PARTIE I: L'ARCHITECTURE

ARCHITECTURE DU PROTOCOLE IP

Architecture du Protocole IP

- L'architecture TCP/IP, crée il y a environ 40 ans par **Vint Cerf** et **Robert Kahn,** est maintenant utilisée sur des milliards d'appareils sur terre.

- Améliorations introduites:
  - Multicast
  - Qualité de service (QoS)
  - Traffic engineering
  - Services temps réel

→, Mais l'architecture a été complètement conservée !!!

De NCP à TCP/IP

- **ARPANET**: projets fondés par Advanced Research Project Agency.
- **NCP**: développé en 1970 pour interconnecter les ordinateurs avec un "Interface Message" (IMP) sur un réseau dorsal avec un débit de quelques Kbps.
- Vers la fin 1971, 15 sites étaient interconnectés avec le protocole NCP (premier noyau d'internet).
- **Robert Kahn** et **Vint Cerf** ont conçu TCP (TCP/IP pas encore couplé) pour remplacer NCP.
- ARPANET → premier réseau opérationnel utilisant le principe de la communication de paquet (*packet switching*) pour la communication entre hôtes.
- Prochaine génération **IPv4** en **1981** → Internet a migré à cette technologie.
- **National Science Foundation** (**NSF**) a joué un rôle majeur dans le développement d'internet.

Principes Fondamentaux de Conception Architecturale de TCP/IP

Buts originaux de TCP/IP :

- La communication internet doit être maintenue même en cas de perte de réseau ou de passerelle (gateway).
- Internet doit supporter plusieurs types d'appareils de communication.
- L'architecture d'internet doit:
  - S'accommoder à une grande variété de réseaux (couche liaison et physique)
  - Permettre une gestion distribuée de ses ressources
  - Dois être rentable
  - Permettre l'ajout d'un hôte facilement
- Les ressources utilisées dans l'architecture d'internet doivent être raisonnables.

Principes Fondamentaux de Conception Architecturale de TCP/IP

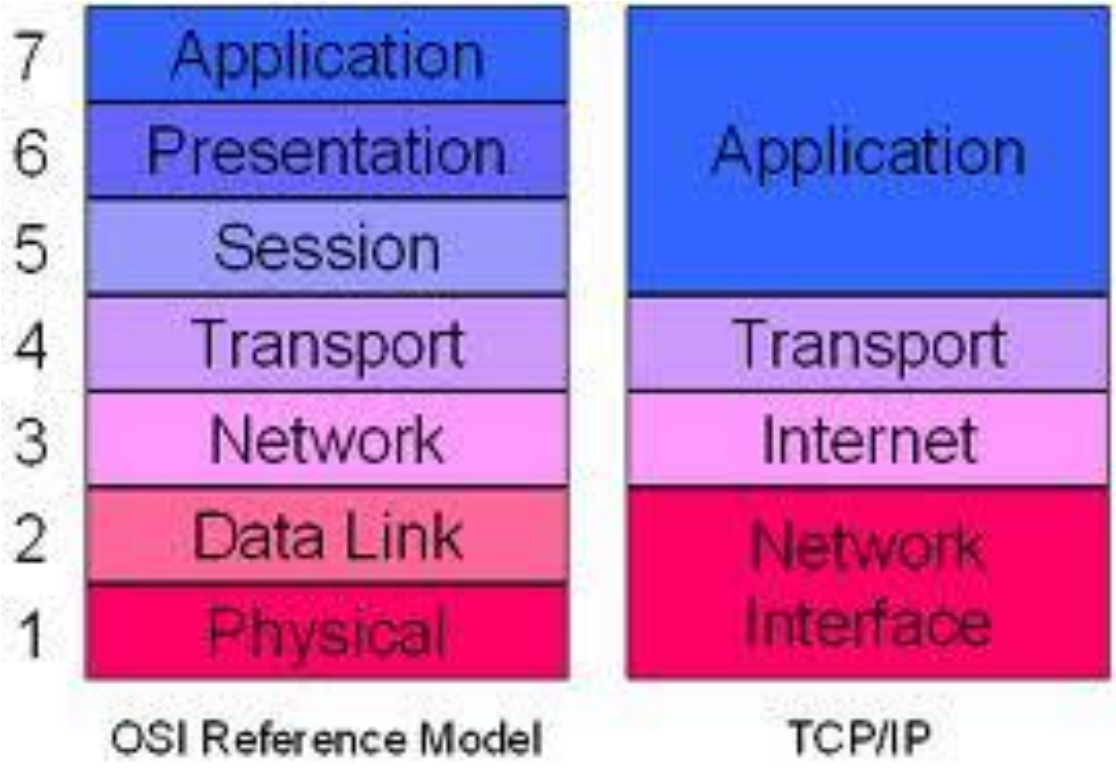
Objectifs: construire un réseau fiable et flexible capable de supporter divers appareils utilisant différentes couches liaison et physique.

- **Flexibilité**→ adoption d'une architecture en couches.
- **Fiabilité**→ technique de protection/restauration (par ex.: MPLS Traffic Engineering).
- Panel **d'appareil supporté** → du microcontrôleur 8-bit aux puissants serveurs hébergés dans les *datacenters*.
- **Extensibilité** → de 9 ordinateurs dans ARPANET en 1970 à plusieurs milliards d'ordinateurs aujourd'hui.

→ La conception réseau requiert de bonnes connaissances des capacités de chaque couche afin de faire le bon choix du protocole.

Principes Fondamentaux de Conception Architecturale de TCP/IP

Concept des couches: une des conceptions clés pour une architecture flexible.



Principes Fondamentaux de Conception Architecturale de TCP/IP

Les couches de l'architecture du protocole TCP/IP (IETF):

- **Couche liaison**(**PHY/MAC**): responsable de transmettre les paquets IP entre deux appareils (point à point ou point à multipoint) :
  - Media access control (MAC)
  - Détection d'erreur et retransmission si nécessaire
  - Contrôle de flux
- **Couche internet** (**IP**): responsable de fournir un service non fiable pour envoyer les paquets de la source à la destination à travers le réseau:
  - Fonction principale: routage
  - ICMP et IGMP (IGMP: pour la partie trafic multicast de la couche IP).
- **Couche transport**: responsable de la communication end-to-end entre deux appareils où les états sont maintenus :
  - TCP: fournis un mécanisme de transport avec détection d'erreur et retransmissions, contrôle de flux, mécanisme de sécurité, etc.
  - UDP: utilisé dans la plupart des cas par des applications qui implémentent elles-mêmes un mécanisme optionnel de détection d'erreur.
- **Couche application**: protocoles de haut niveau qui soutiennent les applications (FTP, TFTP, SNMP; HTTP, etc.)

Optimisation Cross-Layer ?

Isolation stricte entre les couches :

- Avantages:
  - Flexibilité: pas de dépendance entre les couches
- Désavantages:
  - Redondance possible (par ex. : récupération en cas d'erreur et gestion de congestion)

  → Le Cross-layering peut être plus rentable et fiable, mais risqué !

- Optimisation cross layer pour les objets intelligents:
  - Par ex. : « content routing »: router le trafic dans le réseau en fonction du contenu du paquet au niveau de l'application (à la place d'utiliser la destination IP) → réduit le trafic dans le réseau.
  - Essayer de mélanger les couches réseau et liaisons, car maintenir deux couches est lourd pour les réseaux à contraintes.
  - Essayer de router au niveau de la couche liaison.

  → les piles IPv6 légères actuelles ne requièrent que quelques kilobytes de RAM et quelques douzaines de kilobytes de Flash avec une puissance de calcul limitée et qui peuvent fonctionner sur des microcontrôleurs 8 bits d'entrée de gamme.

IP POUR LES OBJETS INTELLIGENTS

Challenges des Réseaux d'Objets Intelligents

Future du protocole IP pour les réseaux d'objets intelligents?

