Université de Strasbourg

Internet Protocol version 6

Julien Montavont

montavont@unistra.fr

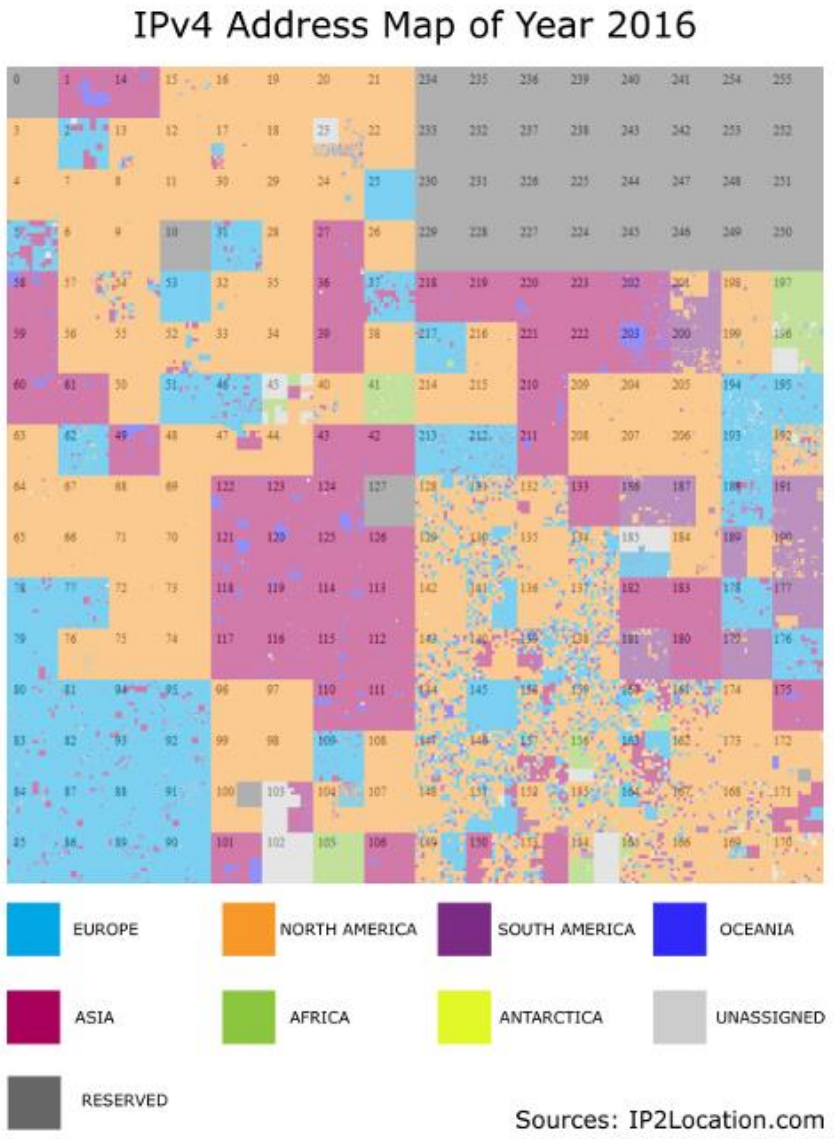
Licence 2 Informatique

Internet - État des lieux

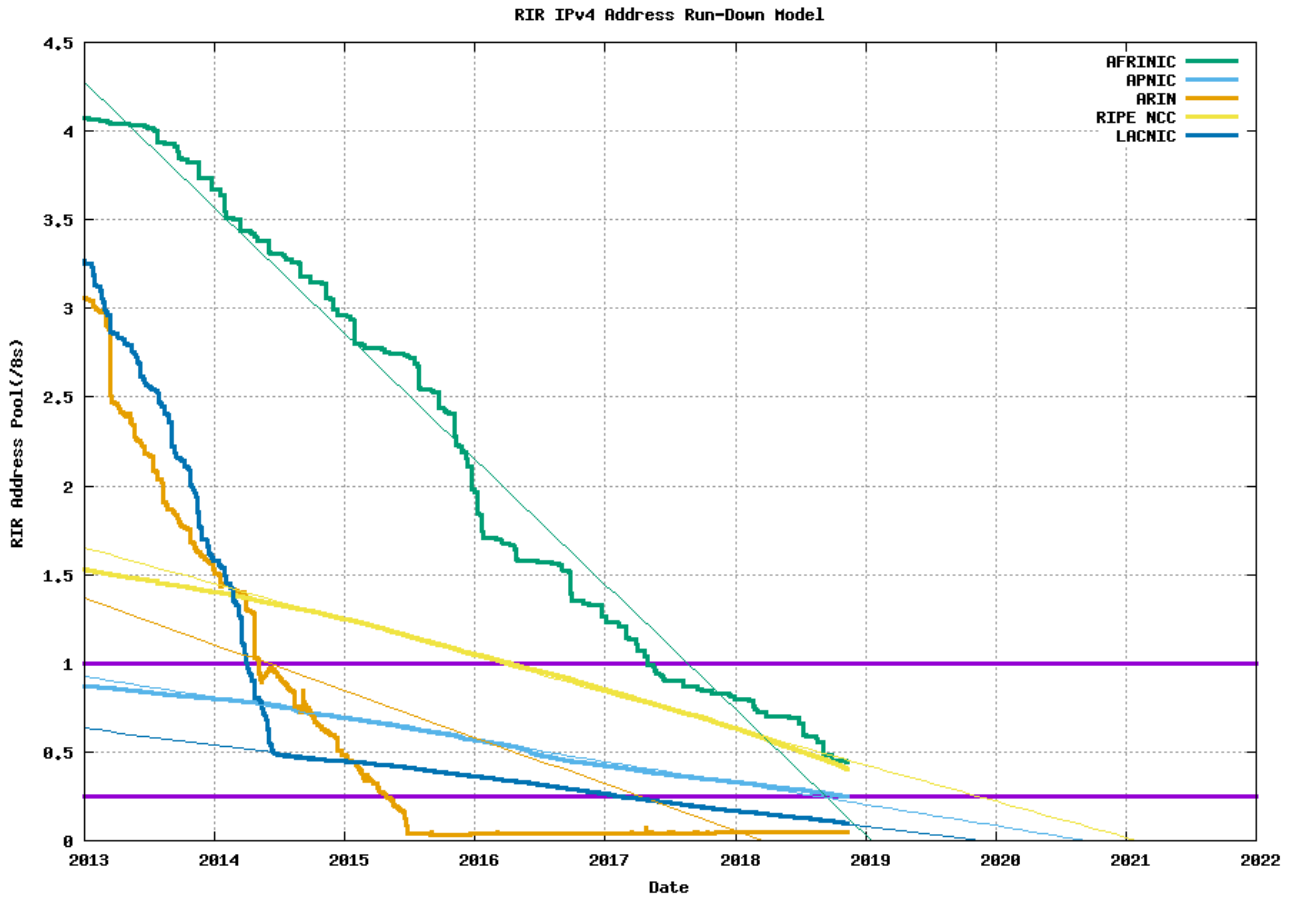
- 1973 : Réseau pour la recherche (environ 100 postes connectés)
- 1992 : ouverture à l'activité commerciale
  - Croissance exponentielle
- 1993 : épuisement de la classe B
- Prévision d'écroulement du réseau pour 1994

