Advanced Networking

Laurent Toutain

October 11, 2012

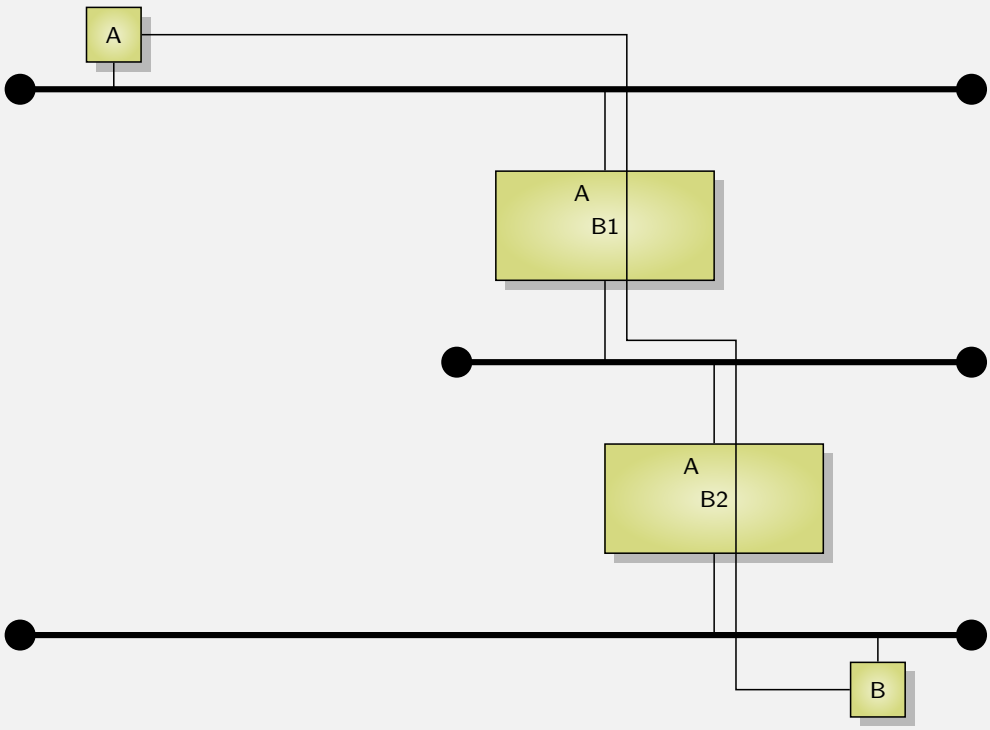
Table of Contents

1. Spanning Tree
   - Bridge loops
   - Algorithm
   - Rapid STP
2. VLAN
   - Introduction
   - IEEE 802.1p/Q
3. Metropolitan Networks
   - QinQ
   - Mac In Mac
4. Concepts
   - Datagram
5. Facts on Addresses
   - Historical view
   - Emergency Measures
   - NAT
   - Prefixes delegation
6. Addresses
   - Notation
   - Addressing scheme
   - Address Format
   - Kind of addresses
7. Protocol
   - IPv6 Header
   - IPv6 Header
   - IPv6 Extensions
   - ICMPv6
   - Impact on Layers 4
8. Associated Protocols & Mechanisms
   - Neighbor Discovery
   - Non-Broadcast Multiple Access (NBMA) Networks
   - Path MTU discovery
   - Examples
   - Neighbor Discovery Security
   - DHCPv6
   - Stateless vs Stateful
9. Forwarding Tables
   - F.I.B
   - Routing Protocols
10. Distant Vector
11. Link States
    - Algorithms
    - Areas
12. BGP
    - External Gateway Protocol
    - AS Peering
    - Border Gateway Protocol
    - BGP Attributes
    - BGP for Large Networks
    - Network Architecture
    - VPN
13. Integration
    - Why IPv6 Integration ?
    - 6 generic scenarios
    - Tools overview
    - Scenarios
    - Backbone operator
    - Internet Access Provider
    - 3G/LTE
    - Enterprise
    - Home network and SOHO