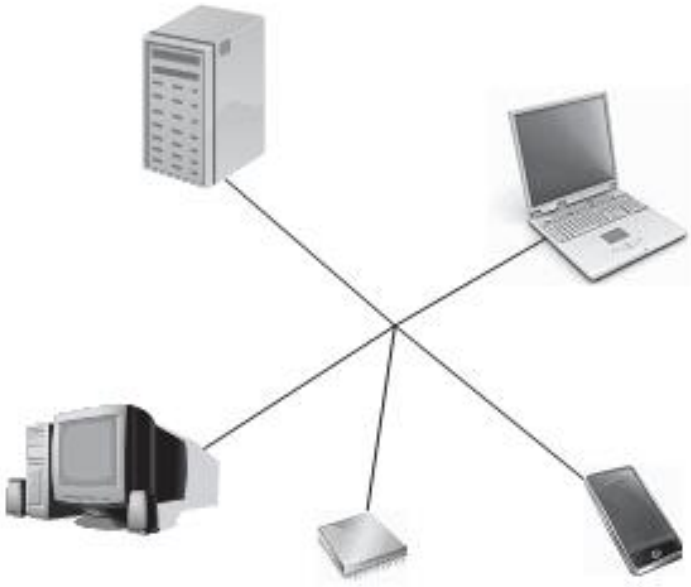
→ Challenges des réseaux d'objets intelligents:

- Évolutivité
- Extensibilité
- Diversité des applications
- Diversité des technologies de communication
- Interopérabilité
- Standardisation
- Technologie de communication avec perte potentielle
- Durée de vie
- Basse consommation
- Faible coût

Caractéristiques de l'Architecture IP

Interopérabilité

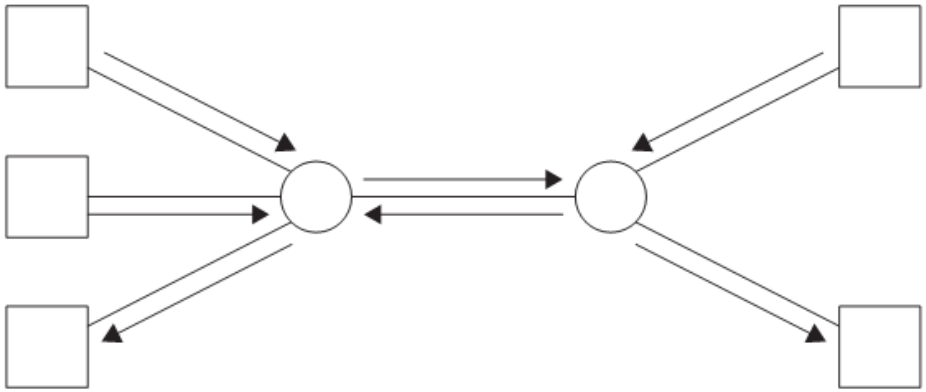
- IP fonctionne avec des couches liaison aux caractéristiques très différentes.
- IP peut fonctionner au travers de différentes plateformes, appareils et moyens de communication.



Caractéristiques de l'Architecture IP

Une architecture polyvalente et évolutive

- La polyvalence s'observe lorsque l'application s'exécute sur les terminaux et que le réseau transporte seulement des données entre eux → Permet au système d'évoluer.



Caractéristiques de l'Architecture IP

Stabilité et universalité de l'architecture

- Les objets intelligents sont conçus pour voir une longue durée de vie (jusqu'à 10 ans) → la technologie de base devrait rester disponible jusqu'à la fin du cycle de vie du système.
- L'architecture IP existe depuis près de 30 ans : la fondation de sa technologie de communication basée sur l'envoi de paquet est restée robuste.
- IP forme la base de l'internet public → Son architecture et ses standards ont devant eux un avenir pérenne.
- Connaissance de l'utilisateur, formations, livres, outils pédagogiques disponibles dans différents langages → Stabilité.

Caractéristiques de l'Architecture IP

Extensibilité

- Peu d'architectures de communication ont connu un déploiement à une telle échelle que IP

→ IP peut être déployé sur un large nombre de systèmes et fonctionner par le biais d'une grande variété d'implémentations différentes de son protocole

Caractéristiques de l'Architecture IP

Configuration et gestion

- IP donne accès à des mécanismes de configuration et de gestion avancés, mais également à une configuration automatique

- Mécanismes de configuration : pour toutes les couches, telles que:
  - Mécanismes automatiques et gérés pour l'assignation d'adresses réseau.
  - Mécanismes de routage.

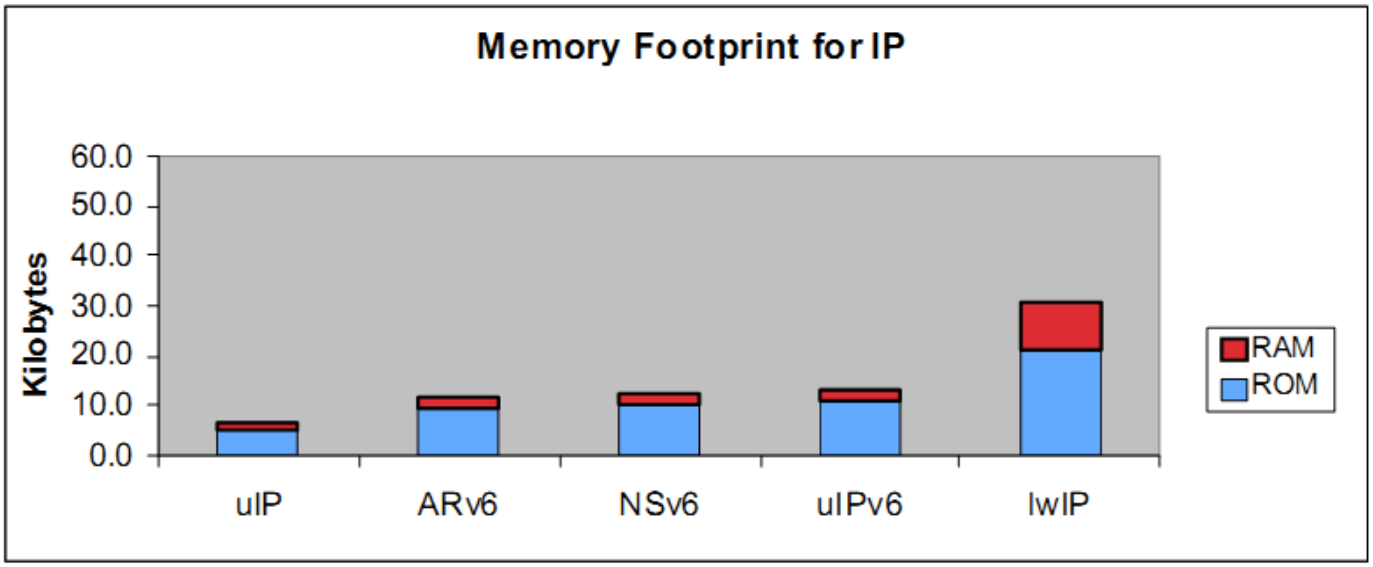
- Mécanismes de gestion : pour toutes les couches, telles que:
  - Dynamic Host Configuration Protocol (DHCP): pour l'assignation d'adresses individuellement ou en masse.
  - SNMP: pour l'inspection d'un réseau, sa configuration et sa performance.

→ Configuration, gestion, installation et la mise en service sont manifestement des problèmes liés aux réseaux d'objets intelligents

Caractéristiques de l'Architecture IP

Empreinte mémoire réduite

– Basse consommation en énergie, petite taille physique et faible coût → contrainte en mémoire et en complexité du software embarqué.



Alternatives à l'Architecture IP

- Au début, la communauté des réseaux de capteurs sans fil (une sous-catégorie des objets intelligents) a rejeté l'architecture IP car n'était, selon eux, pas applicable à ces systèmes.
- Alors, des recherches sur de nouvelles architectures ont été effectuées.
- Après plusieurs années, ils sont revenus aux architectures en couche pour les avantages de la modularité et la séparation des problèmes.
- La communauté des communications sans-fil basse consommation ont effectué la même transition:
  - Fin 1990 : définition d'une nouvelle architecture (ZigBee). ZigBee est initialement défini comme une pile réseau qui fonctionnerait bien avec le transfert de données sans fil et basse consommation pour le contrôle d'applications.
  - En 2009, ZigBee annonce qu'ils vont adopter IP comme mécanisme de communication.