Constat

- Manque d'adresses
  - Vatican City : 921 habitants / 2560 adresses IP
- Explosion des entrées dans les tables de routage
- Manque de souplesse / flexibilité
- Expérience de plus de 20 ans sur IPv4
- Nouveaux besoins



Projection des prefixes /8 disponibles au sein des RIR



Source : potaroo.net

Mesures d'urgence

- Utilisation exceptionnelle des classes B
- Réutilisation de l'espace d'adressage de classe C
- Adoption du modèle Classless Internet Domain Routing (CIDR)
  - Adresse réseaux = préfixe + longueur du préfixe
  - Moins de gaspillage de l'espace d'adressage (préfixe au plus juste)
  - Agrégation => réduction du nombre d'entrées dans les tables de routage

Mesures d'urgence

- Adressage privé
  - Adresses non routable sur Internet (ex : 192.168.0.0/16)
- Interconnexion de réseaux privés
  - Réseau privé caché derrière une (ou plusieurs) adresse publique
  - Nouvel équipement intermédiaire qui fait la relation entre adresse(s) publique(s) et adresses privées
    - Network Address Translation (NAT)

NAT

- ✓ Réduits les besoins en adresses publiques
- ✓ Facilite le plan d'adressage interne
- ✓ (Presque) transparent aux applications
- ✓ Sécurité ?
- × Translation peut être complexe (e.g. FTP)
- × Ne passe pas à l'echelle
- × Introduit des états dans le réseau
  - × Quid du redémarrage ?
  - × Changement des tables de routage ?
- × Sécurité ?

Bilan mesures d'urgence

- Gagner du temps pour développer une nouvelle version du protocole IP
  - Garder ce qui fait le succès d'Internet (e.g. modèle ***Best Effort***)
  - Corriger ce qui ne fonctionne pas avec IPv4
- Débat :
  - Les mesures d'urgence ne sont-elles pas suffisantes ?
  - A-t-on réellement besoin d'une nouvelle version d'IP ?

Pourquoi un nouveau protocole IP ?

- **Deux problèmes à résoudre**
  - Épuisement de l'espace d'adressage publique
    - ***Nouveau protocole doit permettre d'adresser un espace beaucoup plus grand***
  - Explosion du nombre d'entrées dans les tables de routage
    - ***Nouveau protocole doit proposer un routage plus efficace***

De IPv4 à IPv6

- Le rôle de l'IETF et de l'IESG
  - Livre blanc (RFC 1550)
- Plusieurs propositions (21 réponses)
  - CATNIP (Common Architecture for the Internet) RFC 1707
  - SIPP (Simple Internet Protocol Plus) RFC 1770
  - TUBA (TCP and UDP with Bigger Address)
- Technical Criteria for Choosing IP The Next Generation – RFC 1726 (décembre 1994)
- The Recommandation for the IP Next Generation Protocol – RFC 1752 (janvier 1995)

Caractéristiques d'IPv6

- Défini en 1998 dans RFC 2460
  - Adresses sur 128 bits (versus 32 bits pour IPv4)
    - 667 millions de milliards d'adresses par millimètre carré de surface terrestre
  - Adressage hiérarchique (abandonné aujourd'hui)
  - Partie de l'adresse basée sur adresse IEEE802 (e.g. adresse MAC Ethernet)
  - 3 types d'adresses
    - Unicast
    - Multicast
    - Anycast

Plus de broadcast !!

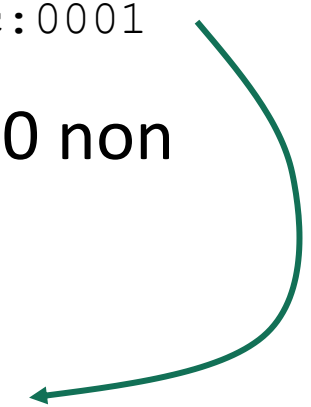
Représentation des adresses IPv6

- Adresses sur 16 octets / 128 bits
- Représentation hexadécimale
- Séparation de chaque 2 octets par « : »
  - Ex :

    `2001:0db8:dead:beef:0000:face:b00c:0001`

- Écriture « simplifiée » en omettant les 0 non significatifs dans chaque 2 octets
  - Ex :

    `2001:db8:dead:beef::face:b00c:1`



Préfixe IPv6

- Notion de préfixe identique à CIDR en IPv4
- Notation sous la forme
  - **Adresse IPv6** / **longueur préfixe**
  - Ex :

`3fffe:302:12::`/**`48`**

Adressage IPv6

- Préfixe unicast ou anycast global (routable)

  `2000::/3`

- Préfixe lien-local
  - Pour dialoguer sur le lien (i.e. ne passe pas les routeurs)

  `fe80::/10`

- Préfixe multicast

  `ff00::/8`

- Préfixe adresses uniques locales (ne passe pas les routeurs)

  `fc00::/7`

- Adresse boucle locale

  `::1/128`

- Adresse non spécifiée

  `::/128`

Allocation d'adresses IPv6

- Gestion des adresses déléguée à
  - Internet Assigned Numbers Authority (IANA)
  - Internet Architecture Board
  - Internet Engineering Steering Group
- Seul 1/8 de l'espace d'adressage IPv6 est alloué pour une utilisation sur Internet
  - Préfixe `2000::/3`
- Reste de l'espace est réservé pour un usage futur

Allocation d'adresses IPv6

IANA

Blocs /12 à /23

Regional Internet Registries (RIR)

Blocs /19 à /32

Local Internet Registries (LIR)

Blocs /48 à /56

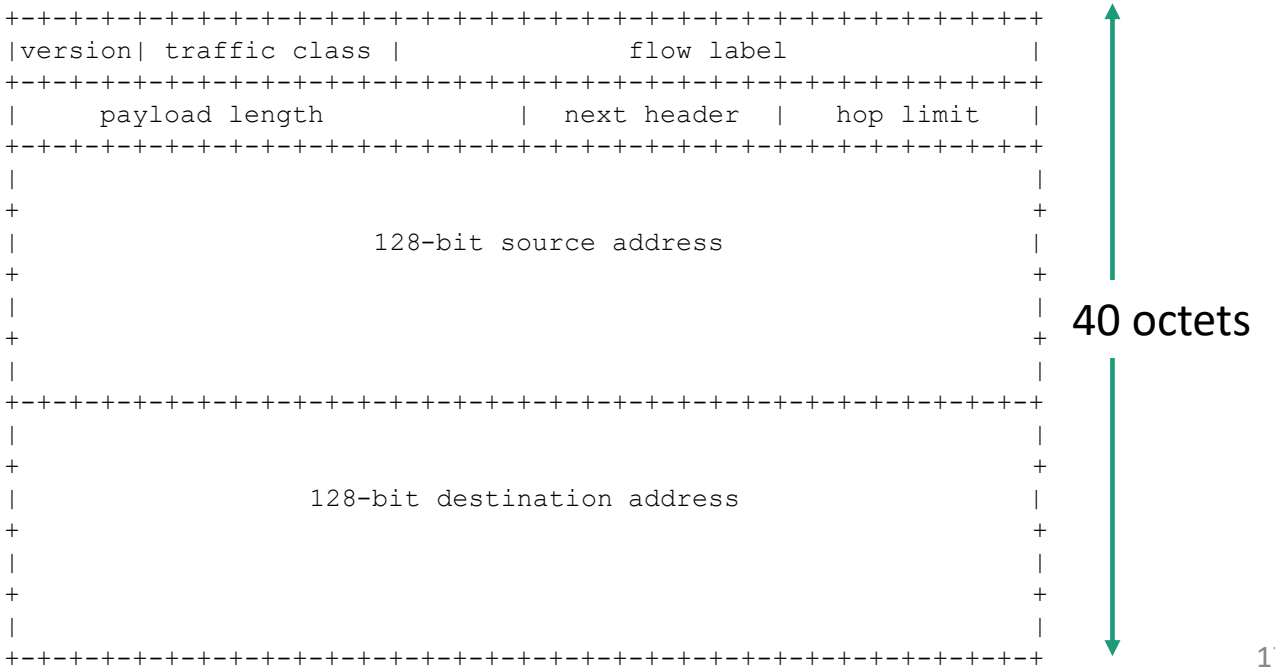
End users

Ex :
RIR peut couper un **/23** en 512 **/32** (1 pour chaque ISP)
ISP peut couper un **/32** en 65 535 **/48**
End user peut couper un **/48** en 65 535 **/64**