Spanning Tree

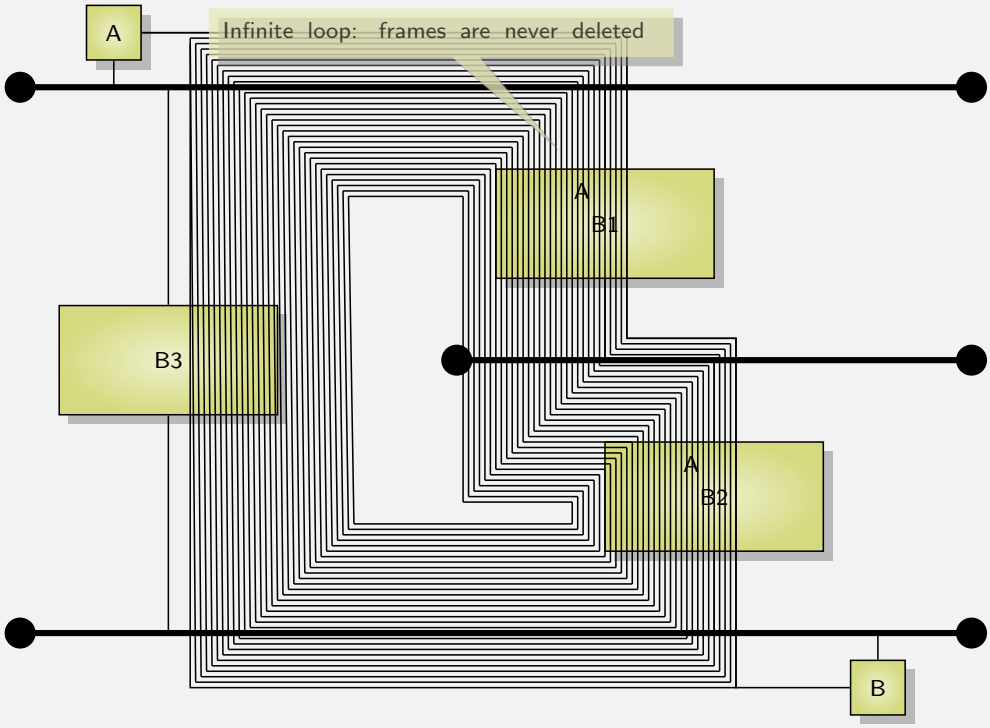
Bridge loops

Several bridges in a network



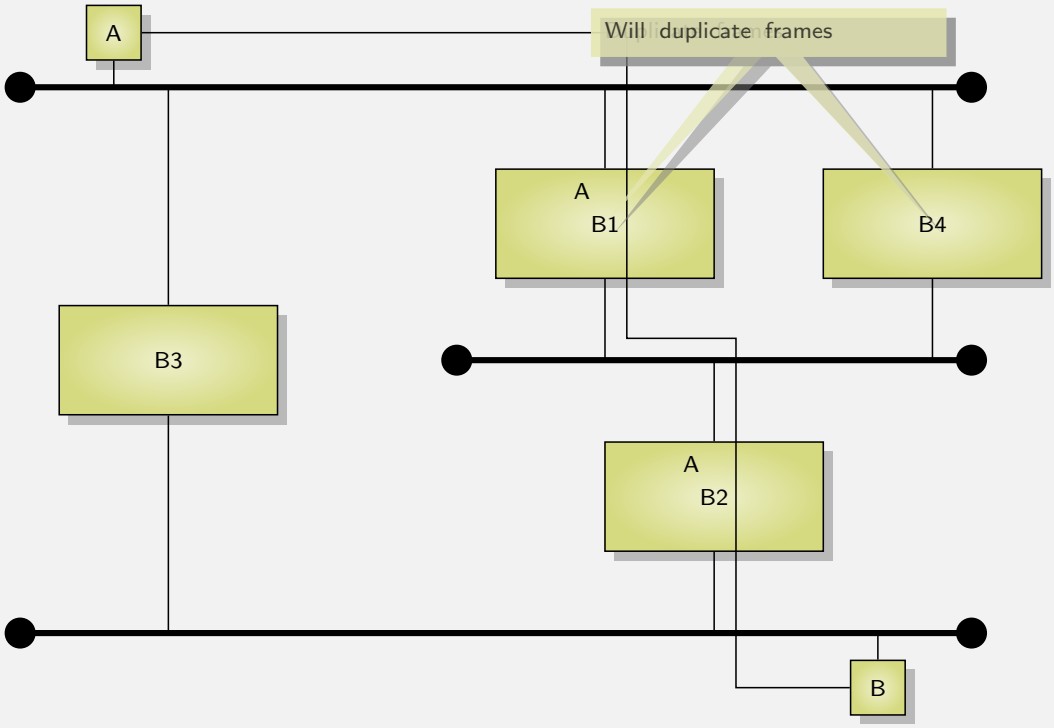
TELECOM Bretagne

Several bridges in a network



TELECOM Bretagne

Several bridges in a network



TELECOM Bretagne

Spanning Tree

- Bridging works well, if there is a single path to go from one point to another
- When there is several way to go to the same destination, there is a loop.
- Frame will turn forever in the network
  - Since frame are not modified by bridging it is not possible to stop them
  - If several paths, frame can also be duplicated
  - Network will collapse by overloading
- Several possible path = a graph
- Single path = a tree
- The goal is to disable some bridge interfaces to get back to a tree structure
- The tree must cover all the graph: Spanning Tree

Comments I

L'interconnexion redondante de réseaux en utilisant plusieurs ponts pose des problèmes. Les messages arrivent en copies multiples au destinataire. Les ponts ne modifiant pas les trames qu'ils recopient d'un réseau à un autre, une marque ne peut pas être ajoutée dans la trame pour indiquer que celle-ci est déjà passée par ce pont. La recopie ne s'arrête jamais et finalement la bande passante du réseau est complètement saturée. Or la possibilité de multiples interconnexions doit pouvoir être admise dans les réseaux :

- soit pour augmenter la fiabilité : si un pont tombe en panne ou si un sous-réseau est coupé, les stations peuvent toujours communiquer par un autre chemin,
- soit parce que le réseau est complexe et l'ajout par erreur d'un pont ne doit pas pénaliser le trafic.

