimal conseillé sur Internet >= 576 octets**
  - *Valeur minimale : 68 octets*

+ Procédure de découverte de MTU dans Internet
  - Basé sur ICMP et les champ Fragmentation de IP

Fragmentation dans IP (1/2)

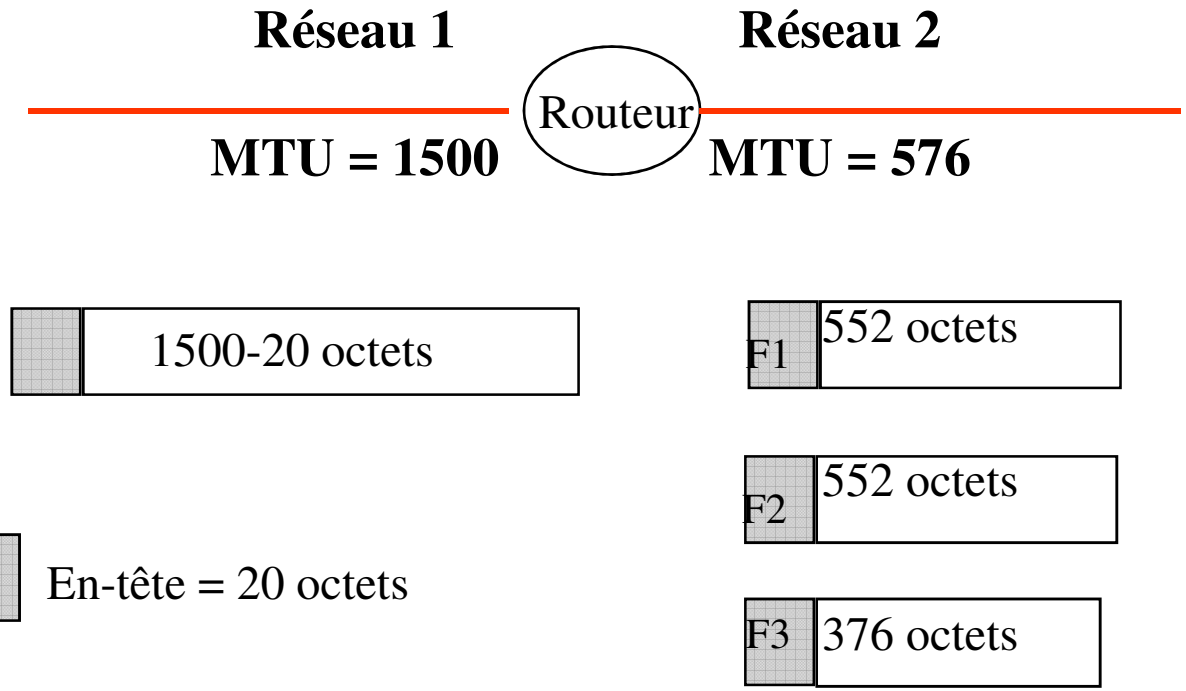
+ Trois champs
+ Identification : référence au datagramme initial.
  - Utilisé par le récepteur pour la reconstitution du datagramme.
  - les fragments auront le même identification
+ Flags
  - 3 bits dont 2 utilisés : (0 , DF , MF)
  - DF : Don't Fragment (interdiction de fragmentation)
  - MF : More Fragment (fragment NON terminal)

Fragmentation dans IP (2/2)

+ Offset :
  - Positionnement relatif dans le paquet initial
  - ***Multiple de 8 octets***
    + 13 bits (alors que taille paquet es sur 16 bits)
+ Exemple :
  - Un paquet de 1500 fragmentés en 3
  - F1 : offset = 0, MF =1
  - F2 : offset = 69 = 552/8 (val. Max., car 560+20>576), MF=1
  - F3 : offset = 138 (69*2), MF=0 (FIN)
+ Fragmentation intermédiaire non désirées
  - Faire travailler le routeur
  - Pas efficace
  - Fragmentation à la source recommandée
    + Découverte de MTU minimal sur le chemin