Pour quoi les Passerelles (Gateways) ne sont pas désirées ?

- Alternative au principe “end-to-end” d’IP pour interconnecter des objets intelligents non-IP à un réseau IP.
- Mais:
  - Une traduction multi-protocoles présente une complexité inhérente:
    - Conversion de format des paquets
    - Traduction des mécanismes implémentés (routage, QoS, récupération d’erreur, transport, gestion, dépannage et modèles de sécurité).
  - Manque de flexibilité et d’extensibilité:
    - Deviens un goulot d’étranglement
    - Influe sur la sécurité globale (états indésirables sur le réseau).

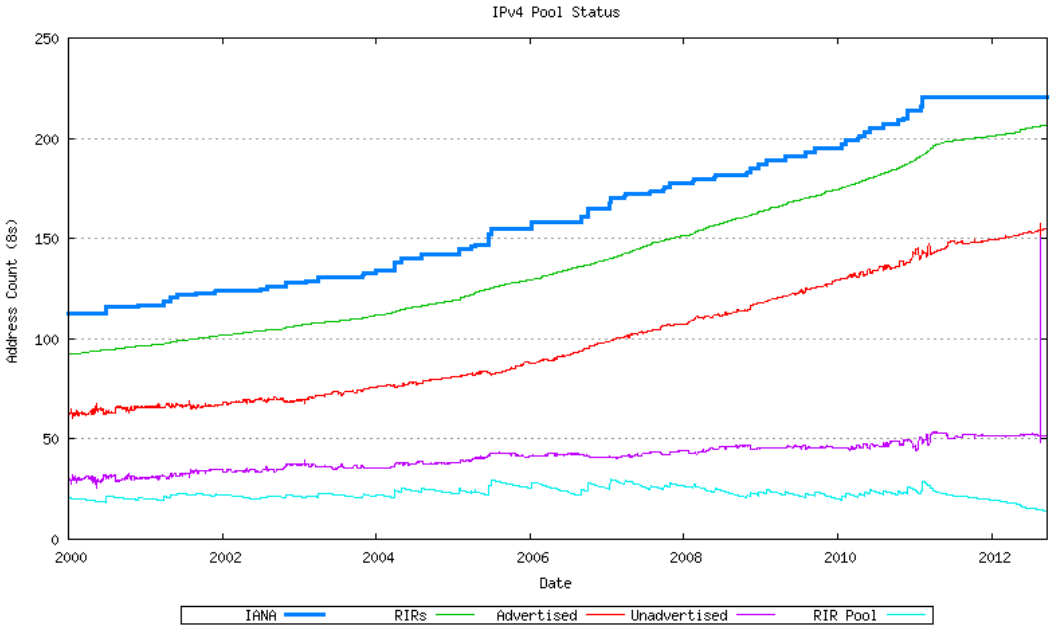
IPV6 POUR LES RÉSEAUX DE OBJECTS INTELLIGENTS

IPv6 - Introduction

- IPv6 est une évolution de IPv4 qui est construite sur la base de celle-ci. IPv6 préserve donc les principes architecturaux de IPv4 tout en implémentant de nouvelles fonctionnalités. De plus:
  - Certains protocoles ajoutés à IPv4, pour certaines utilisations spécifiques, ont été intégrés dans la pile IPv6.
  - L'entête a été modifié afin de laisser plus de place pour l'adresse.
  - Simplification de la couche 3 et de l'architecture globale

L'Épuisement des Adresses IPv4

- Adresse IPv4 codée sur 32-bit → 4,294,967,296 adresses (théoriquement).
- Étant donné la croissance exponentielle des allocations d'adresses Internet, l'IETF, a commencé à prendre des mesures dans les années 90 dans le but de concevoir la nouvelle génération d'adresse IP.
- Quelques stratégies ont été développées pour retarder l'épuisement des adresses.

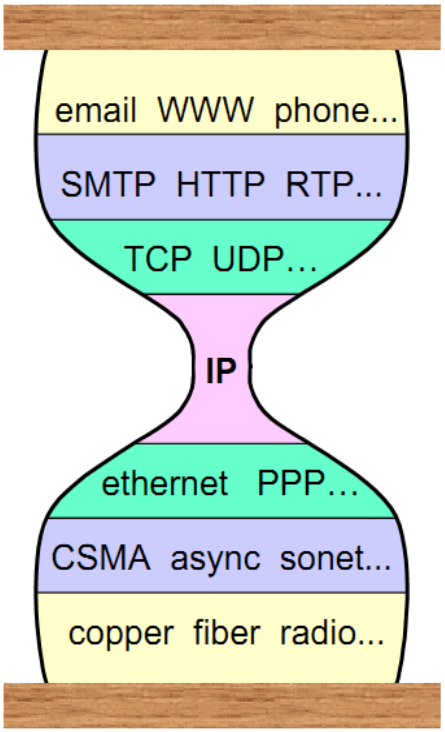


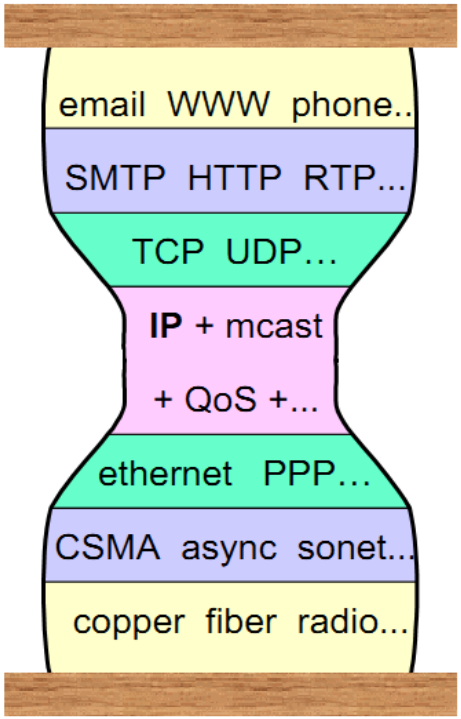
NAT: Une Solution à l'épuisement des Adresses IPv4?

- NAT a été une solution temporaire à l'épuisement des adresses IPv4.
- Il permet l'utilisation d'une seule adresse publique pour connecter plusieurs IP au réseau.
- Inconvénients:
  - NAT introduit des étapes de traitements des données en plus entre les connexions.
  - Les erreurs de NAT ont un impact très négatif sur la communication entre dispositifs.
  - Impacte sur l'extensibilité globale du réseau.
  - Impacte sur le modèle de sécurité.
  - NAT produit des « TCP state violations ».

IPv6 - Architecture

Modèle su sablier établit par Steve Deering (2001) → Permet de montrer qu'IPv6 et basé sur la même architecture qu'IPv4.