En-tête simplifiée

- Moins de champs (environ moitié moins que dans IPv4)
- En-tête de taille fixe (40 octets)
  - Suppression des options - Quid de fonctionnalités futures ?
  - Lecture et traitement plus rapide de l'en-tête par les routeurs

⇒ commutation plus rapide



En-tête simplifiée

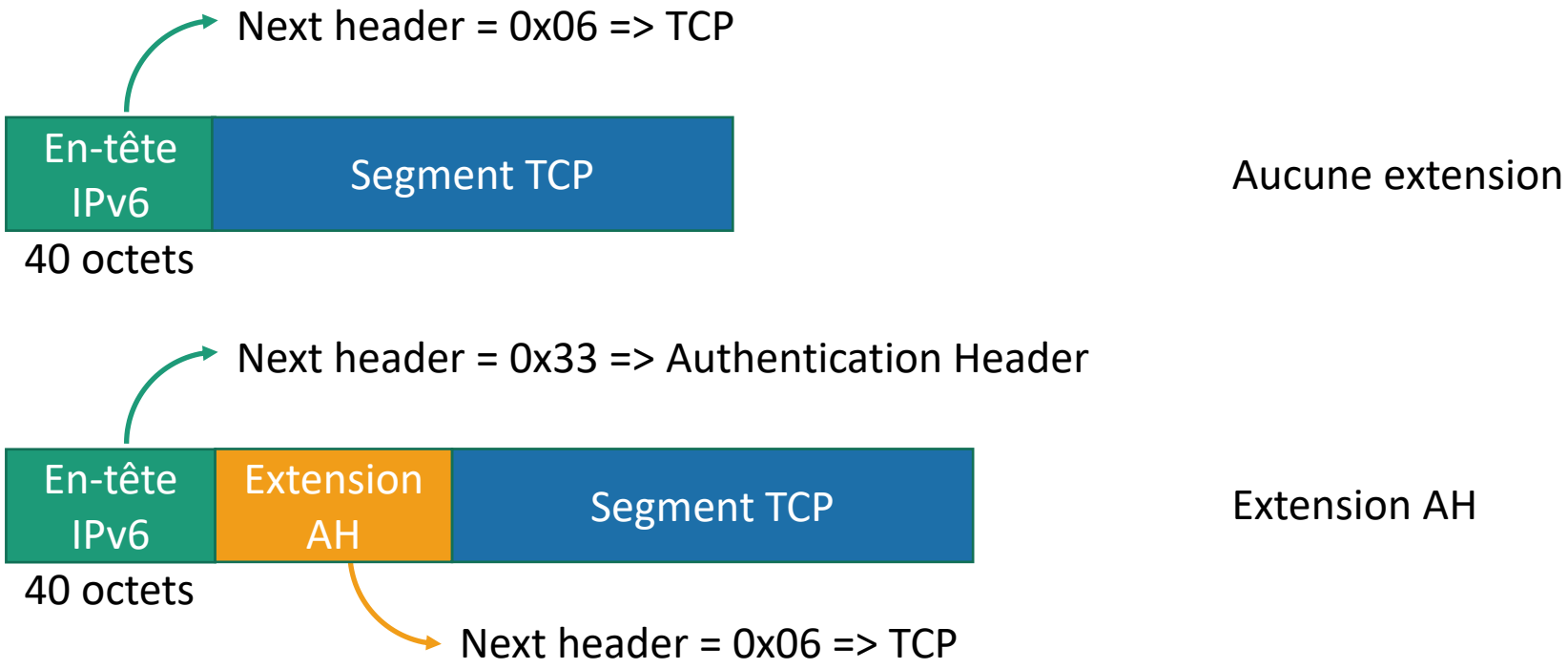
- **Version** (4 bits) : 6
- **Traffic class** (8 bits) + Flow label (20 bits) : réservés pour mécanismes de qualité de service
- **Payload length** (16 bits) : taille des données véhiculées (en octets)
- **Next header** (8 bits) : code d'identification de ce que contient les données (TCP, UDP, autre)
- **Hop limit** (8 bits) : durée de vie du paquet en nombre de sauts

Extensions d'en-tête

- Extensions intercalées entre l'en-tête IPv6 et l'en-tête de niveau transport
- Permet l'ajout de nouvelles fonctionnalités
- Ex :
  - **Fragment** (type 44)
    - Si paquet doit être fragmenté
  - **Authentication Header** (type 51)
    - Permet d'authentifier l'en-tête IPv6
  - **Encapsulating Security Payload** (type 50)
    - Permet de chiffrer les données véhiculées par le paquet
  - …

Extensions d'en-tête

- Ex :



Nouvelles fonctionnalités

- Auto-configuration
  - Concept ***plug and play***
  - Renumérotation simplifiée (i.e. changement de préfixe)
  - Gestion du nomadisme
  - 2 moyens :
    - Serveur de configurations + protocole d'interaction ***Dynamic Host Configuration Protocol*** – DHCPv6 RFC3315
    - ***Stateless Address Autoconfiguration*** – RFC 4862

Auto-configuration sans état

- Étape 1 : création d'un identifiant unique sur 64 bits
  - ***Extented Unique Identifier*** (EUI-64)
  - Basé sur l'adresse MAC de l'interface
  - Dès qu'interface réseau est active (connectivité physique)

Ex :

`00:1b:21:5e:8a:36` adresse MAC de l'interface sur 48 bits

`001b:21`**`ff:fe`**`5e:8a36` ajout du motif **`fffe`** pour passer à 64 bits

`021b:21ff:fe5e:8a36` changement du 2e bit de poids faible du premier octet

Auto-configuration sans état

- Étape 2 : création d'une adresse lien-local
  - Association du préfixe **`fe80::/10`** avec l'EUI-64
  - Ex : `fe80::021b:21ff:fe5e:8a36`
  - vérification de l'unicité de l'adresse ainsi créée (*Duplicate Address Detection*)
  - Si unique, assignation de l'adresse à l'interface et le poste peut l'utiliser