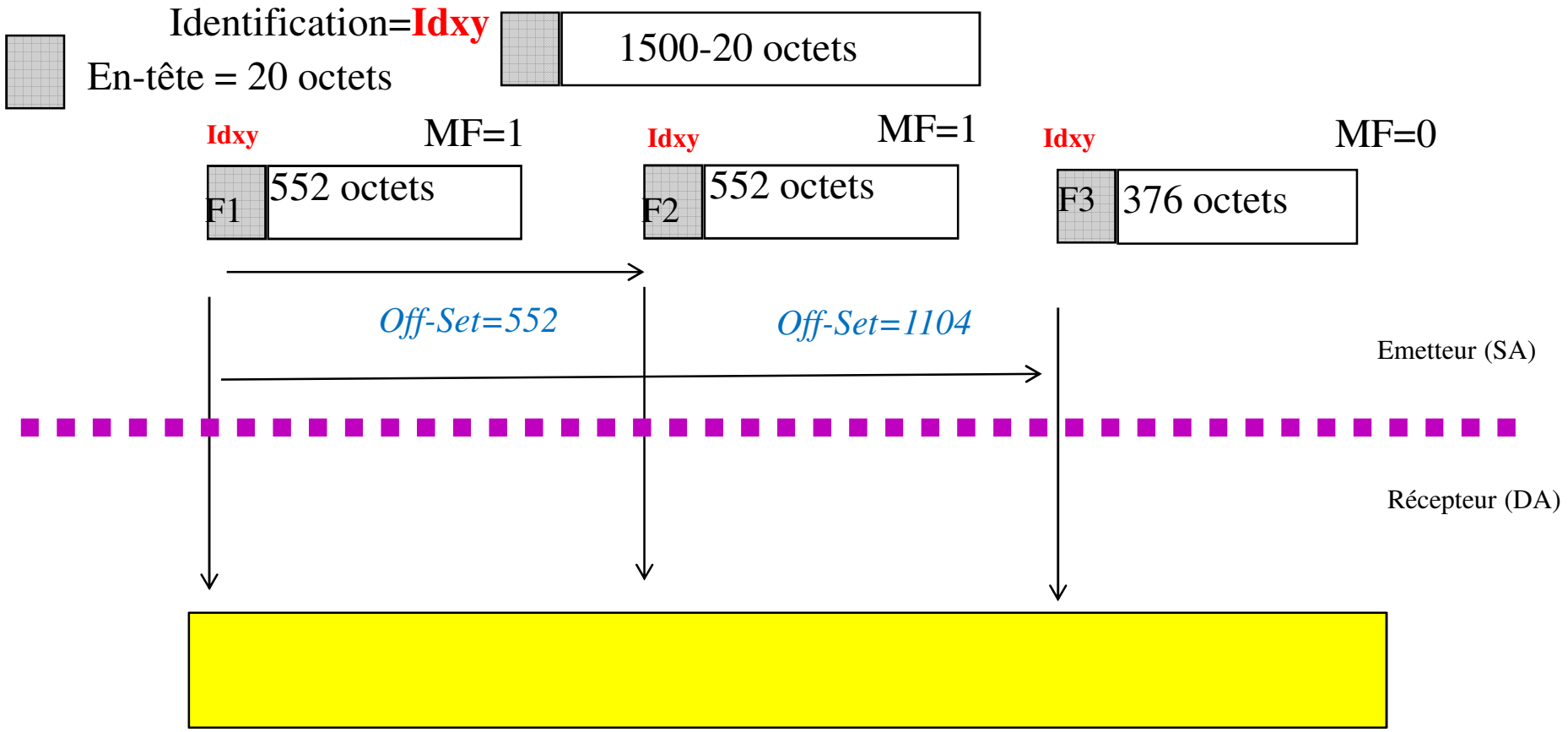
Fragmentation : Exemple

552=**69*8**
556=69*8 + ***4***



1 paquet rempli vs *3* paquets (aucun ***n'est rempli***)

Fragmentation/Restitution



Identification **Idxy**
DA, SA, Protocol

Chantier

TTL

+ Principe :
  - IP : datagramme -> électron libre après émission
  - Nécessité d'éliminer les paquets tournant en rond
    + À cause des risques de boucles en cas de problèmes de routage
+ Idée de départ :
  - Limitation exprimé en **temps**
  - Moins facile à exploiter qu'un crédit
+ Utilisation actuelle : crédit de **poursuite du routage**
  - Décrémentation (de 1) par chaque routeur **intermédiaire**
  - Détruire le paquet (par le **routeur intermédiaire**) si **TTL=0**
    + un message d'erreur est renvoyé à l'émetteur.

IP: Capture Wireshark

```
Frame 17: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: BrocadeC_04:ce:00 (00:1b:ed:04:ce:00), Dst: Sony_bd:20:d0 (00:1d:ba:bd:20:d0)
Internet Protocol Version 4, Src: proxy.enst.fr (137.194.2.20), Dst: dhcp26-233.enst.fr (137.194.26.233)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 40
    Identification: 0xa72b (42795)
  Flags: 0x4000, Don't fragment
      0... .... .... .... = Reserved bit: Not set
      .1.. .... .... .... = Don't fragment: Set
      ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 63
    Protocol: TCP (6)
    Header checksum: 0x6423 [validation disabled]
    [Header checksum status: Unverified]
    Source: proxy.enst.fr (137.194.2.20)
    Destination: dhcp26-233.enst.fr (137.194.26.233)
Transmission Control Protocol, Src Port: ndl-aas (3128), Dst Port: 53555 (53555), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 1d ba bd 20 d0 00 1b  ed 04 ce 00 08 00 45 00   ···· ··· ·····E·
0010  00 28 a7 2b 40 00 3f 06  64 23 89 c2 02 14 89 c2   ·(·+@·?· d#·····
0020  1a e9 0c 38 d1 33 41 10  19 72 0a 0d 9f d1 50 11   ···8·3A· ·r····P·
0030  00 37 9d 4e 00 00 00 00  00 00 00 00              ·7·N···· ····
```