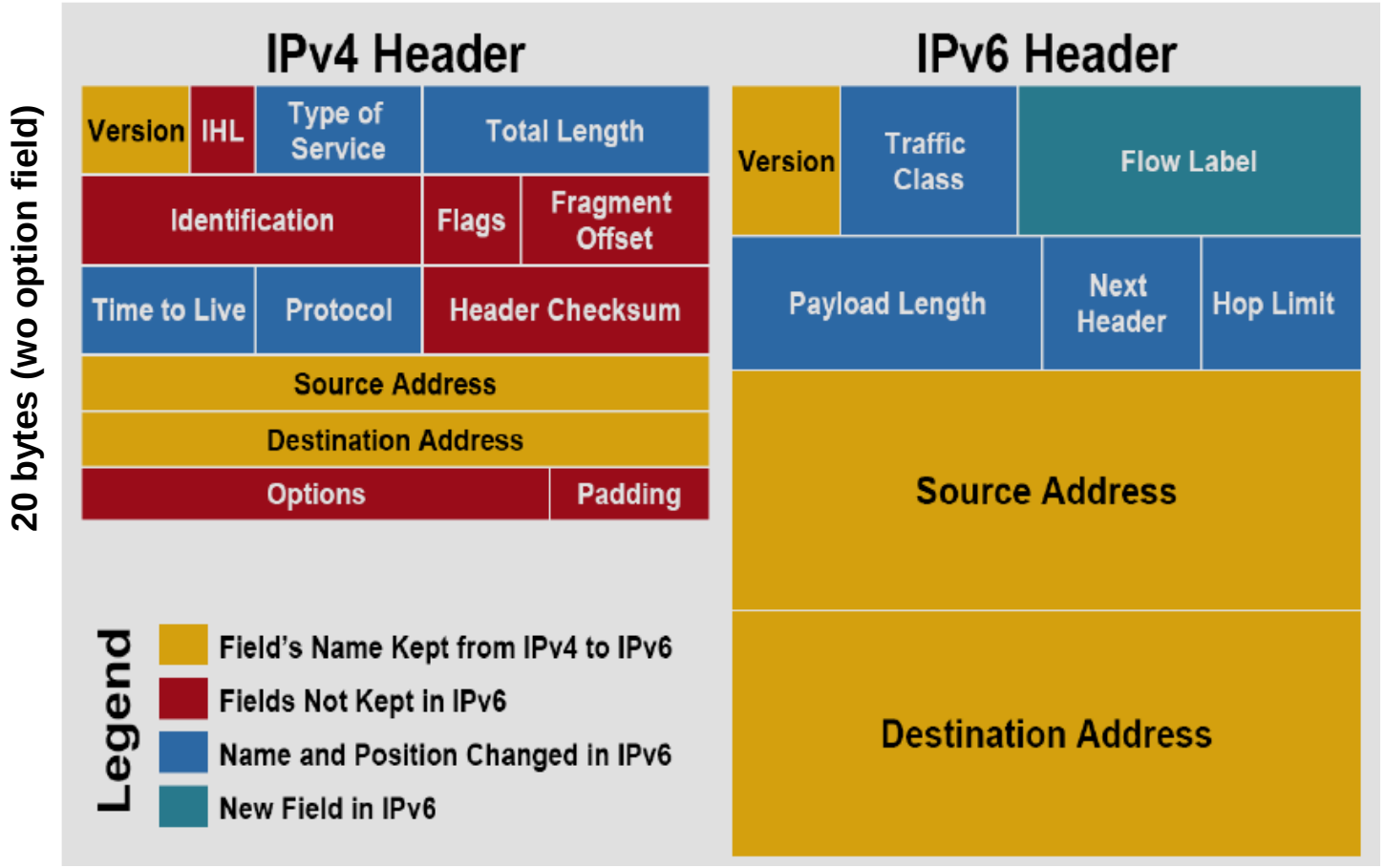




IPv6 pour les Réseaux d'Objets Intelligents

- Fonctionnalités clés de l'IPv6 :
  - **Espace plus grand pour l'adresse (utile pour les réseaux de grande ampleur),** l'augmentation de l'adresse de 32 à 128 bits permet:
    - d'adresser un plus grand nombre de *nodes*
    - d'avoir plus de niveau de hiérarchie
    - faciliter l'autoconfiguration
  - **Autoconfiguration**: ensemble de fonctions (mise à disposition, configuration, gestion d'erreurs, inventaire, analyse de performance) supporté par IPv6 dès sa conception
  - **Changement de *header***: une structure plus simple avec un *header* fixe qui peut être étendu avec des *headers* en chaîne
  - **Authentification et vie privée**: extensions définies pour le support de l'authentification, de la protection des données et de la confidentialité.
  - **Sécurité**: IPSec est obligatoire en IPv6 (optionnel en IPv4).

Entête du Paquet: IPv4 vs IPv6



Ref : Al-Ani, Muzhir & A.A.Haddad, Rola. (2012). IPv4/IPv6 Transition. International Journal of Engineering Science and Technology. 4. 4815-4822.

Entête du Paquet IPv6 – Entête Fixe

- Champs de l'entête:

Version (4 bits): IP version number = 6

Traffic class (8 bits): Class of Service (CoS) du paquet

Flow label (20 bits): permet le marquage d'un flux pour un traitement différencié dans le réseau

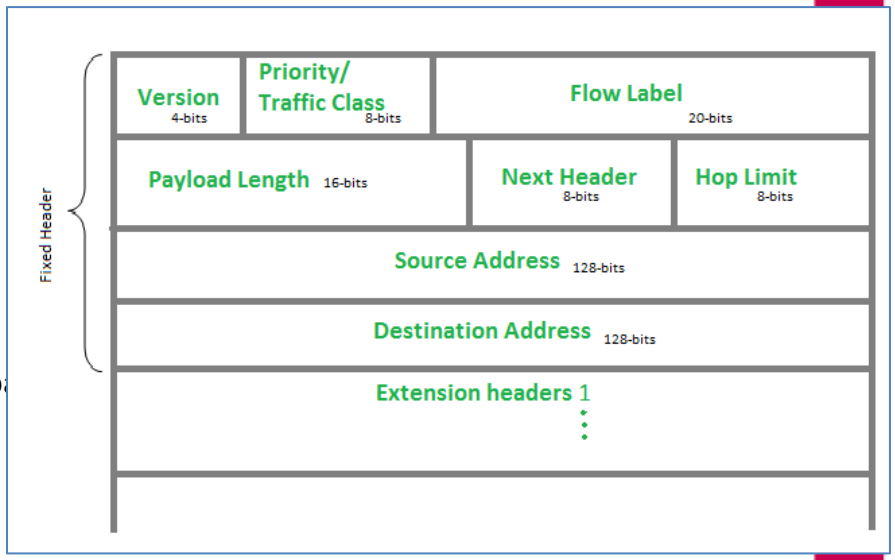
Payload length (16 bits): taille de la charge utile du paquet

Next header (8 bits): identifie le type de header qui suit immédiatement

Hop limit (8 bits): nombre max de saut avant de détruire le p

Source address (128-bits): adresse de la source

Destination address (128-bits): adresse de destination

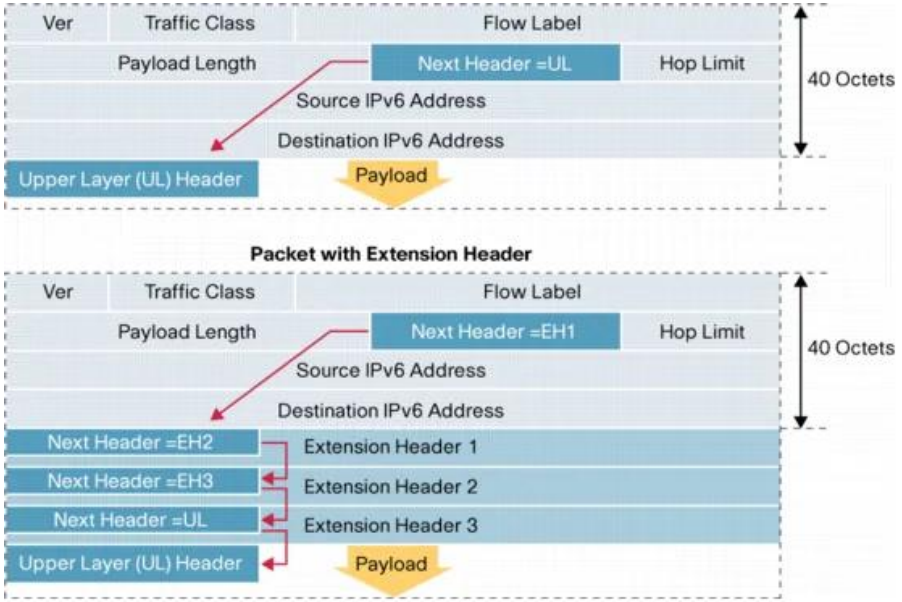


- Le *header* fixe de l'IPv6 fait 40 bytes, celui de l'IPv4 n'en fait que 20 → Le groupe de travail 6LowPAN a spécifié divers schémas de compression de *header*

- Il n'y pas de checksum dans un *header* IPv6 → tous les protocoles de transport permettent d'effectuer un *checksum*

Entête étendue du Paquet sous IPv6

Order	Header Type	Next Header Code
1	**Basic IPv6 Header**	-
2	**Hop-by-Hop Options**	0
3	**Destination Options (with Routing Options)**	60
4	**Routing Header**	43
5	**Fragment Header**	44
6	**Authentication Header**	51
7	**Encapsulation Security Payload Header**	50
8	**Destination Options**	60
9	**Mobility Header**	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58