Auto-configuration sans état

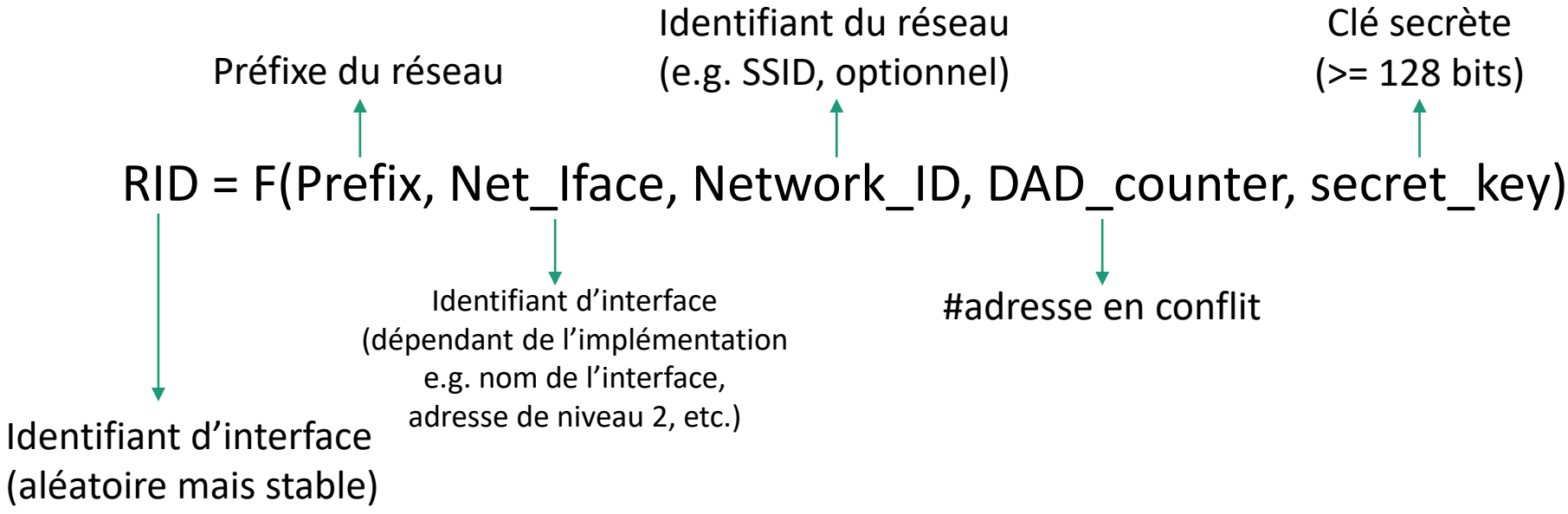
- Étape 3 : création d'une adresse globale
  - Recherche d'un routeur sur le lien pour obtenir le ou les préfixes globaux
    - Recherche active : envoi de messages ***ICMPv6 Router Solicitation*** pour déclencher l'émission de messages **ICMPv6 Router Advertisements**
    - Recherche passive : attente de réception d'un message ICMPv6 Router Advertisements envoyé périodiquement par les routeurs du lien
  - Association du préfixe global avec l'EUI-64
  - Ex : `2001:660:4701:1001:21b:21ff:fe5e:8a36`
  - vérification de l'unicité de l'adresse ainsi créée (*Duplicate Address Detection*)
  - Si unique, assignation de l'adresse à l'interface et le poste peut l'utiliser

Auto-configuration sans état

- EUI-64 d'une interface ne change jamais indépendamment du préfixe réseau
  - Problème de sécurité et de violation de la vie privée (RFC 7721)
    - **Suivi de la localisation**
    - **Corrélation d'activités**
    - **Scan d'adresses**
    - **Exploitation de vulnérabilité**
- **Extension pour la vie privée** (RFC 4941)
  - Création d'adresses temporaires
    - Identifiant d'interface généré (pseudo) aléatoirement par le système d'exploitation
  - Adresses temporaires valides pour une courte période (heures / jours) avant d'être dépréciées
    - Toujours conservées sur l'interface
  - **Complexifie l'administration du réseau (e.g. règle de par-feu)**
  - **SLAAC (RFC 4862) toujours présent**

Auto-configuration sans état

- **Identifiant d'interface stable et opaque** (RFC 7217)
  - Identifiant reste stable pour un même préfixe et change si préfixe change
  - Utilisée par défaut pour SLAAC



Auto-configuration sans état

- F() est fonction pseudo-aléatoire qui génère au moins 64 bits
  - Peut être implémentée comme une fonction de hachage
    - SHA-1 ou SHA-256 [FIPS-SHS]
    - MD5 (acceptable) – RFC1321
- Sur Linux (dans /proc/sys/net/ipv6/conf/XX)

addr_gen_mode – INTEGER
Defines how link-local and autoconf addresses are generated.
0: generate address based on EUI64 (default)
1: do no generate a link-local address, use EUI64 for addresses generated from autoconf
2: generate stable privacy addresses, using the secret from stable_secret (RFC7217)
3: generate stable privacy addresses, using a random secret if unset

stable_secret - IPv6 address
This IPv6 address will be used as a secret to generate IPv6 addresses for link-local addresses and autoconfigured ones. All addresses generated after setting this secret will be stable privacy ones by default. This can be changed via the addrgenmode ip-link. conf/default/stable_secret is used as the secret for the namespace, the interface specific ones can overwrite that. Writes to conf/all/stable_secret are refused.

It is recommended to generate this secret during installation of a system and keep it stable after that.
By default the stable secret is unset.

[FIPS-SHS] NIST, "Secure Hash Standard (SHS)", FIPS Publication 180-4, March 2012, <http://csrc.nist.gov/publications/ fips/fips180-4/fips-180-4.pdf>.

Neighbor Discovery

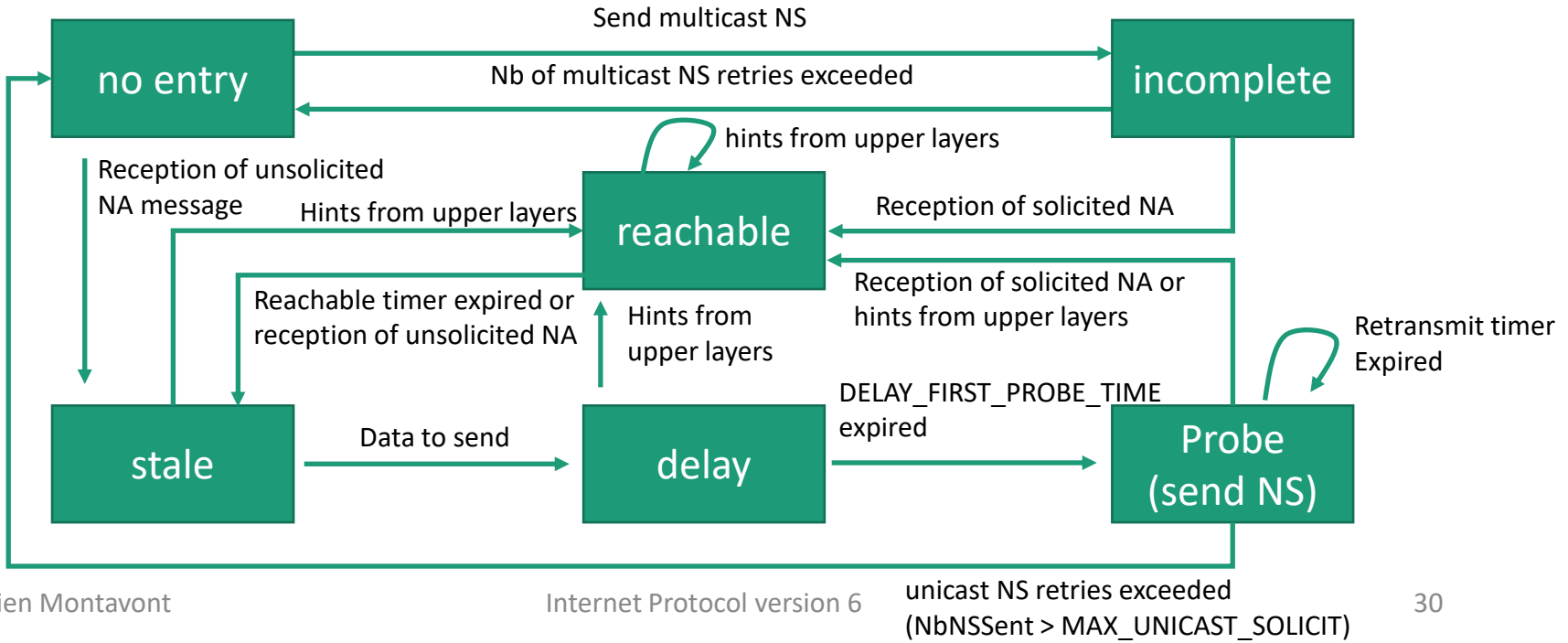
- RFC 1970, RFC 2461, RFC 4861 (~97 pages)
- Cœur du fonctionnement d'IPv6
- Fonctionnalités :
  - Router discovery
  - Prefix discovery
  - Parameters discovery (link MTU, Max hop limit, etc.)
  - Address autoconfiguration
  - Address resolution
  - Next hop determination
  - Neighbor unreachability detection
  - Duplicate adresse detection
  - Redirect