Il faut revenir au cas qui fonctionnait convenablement. Cela revient à éviter la formation de boucles sur le réseau.

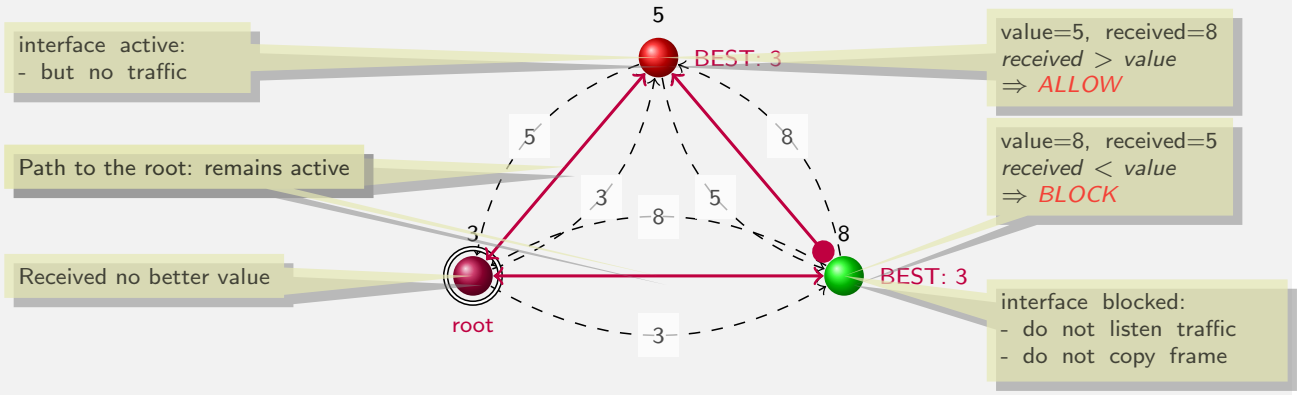
Un ensemble de sous-réseaux interconnectés par un pont peut être assimilé à un graphe. Si on supprime les boucles dans un graphe on obtient un arbre. Si cet arbre passe par tous les arcs, on obtient un arbre de recouvrement total, appelé en anglais *Spanning Tree*.

Spanning Tree

Algorithm

Main idea

- Compute a root (shortest value)
- Take the shortest path to the root
  - Guaranty that all costs will be different
- Disable interfaces leading to a larger cost



Comments I

Le transparent précédent donne une idée intuitive du fonctionnement de l'algorithme du Spanning Tree sur un exemple très simple. Le but est de désactiver des interfaces des ponts pour ne plus écouter ou émettre de trafic. Cet exemple montre clairement une boucle, mais la désactivation d'une interface parmi les 6 permettra de revenir a une structure en arbre.

Dans un premier temps, les nœuds vont échanger leur poids (qui par construction sera unique). Le nœud ayant le poids 3 ne reçoit aucune offre avec une valeur inférieure. Il se déclare racine de l'arbre. Le nœud 5 reçoit sa meilleure offre du nœud 3. Le chemin ayant reçu cette offre devient le chemin vers la racine et sera actif. Même chose pour le nœud 8.

Pour supprimer la boucle la seule possibilité consiste à désactiver une interface sur le chemin qui ne mène pas à la racine. Les nœuds 5 et 8 continue d'envoyer leur poids. Le nœud 5 reçoit une annonce qui est supérieure à la sienne, il bloque l'interface. Par contre le nœud 8 reçoit une annonce inférieure à la sienne, il laisse l'interface active.

On revient à une structure d'arbre, donc sans boucle. Les mécanismes de pontage des trames peuvent donc fonctionner sans problème.