Les *entêtes* doivent apparaître dans un ordre spécifique et ne sont pas traitées par les routeurs intermédiaires
Toutes les implémentations IPv6 doivent supporter les *entêtes* d'extensions suivants :

Hop-by-hop options: transporte de l'information qui doit être traitée par d'autres routeurs
Routing: identifie un ensemble de nœuds qui doivent être empruntés par le paquet lors de son transit
Fragment: un nœud IPv6 fragmente un paquet chaque fois que sa taille est plus grande que celle de la MTU
Destination options: transporte des infos supplémentaires qui doivent être traitées par le nœud de destination
Authentication
Encapsulating security payload

Ref: https://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html

Architecture de l'adressage IPv6

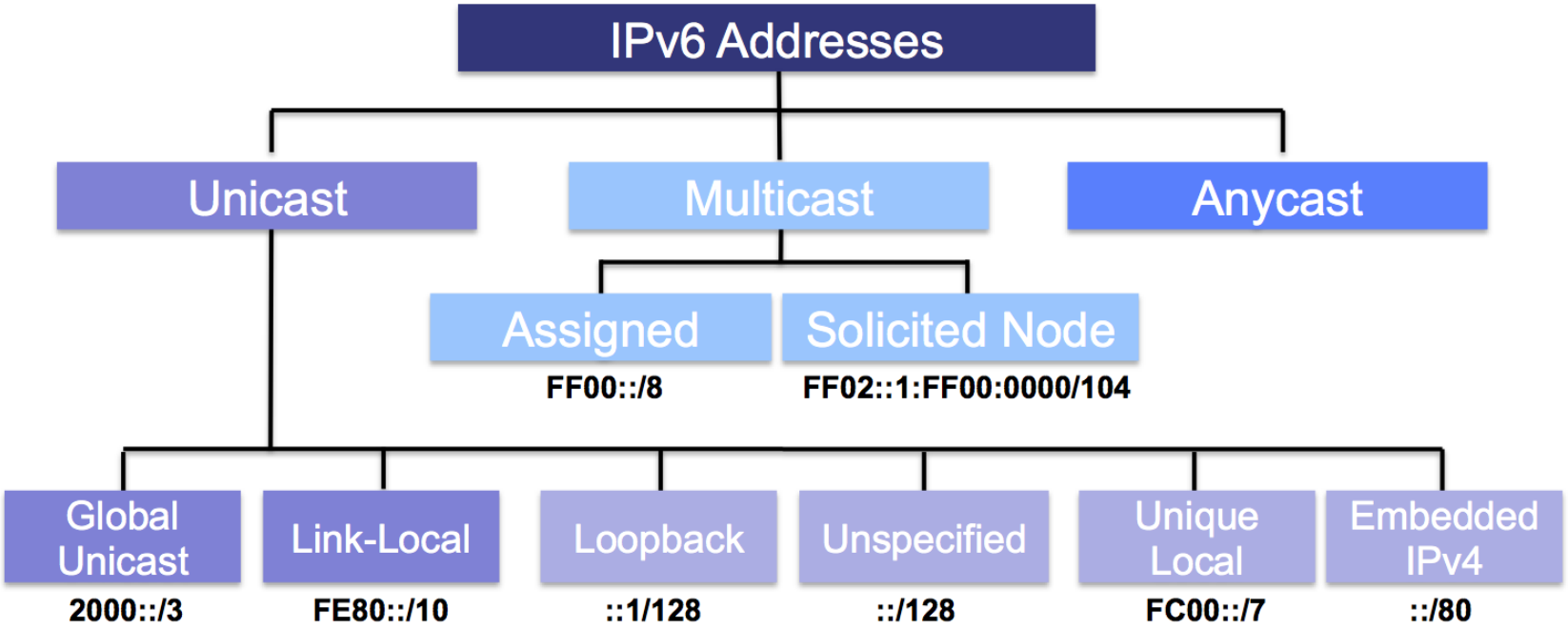
- Adresses de 128 bits → 3.4 10^{38} adresses (4.8 10^{23} adresses par personne sur Terre ou 6.6 10^{23} adresses par mètre carrés).
  - **Unicast address**: identifie de manière unique une interface qui a au moins une adresse locale
  - **Anycast address**: identifie un ensemble d'interfaces, un paquet envoyé à une adresse anycast est seulement délivré à l'une des interfaces de l'ensemble
  - **Multicast address**: un paquet envoyé à une adresse multicast est délivré à toutes les interfaces identifiées par cette adresse. Pas d'envoi broadcast en IPv6, des adresses multicast sont utilisées

- Représentation des adresses IPv6:
  - 32 bit adresse IPv4 : x.y.z.t (par exemple 124.4.12.3) ; une portion représente la partie réseau et le reste la partie hôte.
  - Adresses IPv6 128 bits: **x:x:x:x:x:x:x:x**
    - par exemple 2020:CA28:0000:0000:0023:0222:0000:2900 ≡ 2020:CA28::23:222:0:2900
    - ::1, adresse de boucle retour (équivalent à 127.0.0.1 en IPv4, loopback)
    - :: adresse non spécifiée
  - IPv6 n'impose pas de limite particulière pour la partie réseau.
  - Environnement mixte (IPv4 et IPv6) : 2020:CA28::222:124.4.12.3

Préfixes d'Architecture d'Adressage en IPv6

Allocation	Prefix (binary)	Fraction of address space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP allocation	0000 001	1/128
Reserved for IPX allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Aggregatable global unicast addresses	001x xxxx	1/8
Unassigned	010x xxxx	1/8
Unassigned	011x xxxx	1/8
Unassigned	100x xxxx	1/8
Unassigned	101x xxxx	1/8
Unassigned	110x xxxx	1/8
Unassigned	1110 xxxx	1/16
Unassigned	1111 0xxx	1/32
Unassigned	1111 10xx	1/64
Unassigned	1111 110x	1/128
Unassigned	1111 1110 0	1/512
Link-local unicast addresses	1111 1110 10	1/1024
Site-local unicast addresses	1111 1110 11	1/1024
Multicast addresses	1111 1111	1/256

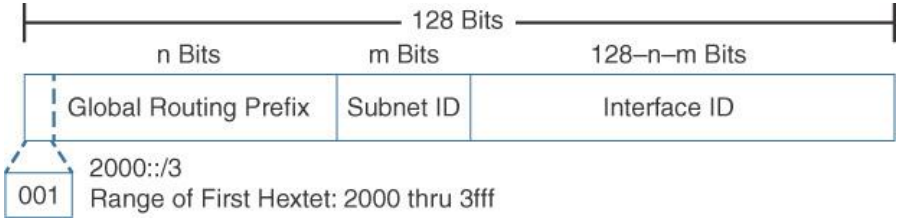
RFC3513: Internet Protocole version 6 (IPv6) Addressing



Ref: http://www.iplogos.fr/ipv6-les-differentes-adresses/

Adressage IPv6 : Unicast

Une adresse unicast est faite de plusieurs préfixes de sous-réseau et d'un identifiant d'interface (généralement : 64bits et 64bits)



Les identifiants d'interfaces sont utilisés pour identifier une interface dans un réseau (unique dans ce réseau et généralement identique à l'adresse dans le réseau de cette interface)

1. **Global unicast**: adresse routable sur "IPv6 Internet", similaire à une adresse publique IPv4
2. **Link-local**: adresse utilisée pour communiquer avec les autres nœuds dans un lien local
3. **Loopback**: adresse de bouclage sur le nœud.
4. **Unspecified address**: utilisée comme adresse source pour spécifié l'absence d'adresse
5. **Unique local**: adresse unique sur le réseau non routable sur "IPv6 Internet"
6. **IPv4 embedded**: adresse IPv6 comprenant l'adresse IPv4 dans les 32 bits les moins significatifs