Neighbor Discovery

- 5 nouveaux messages ICMPv6
  - **Neighbor solicitation** (NS) / **Advertisement** (NA)
    - Résolution d'adresses, détection de duplication d'adresses, vérification si un voisin est toujours joignable
  - **Router Solicitation** (RS) / **Advertisement** (RA)
    - Découverte des routeurs et diffusion des paramètres du lien
  - **Redirect**
    - Pousse de nouvelles entrées dans les tables de routage des hôtes

Neighbor Unreachability Detection

- Neighbor cache
  - Équivalent du cache ARP en IPv4
  - Chaque entrée dispose d'un statut
    - Reachable, Stale, Delay, Probe, Incomplete



Duplicate Address Detection

- Vérification de l'unicité d'une adresse sur un lien
  - Ex : **`fe80::a00:27ff:fee5:1234`**
- Étape 1 : Hôte s'abonne aux groupes multicasts suivants (via message **MLD Report**)
  - All-nodes multicast address **`ff02::1`**
  - Solicited-node multicast address **`ff02::1:ffe5:1234`**

Préfixe `ff02::1:ff00:0/104` + 24 derniers bits de l'adresse qu'on cherche à vérifier

Duplicate Address Detection

- Étape 2 : envoi de ***X*** **Neighbor Solicitation**
  - *X* étant configurable
  - Chaque NS est séparé par *RetransTimer* (configurable)

    Src = :: (unspecified address)
    Dst = ff02::1:ffe5:1234
    Target address = fe80:a00:27ff:fee5:1234

  - Si réception d'un Neighbor Advertisement, alors l'adresse réclamée est déjà attribuée
    - OS génère un nouvel identifiant aléatoire et recommance la procédure
  - Si pas de réponse après *X* envois => adresse unique sur le lien

Router discovery

- Réception d'un **Routeur Advertisement** permet (généralement) d'obtenir les paramètres suivants
  - *Router lifetime* – durée de vie du routeur en secondes
  - *Reachable time* – durée recommandée (en millisecondes) pour l'état ***reachable*** dans le neighbor cache
  - *Retrans timer* – durée recommandée (en millisecondes) pour laquelle les hôtes doivent espacer les NS en cas de non réponse par un NA
  - Adresse de niveau 2 du routeur
  - MTU du lien
  - Préfixe(s) du lien
    - *Flag on-link* – indique si les adresses associées à ce préfixe sont sur le même lien
    - *Flag autonomous address-configuration* – indique si le préfixe annoncé peut être utilisé pour construire une adresse
    - *Prefix length*
    - *Valid lifetime* – durée en secondes durant laquelle ce préfixe est considéré comme valide
    - *Preferred lifetime* – durée en secondes durant laquelle une adresse de ce prefixe est considérée comme préférée

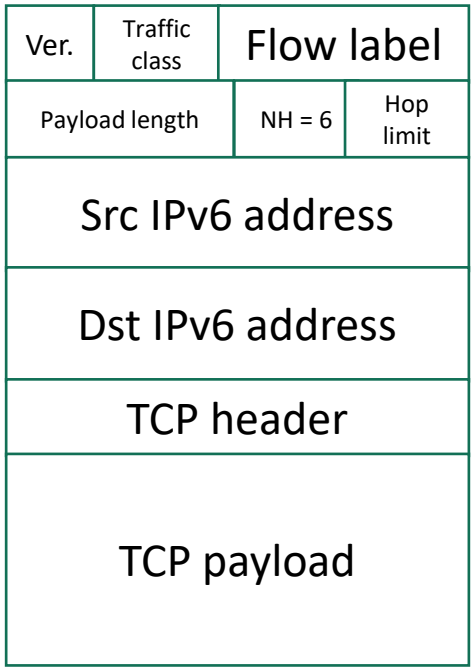
Internet Protocol Security (IPsec)

- Sécurité des communications TCP/IP
  - Niveau applicatif (Pretty Good Privacy – **PGP**)
  - Niveau transport (Transport Layer Security – **TLS** / Secure Socket Layer – **SSL**, Secure Shell – **SSH**)
- IPsec – standard pour assurer des communications privées et protégées au niveau 3
  - Norme prévue pour IPv6
  - Confidentialité, authentification, intégrité et anti-rejeu
  - Plusieurs méthodes d'authentification
    - Clefs partagées, certificats, externes (**radius**, **kerberos**)
  - Plusieurs méthodes de chiffrement
  - Négociation automatique via protocole Internet Key Exchange (**IKE**)