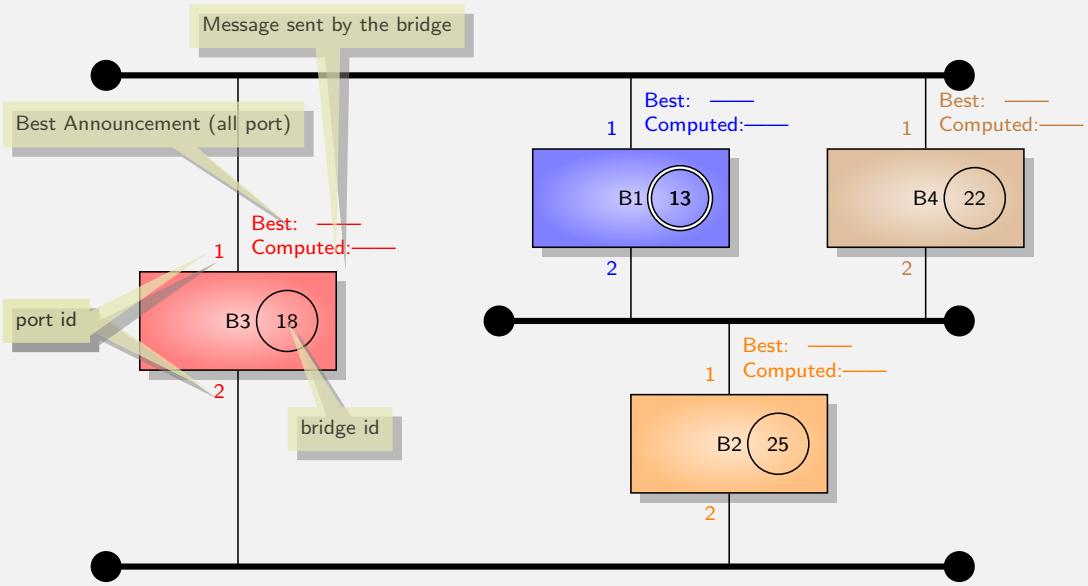
Cost vector

- Each bridge must have an unique id:
  - by construction id are not equal
  - MAC address of one of the interface can be used.
- Cost vector announced by bridge is composed of 4 values
  - Root id: the lowest value the bridge received
    - if no smaller value are received, root id = bridge id
  - Cost to the root: used to select the smallest path to the root
  - Bridge id
  - Port number or interface number
- $V = \langle V_1, V_2, V_3, V_4 \rangle < U = \langle U_1, U_2, U_3, U_4 \rangle$
  - if $V_1 < U_1$ or
  - if $V_1 = U_1$; $V_2 < U_2$ or
  - if $V_1 = U_1$ and $V_2 = U_2$; $V_3 < U_3$ or
  - if $V_1 = U_1$ and $V_2 = U_2$ and $V_3 = U_3$; $V_4 < U_4$

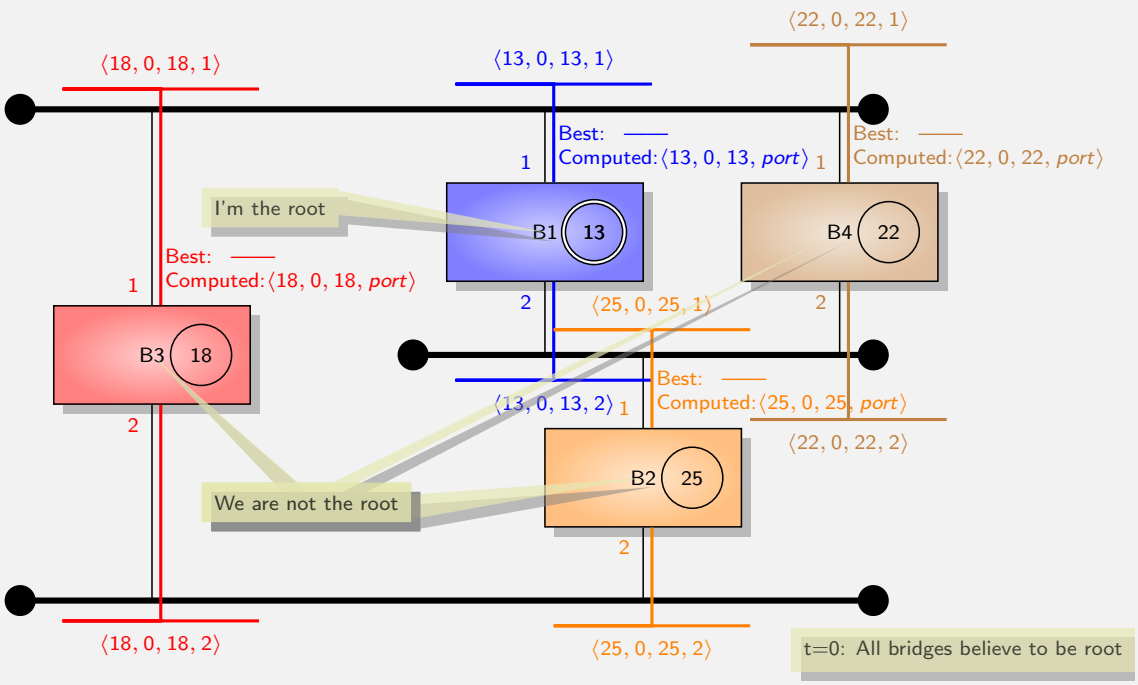
Algorithm overview

1. Each bridge/switch believes to be the root
2. When receiving a message smaller all the bridges except one will not be the root
   - 2.1 The port on which the smallest (best) message is received becomes the root port
   - 2.2 Other port must be active (Designated) or passive (no forwarding: Alternate) ports
     - if a smaller message than the one generated by be bridge is receive the port is alternate
     - if the bridge generate the best message for the link, the port is designated.
3. if the topology changes, the bridge informs the root:
   - 3.1 the root informs all the bridges of a topology change
   - 3.2 all bridges discard their forwarding table (i.e. MAC address location)
   - 3.3 all bridges start again learning process (and continue to forward).