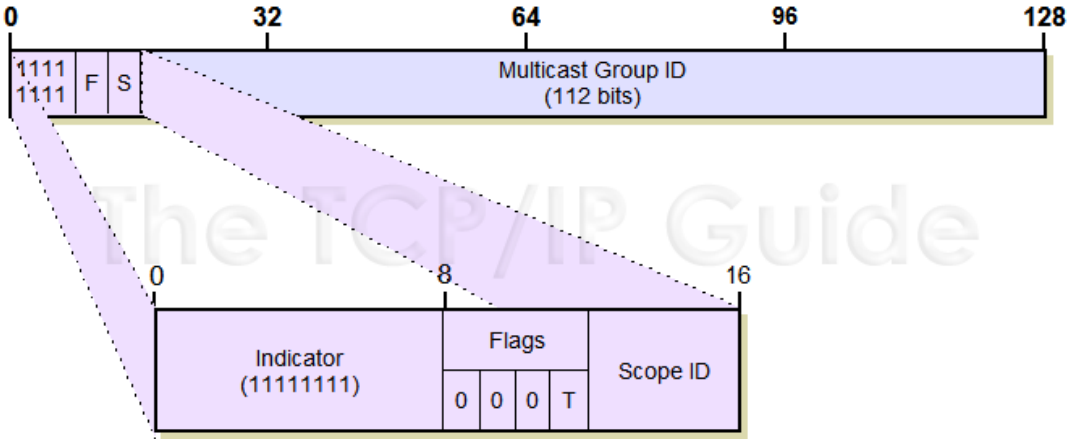
Adressage IPv6 : Anycast

- L'adresse anycast est allouée pour un ensemble d'interfaces qui appartiennent à des routeurs différents
- Quand un paquet est destiné à une adresse anycast, il est délivré à l'interface le plus proche (déterminé par le protocole de routage) possédant cette adresse anycast
- Une adresse anycast doit être assignée à un routeur et non à un hôte et ne peut être utilisée comme une adresse source

- Exemple : l'adresse anycast de sous-réseau :
  - Préfixe de sous-réseau de n bits qui spécifie un réseau suivi de 128-n bits tous à 0
  - Un paquet envoyé à l'adresse anycast de sous-réseau est délivrée à l'un des routeurs sur ce sous-réseau.

Adressage IPv6 : Multicast

- Une adresse multicast identifie un groupe de nœuds (groupe de multicast) et ne doit pas être utilisée comme une adresse source dans un header.
- Format:
  - Commence par FF (8 bits).
  - Suivi d'un champ de flags (**flag field**) de 4 bits : le flag T détermine si l'adresse de multicast est permanente (T=0) ou est une adresse temporaire (T= 1).
  - Suivi par un champ de scope (**scope field**) de 4 bits : scope de l'adresse multicast (réseau local, local-nœud, site local etc.).
  - Se termine avec 112 bits d'identification (**group ID**).



L'ICMP pour IPv6

- Le composant ICMPv6 est un composant clé de l'architecture IPv6
- Les fonctionnalités de l'IPv6 sont étendues avec d'autres fonctionnalités comme l'ARP et l'IGMP
- ICMPv6 identifié par un nouveau type de protocole (**type 58**).

- Structure d'un message ICMPv6:
  - 8-bit type field: type du message
  - 8-bit code field: détails additionnels pour le type de message
  - 16-bit checksum
  - Variable length data field

- Catégories de messages ICMPv6 :
  - **Messages d'erreurs**: nombre compris entre 0 et 127.
  - **Messages d'informations**: nombre compris entre 128 et 255.

Messages d'Erreurs ICMPv6

	Type	Code	Description
Destination unreachable	1	0	No route to destination (no routing entry for the packet). This does not include packet drop error due to congestion.
		1	Communication with destination administratively prohibited (e.g., case of a firewall that cannot forward the packet because of filtering action triggered by policy).
		3	Address unreachable for other reasons.
		4	Port unreachable.
Packet too big	2	0	The packet size exceeds the MTU of the outgoing link.
Time exceeded	3	0	Sent when the hop limit field is equal to 0 or the received packet has a hop limit equal to 0.
Parameter problem (problem with the field of the IPv6 header or extended header)	4	0	Erroneus header field encountered.
		1	Unrecognized next header type encountered.
		2	Unrecognized IPv6 option encountered.

Messages d'Information ICMPv6

	Type	Code
Echo request	128	0
Echo reply	129	0
Router solicitation (RS)	133	
Router advertisement (RA)	134	
Neighbour solicitation (NS)	135	
Neighbour advertisement (NA)	136	
Redirect	137	

Neighbour Discovery Protocole

- Neighbour Discovery Protocol (ND) fournit un ensemble de fonctionnalités d'autoconfiguration :
  - Exploration de la présence de voisins sur le réseau
  - Recherche de *routers* sur la liaison
  - Recherche des adresses des couches liaison du réseau
  - Information sur le maintien de l'accessibilité au voisinage actif
- ND joue un rôle important dans les réseaux objets intelligents IP:
  - Découverte de *routeurs*
  - Recherche du préfixe
  - Recherche des paramètres
  - Adresses d'autoconfiguration
  - Résolution de l'adresse (ARP)
  - Détermination du prochain *hop*
  - Détection d'inaccessibilité du voisinage (NUD)
  - Recherche de duplication d'adresse (DAD)
  - Redirection

Neighbour Discovery Protocole

ND définit 5 nouveaux types de messages ICMP :

- **Neighbour Solicitation** (NS) Message: utilisé pour la résolution d'adresse, détection d'inaccessibilité NUD et détection de duplication d'adresse (DAD)
- **Neighbour Advertisement** (NA) Message: utilisé pour fournir la couche d'adresse du réseau à un nœud qui le demande (réponse à un NS) ou pour informer d'un changement de couche d'adresse
- **Router Advertisement** (RA) Message: utilisé périodiquement par les routeurs pour avertir de leur présence en plus de plusieurs paramètres internet (préfixes de réseaux par exemple)
- **Router Solicitation** (RS) Message: envoyé par un hôte pour avoir un message RA en réponse sans avoir à attendre l'expiration du *timer* périodique du RA
- **Redirect** Message: envoyé par un *routeur* pour informer un hôte d'un meilleur nœud pour arriver à destination

IPv6 Autoconfiguration

- La faculté pour un nœud de supporter l'autoconfiguration est très importante, particulièrement quand le nombre de nœuds est important et les nœuds non utilisés.

  → L'ensemble des fonctions d'autoconfiguration supporté par l'IPv6 est particulièrement adapté aux réseaux d'objets intelligents

- Construire l'adresse réseau local:
  - Quand une interface est initialisée, le nœud fabrique son adresse locale en utilisant le préfixe de réseau local FE80::0/10 suivi de 54 bits à 0 et l'identifiant de l'interface
  - A ce stade, le nœud peut communiquer avec n'importe quel autre nœud dans le même réseau (les paquets ne sont pas transmis sur d'autres réseaux par les routeurs)

- Processus d'autoconfiguration sans état:
  - Permets à un nœud de générer ses adresses de réseau local, de site local et global en utilisant une combinaison d'informations locales et d'informations fournies par les routeurs sans configuration sur l'hôte, une configuration minime du routeur et sans serveur externe (en contraste avec les configurations fournies par le protocole DHCPv6).