Wireshark *(https://www.wireshark.org/) capture de traces et analyseur de protocoles*

IP : en guise de conclusion

+ Une solution éprouvée (vu le volume du trafic IP…)
+ Quelques défauts
  - En-tête variable
    + Examen systématique de IHL
    + Peu rentable car la majorité avec IHL=5
  - Calcul de checksum de l'en-tête
    + Consommateur CPU
  - Capacité de gestion limité
    + ToS -> DSCP est une évolution récente
  - Possibilité très limités d'étendre/faire évoluer les options

+ … vers IPv6 ….

ICMP

Principe

Format

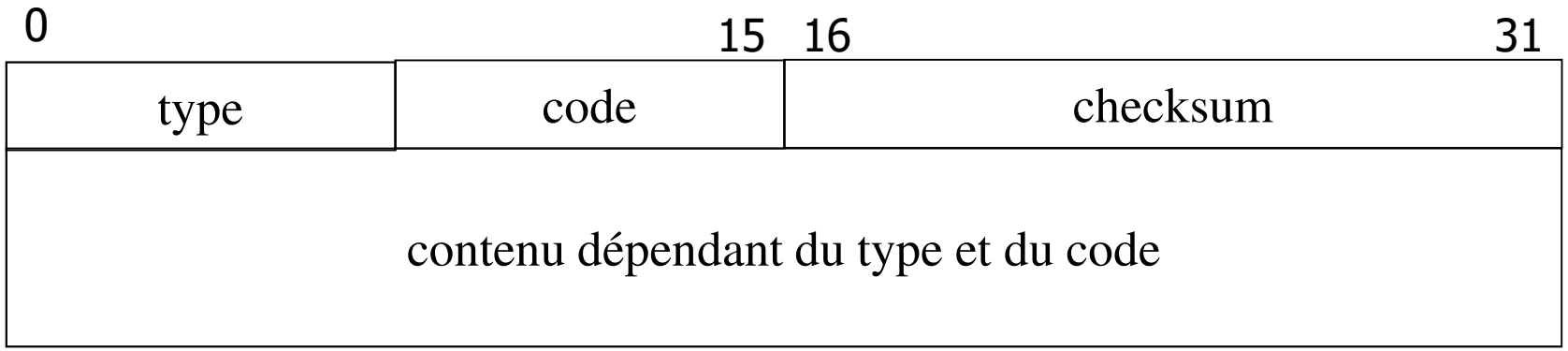
Fonctionnalités

Usages

Généralités

+ ICMP = INTERNET Control Messages Protocol
+ Compagnon de IP (fonctionne ensemble)
+ Protocole de signalisation
+ Transmis via un paquet IP (protocol = 1)
+ Deux catégories de messages
  - Query : Information diverses
    + Génération suivant besoins
  - Error : Signalisation d'erreur
    + Génération suite au constat d'une erreur (perte de paquet en gén.)

Format


0
15
16
31
type
code
checksum
contenu dépendant du type et du code

Règles pour ICMP Error

+ Pas de génération de ICMP Error pour la perte de
  - un autre message ICMP
    + Pour ne pas tourner en round
  - un paquet destiné à une adresse broadcast
  - un paquet dont l'expediteur n'a pas une adresse unique (adresse zéro, bouclage, adresse broadcast)
    + ICMP serait envoyé à tout le monde sinon!
  - un fragment autre que le dernier
+ Insertion des info (les en-têtes) dans le contenu
  - l'entête IP,
  - l'entête du protocole supérieure (e.g. TCP)
  - les 8 premiers octets des données utilisateurs.