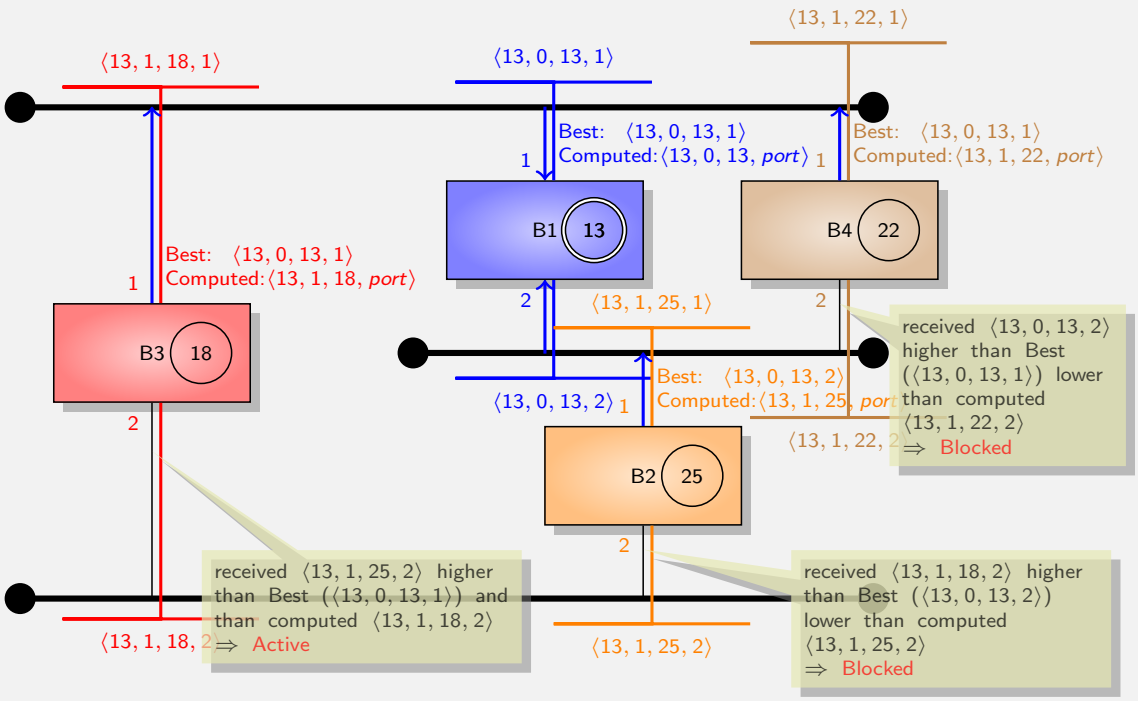
Example



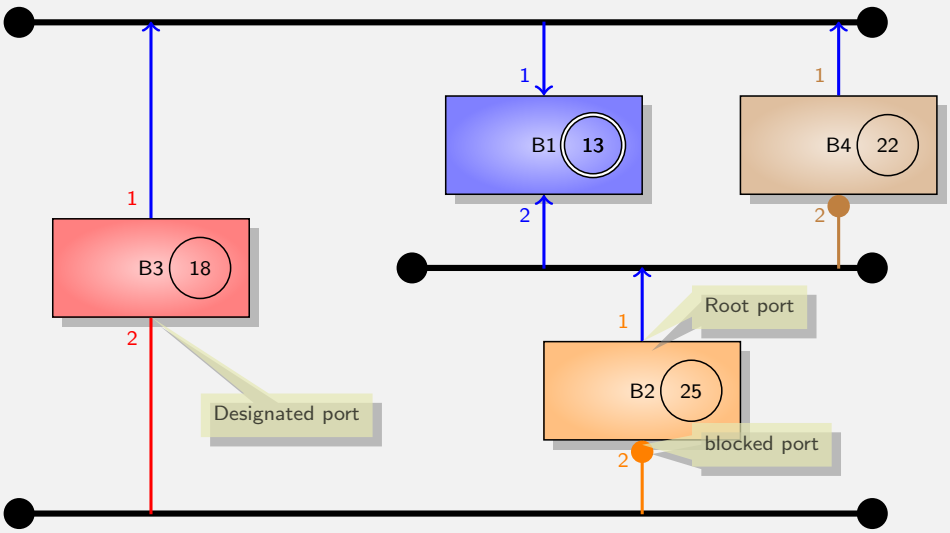
Example



Example



Example



Comments I

L'algorithme de Spanning Tree est relativement simple à mettre en œuvre. Il consiste à élire un pont particulier appelé racine et à choisir un chemin unique entre les ponts et la racine. Contrairement aux ponts transparents qui ne nécessitent pas d'adresse MAC, l'algorithme du Spanning Tree requiert que chaque pont ait une adresse sur le réseau pour échanger des messages conduisant à l'élection de la racine.