IPv6 Quality of Service

- L'objectif des mécanismes de QoS est d'attribuer différentes priorités au trafic → fournir un traitement différent aux paquets selon leur **Class of Service** (CoS). Similaire en IPv4 et IPv6.
- Modèle Diffserv:
  - Premier pas : le trafic est marqué à la frontière du réseau → définition du champ de **Traffic Class** (TC) du *header* IPv6. Le champ TC contient :
    - **Diffserv Code Point** (DCP): 6 bits, définit le comportement par hop (PHB)
    - **Explicit Congestion Notification** (ECN): 2 bits, marque explicitement la présence d'une congestion
  - Chaque *router* sur le chemin peut traiter le paquet en conséquence en utilisant le **PHB** défini pour leur CoS. Deux catégories de mécanismes :
    - **Gestion du trafic**: Utilisation de mécanismes de file d'attente
    - **Évitement de congestions**: quand les files d'attente se remplissent, plusieurs mécanismes peuvent être utilisés pour commencer à éliminer des paquets en utilisant une approche statistique
- Modèle IntServ:
  - Le modèle de service intégré repose sur des mécanismes de réservation des ressources dans le réseau pour des flux critiques
  - Son utilisation dans les réseaux d'objets intelligents est improbable dans un futur proche

IPv6 sur un réseau à squelette IPv4

- Bien que la plupart des réseaux migreront probablement d'IPv4 à IPv6 dans les 3 à 5 prochaines années, l'IPv4 restera utilisé dans beaucoup de réseaux dans les décennies à venir

- Solutions possibles:
  - Mécanismes de *tunneling* : nécessite quelques configurations sur les routeurs aux frontières du réseau et n'offrira pas toujours le chemin optimal
  - Avoir 2 piles sur les routeurs et les hôtes du réseau

- Exemple d'implémentation: **6to4**
  - Utilise des tunnels dynamiques pour interconnecter des ilots IPv6 à un réseau IPv4

Simplifier les adresses suivantes :

- fe80:0000:0000:0000:0000: 4cff:fe4f:4f50
- 2001:0688:1f80:2000:0203:ffff:0018:ef1e
- 2001:0688:1f80:0000:0203:ffff:4c18:00e0
- 3cd0:0000:0000:0000:0000:0040:0000:0000:0cf0
- 0000:0000:0000:0000:0000:0000:0000:0000
- 0000:0000:0000:0000:0000:0000:0000:0001

Réponses

- fe80::4cff:fe4f:4f50
- 2001:688:1f80:2000:203:ffff:18:ef1e
- 2001:688:1f80::203:ffff:4c18:e0
- 3cd0::40:0:0:cf0
- :: (adresse indéterminée)
- ::1 (adresse de bouclage)

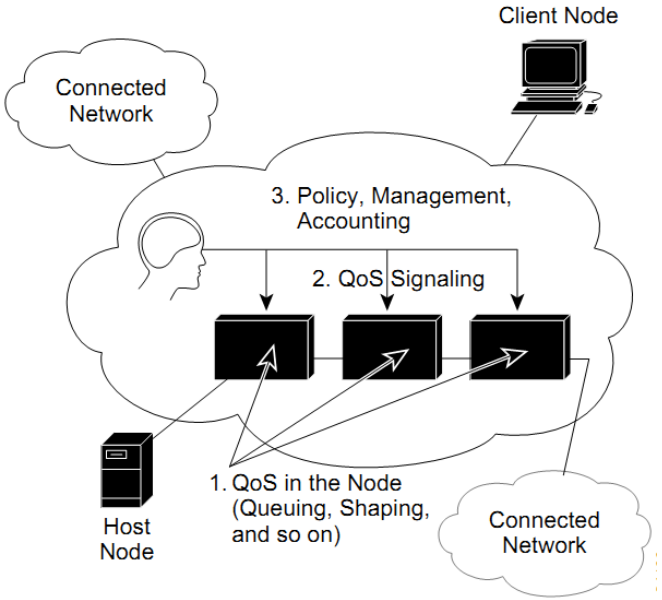
ROUTAGE

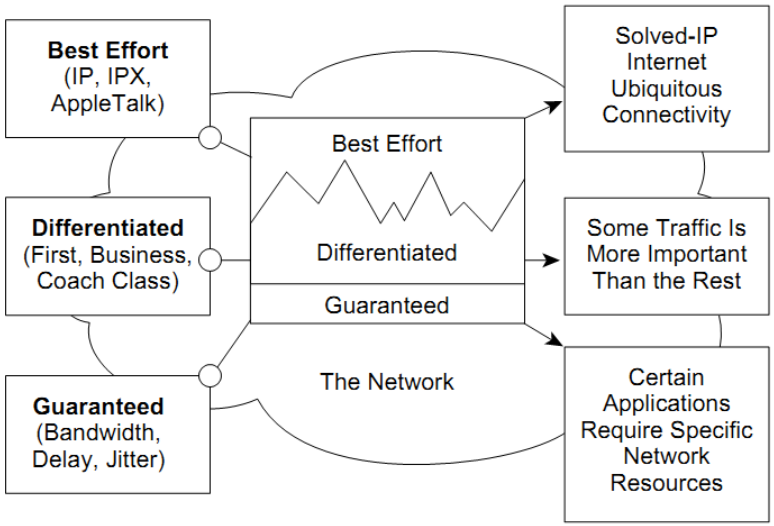
Routage dans les Réseau IP

- **Protocole de routage** → détermine le “meilleur” chemin (e.g., nombre minimum de sauts, coûts minimums, etc.) pour atteindre la destination selon les différents paramètres et objectifs.
- **Tables de routage**, récurrent dans les routeurs, elles indiquent le meilleur chemin jusqu’au(x) prochain(s) saut(s) pour chaque destination atteignable.
- Protocoles de routage:
  - Intra-domaine: e.g., RIP, IS-IS, OSPF, OLSR, AODV
  - Inter-domaine: e.g., BGP

Routage IP et QoS

- **Qualité de service (Quality of Service, QoS**): capacité de réunir certains critères pour le trafic tels que les retards de réseau et gigue (jitter) ou la probabilité de perte de paquets.
- La combinaison du protocole de routage et des mécanismes de QoS le long du chemin de transmission détermine le niveau de QoS fournit au trafic selon ses CoS (Class of service)





Routage IP et Fiabilité de Réseau

- Fonctions principales d'un protocole de routage→ Trouver un chemin alternatif lorsqu'une erreur de lien ou de nœud intervient (**rerouting**).
- Temps pour trouver un chemin alternatif: temps de convergence (**convergence time)**.
- Nouveaux protocoles et technique de détection d'erreur, telle que le protocole de détection de transfert bidirectionnel (BFD).

- Reroutage à couches multiples doit être effectué de façon synchronisée (par ex., IP sur le réseau optique).

- Un autre mécanisme bien connu: envoyer deux fois le même paquet en s'assurant d'utiliser un chemin différent pour chaque paquet.

Routage dans LLNs

- **LLNs** (Low-power and Lossy Networks): réseaux construits à base des objets intelligents aux ressources réduits et interconnectés au travers des couches liaison instables et à basse vitesse → nouvelles contraintes et défis sur les protocoles de routages.

Internet actuel	Réseaux de faible puissance et avec beaucoup de perte
Les nœuds sont des routeurs	Les nœuds sont des capteurs / actionneurs et des routeurs
IGP avec des centaines de nœuds	Un ordre de grandeur en termes de nombre de nœuds
Les liens et les noeuds sont stables	Les liens sont très instables et les nœuds disparaissent beaucoup plus souvent
Contraintes des nœuds ou bandes passantes de liaison sont généralement « non-issues »	Nœuds / liens sont fortement limitées
Le routage n'est pas sensible aux applications	Un routage dépendant de l'application est nécessaire

- Un nouveau protocole de routage IP nommé RPL a été conçu pour fonctionner avec l'ensemble des contraintes des réseaux des objets intelligents.

PROTOCOLES DE TRANSPORT

Protocoles de Transport

- Les applications n'utilisent pas directement IP pour communiquer entre elles, elles utilisent un protocole de transport.

- Les protocoles de transport les plus utilisés sont:
  - **UDP** (**User Datagram Protocol**):
    - Transmission selon la règle du Best-effort (le mieux possible)
    - Unité de transport: datagramme.
  - **TCP** (**Transmission Control Protocol**):
    - Flux de donnée fiable.
    - Unité de transport : segments.

User Datagram Protocol

- Protocole le plus simple de la suite TCP/IP (RCF768)
- Transmission des datagrammes selon la règle du Best-effort.
- Beaucoup d'applications IP fonctionnent avec UDP (exemple: audio ou donnée temps réel, donnée sensible au temps…), protocoles de requête-réponse simple (DNS,…)

- En-tête d'UDP:
  - **Source port**: numéro de port du processus qui a envoyé le datagramme.
  - **Destination port**: numéro de port du processus qui doit recevoir le datagramme.
  - **Length**: longueur, en byte, de la donnée qui suit l'en-tête.
  - **Checksum**: checksum Internet (16-bit) de la donnée du datagramme, de l'en-tête UDP, et des adresses IP source et destination.