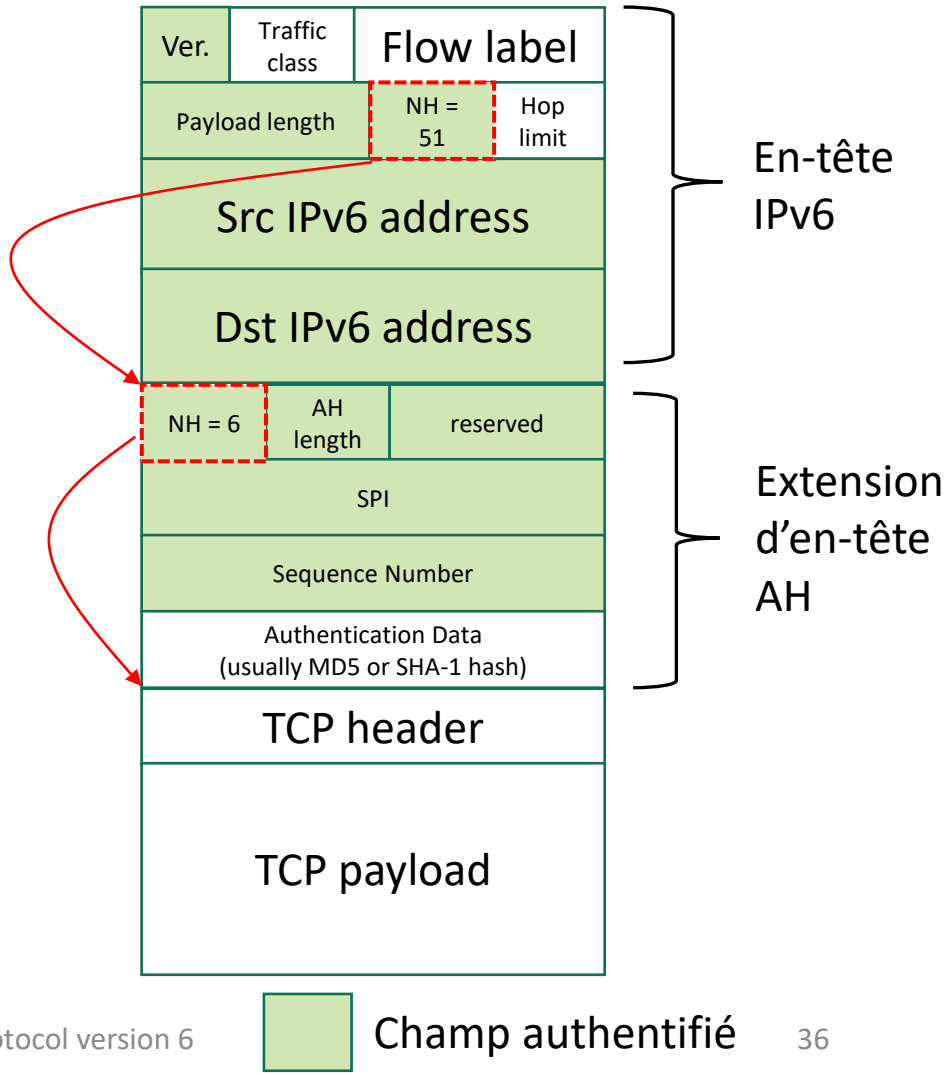
Internet Protocol Security (IPsec)

- **Transport mode**
  - Connexion sécurisée entre deux hôtes
  - Encapsulation des données IP
  - Ne traverse pas les NAT (car adresses IP protégées)
- **Tunnel mode**
  - Connexion sécurisée entre deux routeurs
  - Encapsulation complète du paquet IP
  - Utilisé pour créer des Virtual Private Network (VPN)
    - D'autres technologies / méthodes existent
  - Traverse les NAT (avec ESP, impossible avec AH)