Chaque pont, en plus d'une adresse MAC, possède un identificateur. L'attribution de bonnes valeurs aux identificateurs par l'ingénieur réseau permet d'influer sur l?arbre couvrant pour qu'il utilise au mieux les ressources du réseau. La réalisation de l'arbre couvrant va se faire en désactivant (inhibant) certains des ports des ponts. Ainsi les boucles seront éliminées sur le réseau. L'envoi des messages appelés BPDU (pour le terme anglais : Bridge Protocol Data Unit) se fait en utilisant une adresse de multicast de niveau MAC. Cette adresse réservée à l?algorithme du Spanning Tree dispense l'administrateur du réseau d'indiquer les autres ponts voisins. Le pont les découvrira en écoutant le réseau.

Les ponts vont échanger des messages contenant :

- l'identificateur supposé de la racine par le pont émetteur du message,
- le co˚t de la liaison entre le pont et la racine supposée, c'est-à-dire le nombre de ponts qu'un message devra traverser pour atteindre la racine. Pour un pont se supposant être la racine, le co˚t est nul,
- l'identificateur du pont émetteur,
- le numéro du port sur lequel le message est émis.

Une configuration est meilleure qu'une autre si :

- l'identité de la racine est la plus petite,

Comments II

- en cas d'égalité sur l'identité de la racine, le co˚t du chemin vers la racine est plus petit,
- en cas d'égalité des deux premiers champs, l'identificateur de l'émetteur du message est le plus petit,
- finalement, si les trois premiers champs sont identiques, le message a été émis sur le port ayant le numéro le plus petit.

Intuitivement, cet ordre de priorité se comprend aisément. Le consensus va se faire pour désigner le pont racine comme le pont ayant le plus petit identificateur. Si plusieurs chemins sont possibles pour aller vers la racine, le chemin le plus court sera choisi. S'il existe plusieurs possibilités pour aller vers la racine avec le même coût, arbitrairement, le chemin proposé par le pont ayant le plus petit identificateur sera choisi. Si ce pont propose plusieurs fois le même chemin (cas où plusieurs ports sont connectés sur le même sous-réseau), le port le plus petit sera choisi.

A l'initialisation d'un pont, celui-ci se considère comme racine, il émet périodiquement sur chacun de ses ports un message $\langle id, 0, id, n \int du\ port \rangle$. Si, sur un des sous-réseaux, auquel il est connecté, circule un message contenant une meilleure configuration :

- la voie par laquelle cette meilleure configuration a été reçue devient le chemin pour la racine,
- une nouvelle configuration est calculée. Le premier champ reprend le champ du meilleur message, le co˚t est augmenté de 1. Les deux derniers champs ne sont pas modifiés. Cette nouvelle configuration sera émise périodiquement.

Comments III

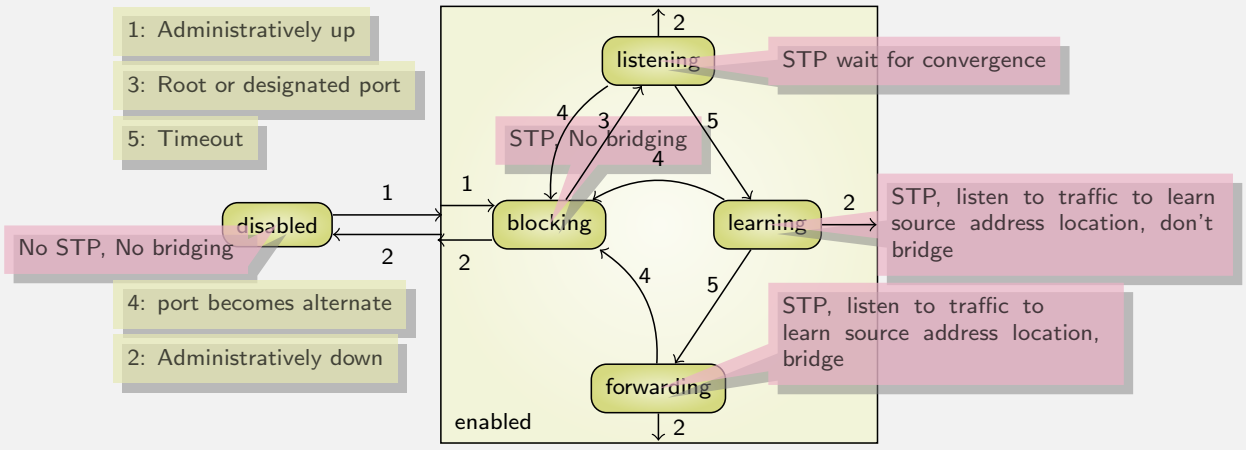
Le pont détermine ensuite quels ports ne menant pas à la racine doivent être activés ou inhibés en transmission. Il regarde le meilleur message de configuration reçu sur chacun de ses ports. Si le meilleur message de configuration pour un port donné est compris entre la meilleure configuration reçue (vrai par construction de l'algorithme) et la configuration calculée, alors le port est inhibé. Par contre, si aucun des messages reçus sur un port n'est inférieur à la configuration calculée, alors le port reste actif.