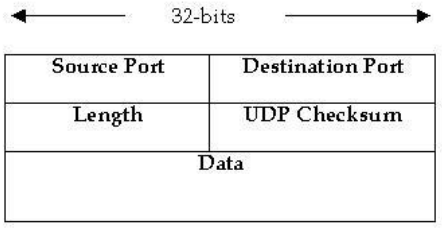


Figure 8: UDP segment structure

Transmission Control Protocol

- Permet l'échange de donnée fiable, le meilleur des services "best-effort" fourni par IP.

- La fiabilité est assurée par la mise en mémoire tampon des données, ainsi que les accusés de réception positifs (ACK) et les retransmissions.
- Mécanismes pour atteindre une **transmission fiable** orientée octet (byte-oriented)
  - Acknowledgments et retransmissions: Toutes les données envoyées par TCP sont acquiescé par le récepteur. S'il ne reçoit pas l'ACK dans un certain laps de temps, il renvoie les données.

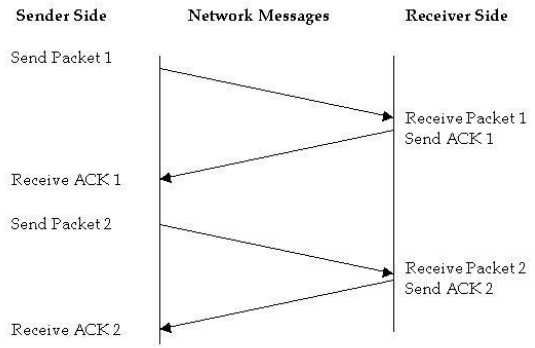


Figure 9

Transmission Control Protocol

– Fenêtre glissante: le récepteur averti du nombre d'octets qu'il est capable de recevoir et l'émetteur envoie seulement le nombre d'octets possible. Dès que le récepteur a reçu une donnée, il est capable d'en recevoir d'autres.

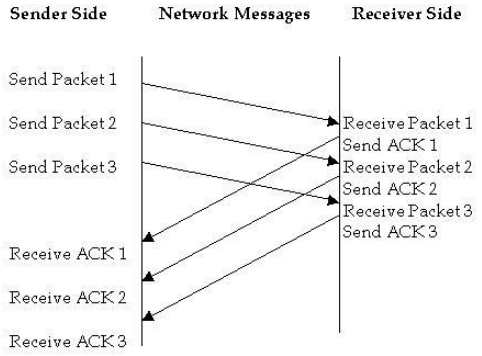
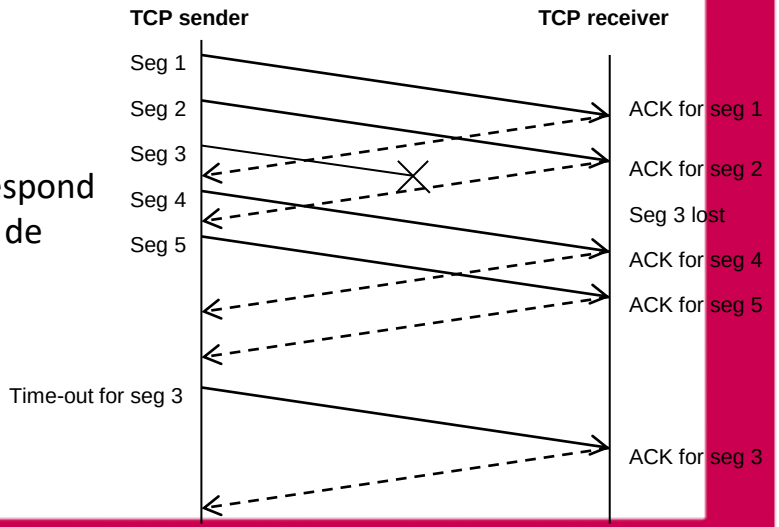


Figure 10

– Numéro de séquence: un numéro de séquence correspond à chaque octet dans le flux de donnée TCP, il permet de faire correspondre chaque ACK avec sa donnée.



Transmission Control Protocol

- TCP header:
  - **Source port** (16-bit): numéro du port du processus d'envoi.
  - **Destination port** (16-bit): numéro du port du processus de réception.
  - **Sequence number** (32-bit): numéro de séquence du premier octet de la donnée contenue dans le segment.
  - **Acknowledgment number**: Si le flag ACK est ON, le numéro de l'ACK (acknowledgment number) contient le numéro de séquence (32bits) du prochain octet que le récepteur attente.
  - **Hlen** (4-bit): longueur de l'en-tête (options incluses) divisé par quatre.
  - **Flags** (6-bit): six flags: FIN, SYN, RST, PSH, ACK and URG.
  - **Window** (16-bit): nombre d'octets que le récepteur peut recevoir.
  - **Checksum** (16-bit): checksum Internet de la donnée, de l'en-tête TCP, et des IP source et destination.
  - **Urgent pointer**: Si le flag URG est ON, il indique le flux d'octets qui contient les données dont l'application a défini comme urgents.

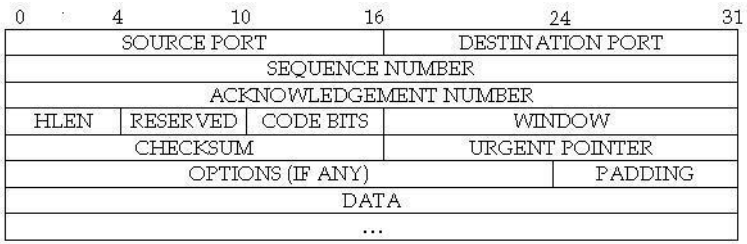


Figure 11

Transmission Control Protocol

- Options TCP :
  - Information de contrôle supplémentaire entre l'en-tête TCP et les segments de donnée.
  - Exemple: accusé de réception sélectif (SACK) et extension TCP pour réseau à haut débit.
  - Option importante pour les objets intelligents → **Maximum segment size** (MSS): plus grande taille de segment possible, il est envoyé par les deux parties lors de l'ouverture d'une connexion.
- Round-trip time estimation:
  - Temps d'attente d'un ACK avant la retransmission du segment.
  - Mesure du temps aller-retour, prise lors de chaque fenêtre.
- Flow control:
  - Assure que l'émetteur ne surchargera pas le récepteur.
  - L'espace de mémoire tampon disponible pour une connexion→ **window of the connection**.
- Congestion control:
  - Essayer d'éviter le dépassement de mémoire tampon du routeur
  - Deux mécanismes:
    - **Slow start**: Détecte la bande passante disponible lors d'un envoi.
    - **Congestion avoidance**: adapte la vitesse d'envoi en fonction de la bande passante.

UDP pour les Objets Intelligents

- Avantages:
  - Faible surcharge pour l'en-tête et la logique du protocole →l'émission et la réception consomme moins d'énergie et chaque paquet a plus de place pour les données de la couche application.
  - Simplicité→ idéale pour les systèmes avec peu de mémoire.
  - Convient à beaucoup d'application qui demande un rapport périodique peu fiable (exemple: domotique, capteur de température…)
  - Aussi bien adapté pour des applications qui requièrent leur propre mécanisme de routage.
  - Pour les applications qui doivent utiliser une transmission par multidiffusion (multicast).

- Inconvénients:
  - UDP ne fournit aucun mécanisme de récupération des paquets perdus.
  - UDP ne fournit aucun mécanisme pour fragmenter les données en paquets de taille appropriée pour la transmission.

TCP pour les Objets Intelligents

- Avantages:
  - Des réseaux d'objets intelligents fonctionnent sur des liaisons où les paquets peuvent être perdus → TCP offre un mécanisme fiable qui retransmet automatiquement les paquets perdus.
  - Les réseaux d'objet intelligent doivent souvent collaborer avec des systèmes existants où TCP est très largement utilisé → TCP offre la possibilité de communiquer directement avec les systèmes existants.
  - Les réseaux d'objet intelligent ont souvent des petites tailles de paquet→L'option MSS de TCP est très utile pour les systèmes à faible mémoire et les systèmes limités par la taille des paquets.
  - La base de TCP est très simple, sans ses mécanismes complexes pour améliorer les performances à haut débit→ assez simple pour être implémenté sur des objets intelligents à faible mémoire (exemple: uIP).

- Inconvénients:
  - Grande taille de l'en-tête
  - La logique du protocole de la couche transport est assez complexe

Conclusions