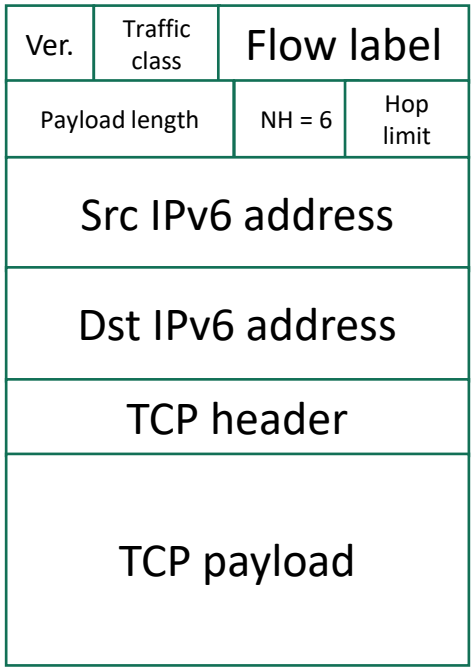
IPsec : Authentication Header



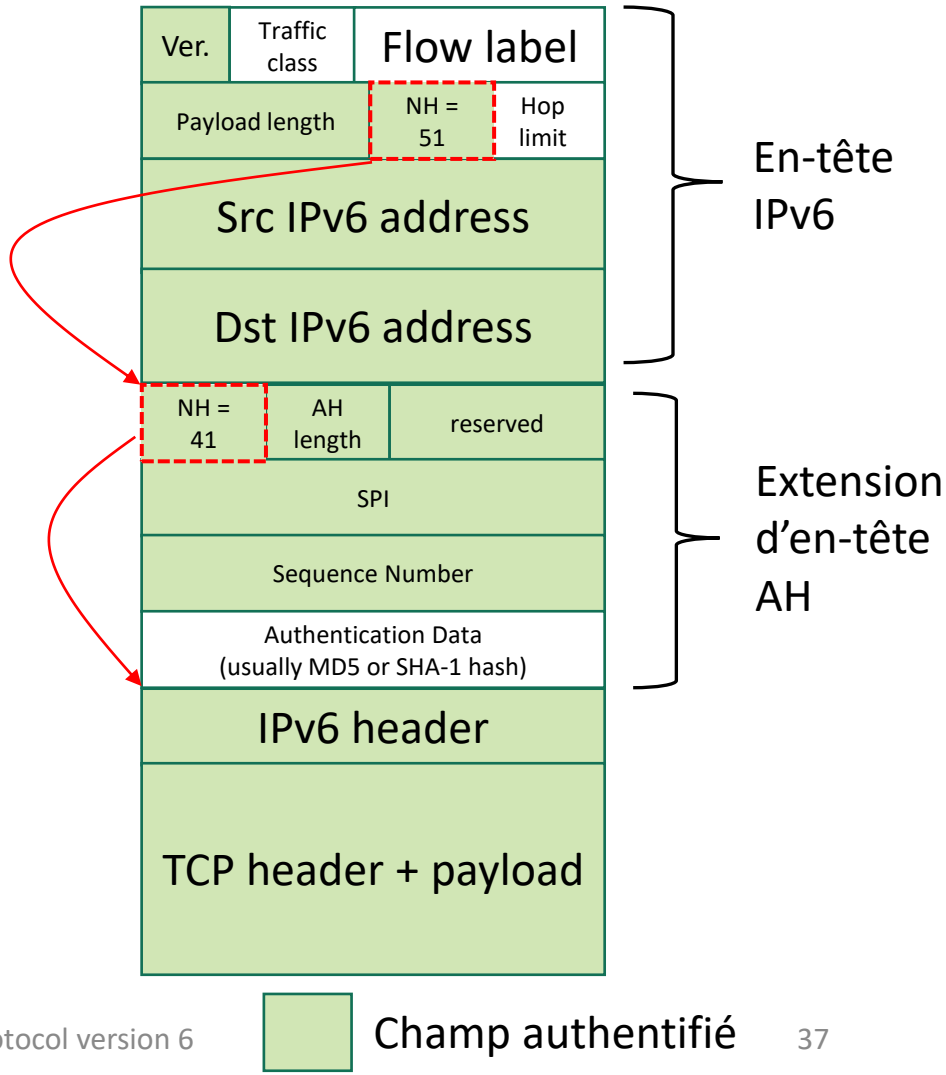
IPsec in AH transport mode



IPsec : Authentication Header

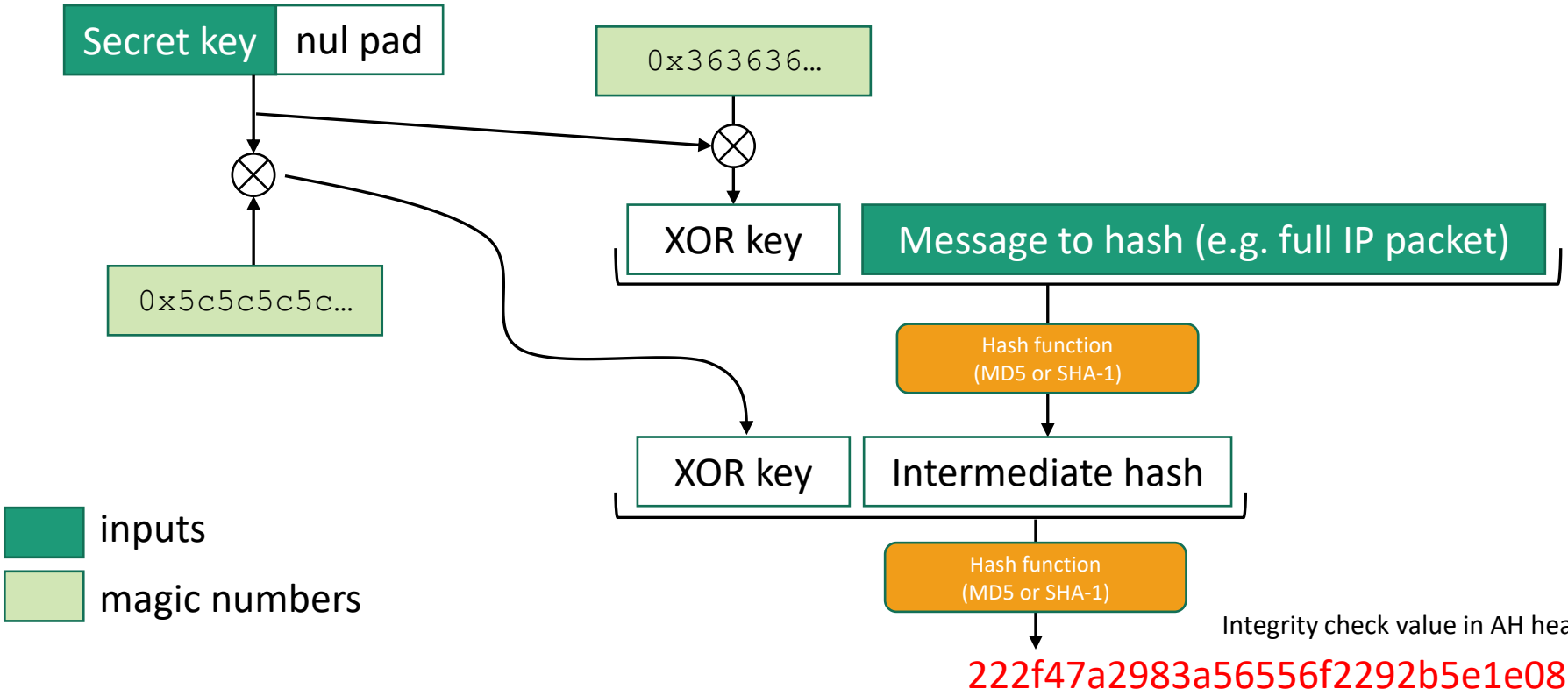


IPsec in AH tunnel mode

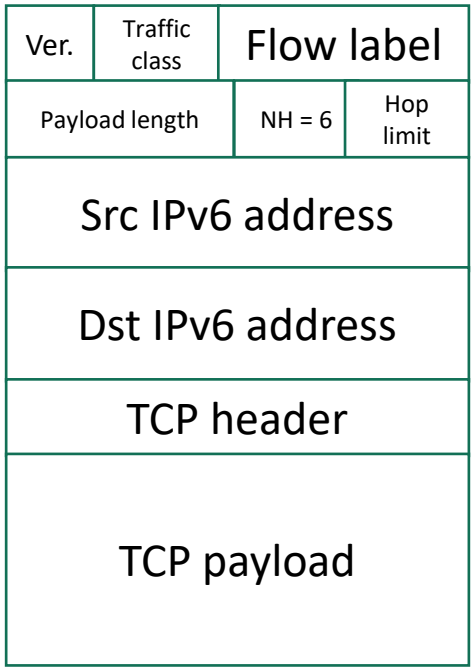


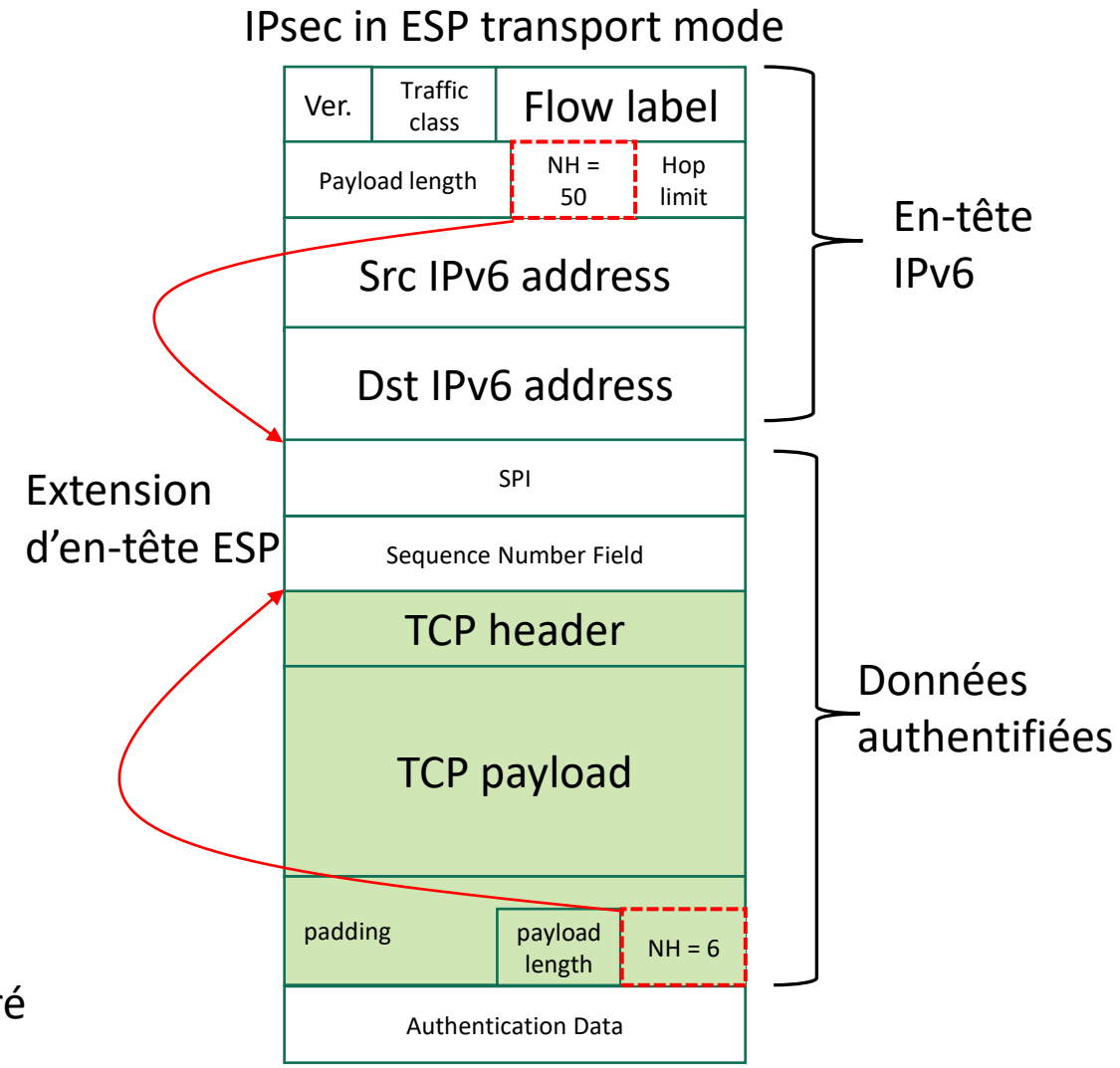
IPsec – algo d'authentification

- Empreintes cryptographiques (hash code)
  - Message Digest 5 (MD5)
  - Secure Hash Algorithm (SHA-1)