Spanning Tree

Rapid STP

Interface states

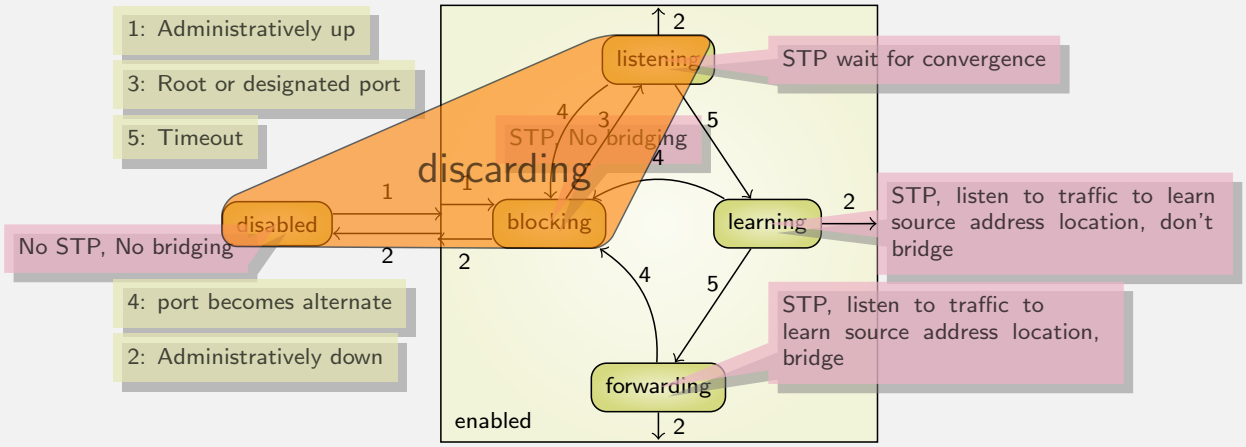
In STP, an interface may have different states:



Timer 5 is about 15 sec + up to 40 sec to detect failure ⇒ more than 1 minute to forward frames.

Interface states

In STP, an interface may have different states:



Timer 5 is about 15 sec + up to 40 sec to detect failure ⇒ more than 1 minute to forward frames.

Comments I

Un port est dans l'état désactivé (*Disabled*) s'il n'est pas connecté à un sous-réseau ou si l'administrateur du réseau l'a explicitement configuré ainsi. Un port dans cet état ne participe ni à la retransmission des trames, ni à l'algorithme du Spanning Tree.

Dans l'état activé, déclenché par configuration par l'administrateur du réseau (action 1 sur le schéma précédent), le port va participer à l'algorithme du Spanning Tree qui permettra de déterminer s'il activera sur ce port à le relayage des trames. Un port qui ne participe pas au processus de relayage, ne peut ni recopier en mémoire les trames qui circulent sur le sous-réseau auquel il est attaché, ni émettre les trames que d'autres ports ont reçues. On distingue quatre sous-états qui peuvent être quittés à n'importe quel instant pour revenir dans l'état désactivé, sous l'action de l'administrateur réseau ou la déconnexion du réseau (action 2):

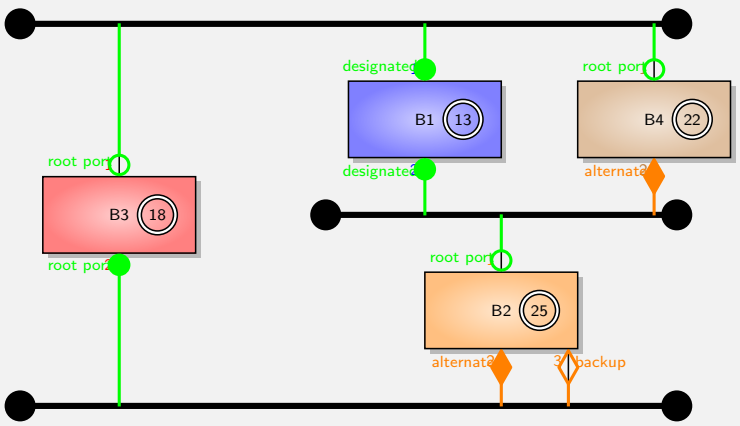
- dans l'état bloqué (*blocked*), le port ne participe pas à l'interconnexion, il participe au déroulement de l'algorithme du Spanning Tree ;
- dans l'état écoute (*listening*), le port a reçu des messages de configuration qui l'ont fait passer de l'état bloqué à cet état (action 3). Il sait que son port sera potentiellement un port vers la racine ou un port désigné. Le pont ne participe toujours pas à l'interconnexion, ceci pour éviter la formation de boucles qui pourraient survenir dans l'état transitoire. Le fait que le port participe de nouveau à l'interconnexion des réseaux est probablement d ° à la défaillance d'un autre pont sur le réseau ou à la coupure d'un lien. L'algorithme privilégie une rupture de la connectivité pendant une courte période. Le pont n'écoute pas non plus le trafic pour former une base de données d'adresses car pendant la période transitoire, la localisation des émetteurs peut changer. Dans cet état (mais également dans les autres) le port peut immédiatement retourner dans l'état bloqué (action 4) si l'algorithme du Spanning Tree reçoit des messages de configuration contraires indiquant que l'état du port est alternatif ;

Comments II

- dans l'état apprentissage (*learning*), le port ne participe toujours pas à l'interconnexion, mais écoute le trafic pour repérer les stations qui lui sont rattachées. Cette période d'apprentissage permet au port, lorsqu'il deviendra actif, de ne pas inonder le réseau de recopies inutiles dues à une absence de l'adresse de l'émetteur dans la base de données de filtrage.
- dans l'état relayage (*forwarding*), le pont participe à l'interconnexion des réseaux tout en continuant a dérouler l'algorithme du Spanning Tree.