Les Messages : type/code/description

0	0	réponse echo (ping)
3		destination inaccessible
	0	réseau inaccessible
	1	machine inaccessible
	2	protocole inaccessible
	3	port inaccessible
	4	fragmentation nécessaire
	5	échec de la route source
	6	réseau de destination inconnue
4	0	débit trop élevé
5	0	redirigé
8	0	requête echo (ping)
9	0	avertissement du routeur
10	0	sollicitation du routeur
11		temps dépassé:
	0	TTL vaut 0 pendant le transit
	1	TTL vaut 0 pendant le réassemblage
12		problème de paramètre
	0	mauvaise entête IP
	1	option requise manquante
13	0	requête timestamp
14	0	réponse timestamp
17	0	requête de masque d'adresse
18	0	réponse du masque d'adresse

Exemples

+ Echo/Ping
  - Couple Echo Request (8), Echo Reply (0)
  - Visibilité applicative : PING
  - Usages divers
  - Test si la machine distante est en activité
  - Métrologie
+ Découverte MTU
  - Un paquet assez long avec DF=1
  - Retour ICMP 3 /4 (fragmentation nécessaire) avec MTU

Traceroute (1/2)

+ Application basée sur ICMP
+ Principe : forcer les routeurs à se « démasquer » en envoyant un ICMP
+ Boucle commencée avec TTL=1
  - Premier routeur démasqué et ainsi de suite
  - Même processus répété trois fois par étape
+ Processus s'arrête lors de réception d'un ICMP 3/3
  - Un port fictif est choisi express pour forcer ICMS 3 / 3 (port inconnu) par le destinataire

Traceroute (2/2)

```
+  1  toutatis-local.res.enst.fr (137.194.192.14)  2 ms  2 ms  2 ms
+   2  panoramix-bbone.enst.fr (137.194.2.5)  2 ms  2 ms  2 ms
+   3  enst-paris.rerif.ft.net (192.33.155.1)  2 ms  3 ms  2 ms
+   4  danton1.rerif.ft.net (193.48.58.105)  179 ms  132 ms  109 ms
+   5  stlamb3.rerif.ft.net (193.48.53.49)  107 ms  229 ms  183 ms
+   6  stamand1.renater.ft.net (192.93.43.115)  159 ms  75 ms  63 ms
+   7  stamand3.renater.ft.net (192.93.43.17)  111 ms  171 ms  129 ms
+   8  rbs1.renater.ft.net (192.93.43.170)  315 ms  253 ms  139 ms
+   9  Paris-EBS2.Ebone.NET (192.121.156.226)  121 ms  101 ms  291 ms
+  10  icm-dc-1.icp.net (192.121.156.202)  315 ms  279 ms  249 ms
+  11  icm-dc-1-F0/0.icp.net (144.228.20.101)  205 ms  250 ms  348 ms
+  12  icm-fix-e-H2/0-T3.icp.net (192.157.65.122)  304 ms  341 ms  420 ms
+  13  * * mf-0.enss145.t3.ans.net (192.203.229.246)  296 ms
+  14  t3-2.cnss56.Washington-DC.t3.ans.net (140.222.56.3)  287 ms  360 ms  262 ms
+  15  t3-0.cnss32.New-York.t3.ans.net (140.222.32.1)  234 ms  186 ms  241 ms
+  16  t3-0.cnss48.Hartford.t3.ans.net (140.222.48.1)  129 ms  230 ms  200 ms
+  17  t3-2.cnss40.Cleveland.t3.ans.net (140.222.40.3)  216 ms *  509 ms
+  18  t3-0.enss131.t3.ans.net (140.222.131.1)  293 ms  264 ms  227 ms
+  19  fdd0.michnet1.mich.net (192.203.195.4)  263 ms  293 ms  373 ms
+  20  nic.merit.edu (35.1.1.48)  238 ms  453 ms *

+  (1996, ENST)
```

Traceroute (2/2 bis)

+ Détermination de l'itinéraire vers www.l.google.com [173.194.37.104]
+ avec un maximum de 30 sauts:

+ 1 1 ms 5 ms 1 ms 192.168.1.1
+ 2 2 ms 3 ms 1 ms 10g2-ls.enst.fr [137.194.26.252]
+ 3 7 ms 2 ms 1 ms goudurix.enst.fr [137.194.4.252]
+ 4 3 ms 7 ms 4 ms gi9-22.ccr02.par04.atlas.cogentco.com [130.117.8.49]
+ 5 3 ms 5 ms 7 ms te0-2-0-4.mpd21.par01.atlas.cogentco.com [130.117.50.145]
+ 6 13 ms 11 ms 12 ms te8-7.mpd02.lon01.atlas.cogentco.com [130.117.3.6]
+ 7 16 ms 14 ms 17 ms google.lon01.atlas.cogentco.com [130.117.14.90]
+ 8 20 ms 19 ms 11 ms 64.233.175.27
+ 9 23 ms 13 ms 14 ms 209.85.251.202
+ 10 12 ms 15 ms 22 ms lhr14s02-in-f104.1e100.net [173.194.37.104]

+ (1-déc-2010, B310, TPT)