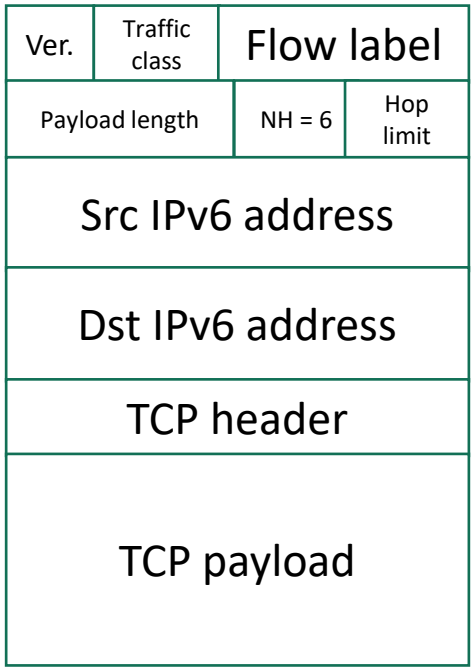


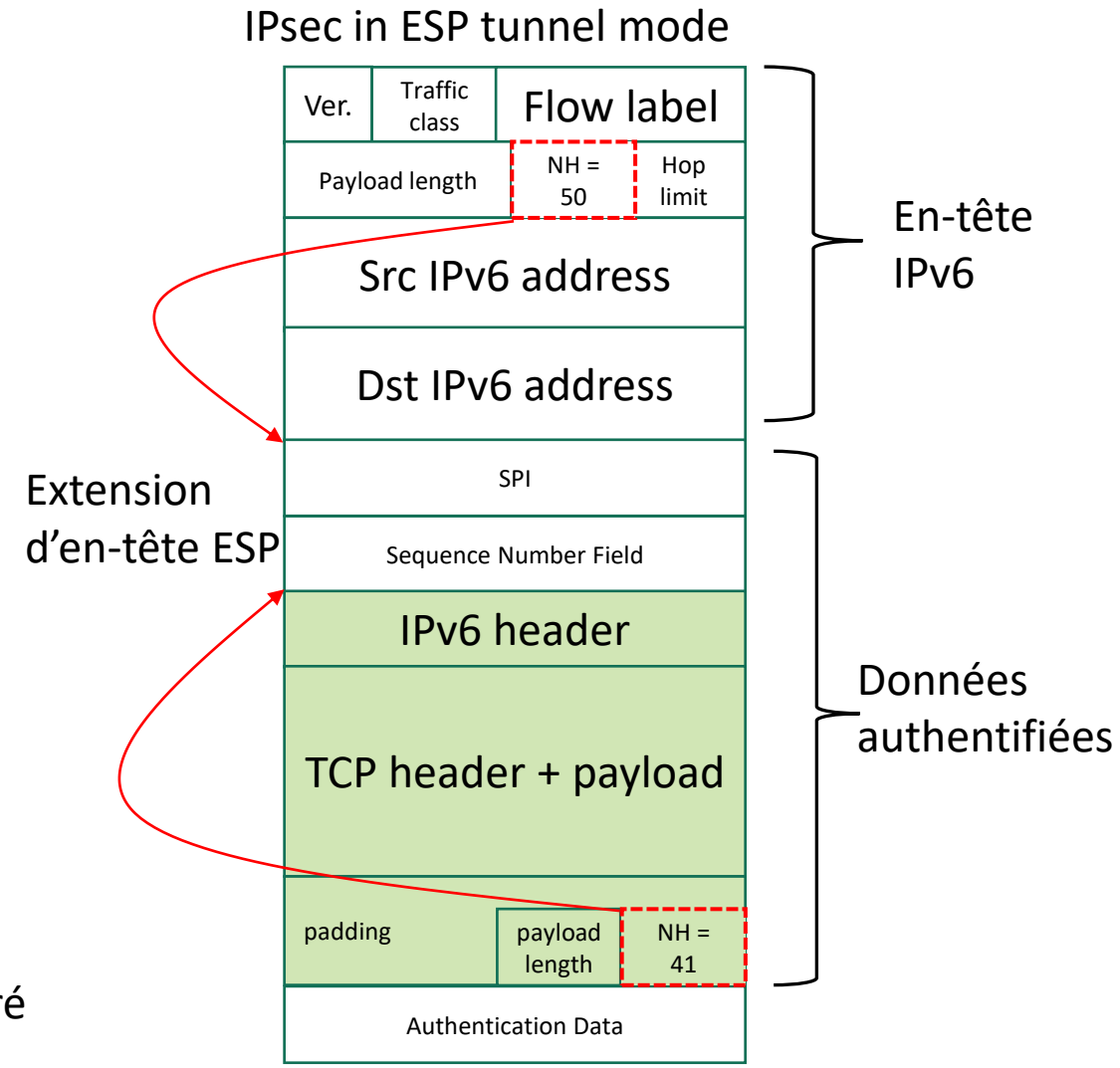
IPsec : Encapsulating Security Payload





IPsec : Encapsulating Security Payload





Champ chiffré

Conclusions

- Nouveau protocole IPv6 (1998)
  - Résout les problèmes liés à IPv4
  - Simplification + nouvelles fonctionnalités
- Adoption lente => pas de *killer application* IPv6
  - 2011 – google déploie IPv6 pour ses services
  - 2014 – 3,6% des utilisateurs google utilisent IPv6
  - 2018 – 24,65% des utilisateurs google utilisent IPv6

    https://www.google.com/intl/en/ipv6/statistics.html
- Pb de la poule et de l'œuf
  - Pas d'opérateur => pas de transition
  - Pas de clients => pas d'opérateur

Migration « douce »

- IPv4 et IPv6 ne sont pas compatibles
- Traducteurs de protocole
  - Ex : Réseau (NAT-PT, NAT64), Transport (TRT RFC 3142), Applicatif (DNS-ALG RFC 2766)
  - Pb d'échelle, pas d'état global, pb de routage…
- Double pile IPv4/IPv6
  - 1ère étape – attribution d'adresses IPv4 et IPv6 aux hôtes
    - Îles IPv6 connectées par des tunnels IPv4
  - 2e étape – généralisation de la double pile
  - 3e étape – abandon progressif d'IPv4

Tunnel v4/v6

- Service de tunnel (tunnel broker) pour fournir une connectivité IPv6
  - Paquets IPv6 encapsulés dans des paquets IPv4
  - Ex : SixXS, Freenet6, Hurricane Electric, BT IPv6
- Tunnels automatiques
  - 6to4 (RFC 3056)
    - Préfixe IPv6 réservé (`2002::/16`)
    - Paquets IPv6 encapsulés dans des paquets IPv4
    - Ne traverse pas les NAT
  - Teredo (RFC 4380)
    - Paquet IPv6 encapsulés dans des datagrammes UDP
    - Traverse les NAT
  - 6rd (RFC 5569)
    - Proche de 6to4 (mais préfixe IPv6 spécifique à l'opérateur et non le préfixe réservé)
    - Utilisé chez le FAI Free