Port Role and State

	discarding	learning	forwarding
Designated			
root port			
alternate			
backup			



Network recovery

- Each bridge send periodically a ST message
  - interval: between 1s and 10s (recommended: 2s)
- Each message contains an age field giving the time the root has been seen.
- If age reach a limit (between 6s and 40s (recommended 20s)), this implies a connectivity problem between the bridge and the root.
  - Spanning Tree construction is restarted (start in the blocking state)
  - new root may be selected.
- Network may take about 1mn to recover:
  - Up to 20s to detect a path to root failure
  - 15s to move from learning
  - 15s to move to forwarding state.

Comments I

En cas de panne d'un équipement ou d'une liaison, le temps de passage de l'état bloqué à l'état relayage est relativement long puisque les temporisateurs (action 5) sont configurés avec des durées de 15 secondes et avec le paramétrage par défaut du Spanning Tree, il faut jusqu'à 20 secondes pour qu'un pont détecte une rupture de connectivité vers la racine.

A l'origine du protocole, cette longue rupture de connectivité était considérée comme salutaire comparée aux dommages des boucles. Mais avec le développement de nouvelles applications multimédias ou de voix sur IP, cette perte de connectivité peut avoir des conséquences graves pour les applications. Pour réduire ces durées, l'algorithme du Spanning Tree a été adapté pour réagir plus vite. Il a été défini par le groupe IEEE 802.1w sous le nom de Rapid Spanning Tree Protocol (RSTP).

Slow Convergence

- When SPT has been developed, preventing loops was more important than time to recover when a bridge or a link fails.
  - File transfer, mail,. . . are not real time
  - It may take more than a minute to recover
- Nowadays with real time protocol such as VoIP or streaming these delays are not acceptable
- Rapid Spanning Tree Protocol developed by IEEE 802.1w is backward compatible with STP.
  - if 3 consecutive ST messages are lost, ST topology is restarted
    - up to 6s to restart Spanning Tree construction
- Adapt to switched technologies
  - Link to other switches
  - Link to hosts (No Bridge PDU).

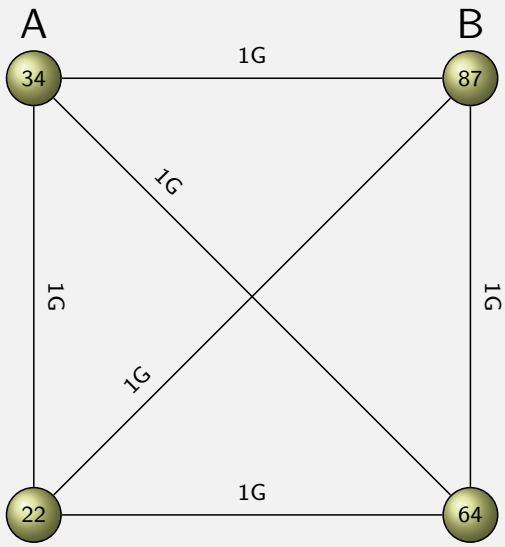
Comments I

La principale modification concerne la détection de la perte de connectivité avec la racine. RSTP considère à la place la perte de connectivité avec le pont qui conduit vers la racine. Chacun des ponts émet une message de Spaning Tree (Bridge PDU ou BPDU) toutes les 2 secondes. Si un pont perd 3 BPDU consécutifs, il lance une phase de reconfiguration avec recherche d'une nouvelle racine. L'algorithme convergeant rapidement que quelques secondes pour reconstruire l'arbre couvrant. De plus, il ne repasse pas par les phases d'écoutés et d'apprentissage. La perte de connectivité est donc réduite à quelques dizaines de secondes.

RSTP utilise aussi les propriétés des réseaux construit sur des architectures commutées oˇ il existe que deux types

de liaisons: celles entre deux commutateurs, celles entre un commutateur et un équipement terminal.

Questions



Which router becomes root?

Which ports are root ports?

Which ports are alternate ports?

Will a frame from A to B follows the optimal path?

Are all the link used ?

What happen if root bridge links were 10 Mbit/s links? How to solve this?

Alternatives to Spanning Tree

- Spanning Tree can be viewed as a fuse to protect network but
  - Traffic Engineering is quite complex
  - The shortest path is not always used
  - Some links are not used
- Alternatives based on routing protocol are currently standardized:
  - IETF TRILL (Transparent Interconnection of Lots of Links)
    W http://datatracker.ietf.org/wg/trill/
  - Shortest Path Bridging (SPB) (IEEE 802.1aq)
    W http://www.ieee802.org/1/pages/802.1aq.html

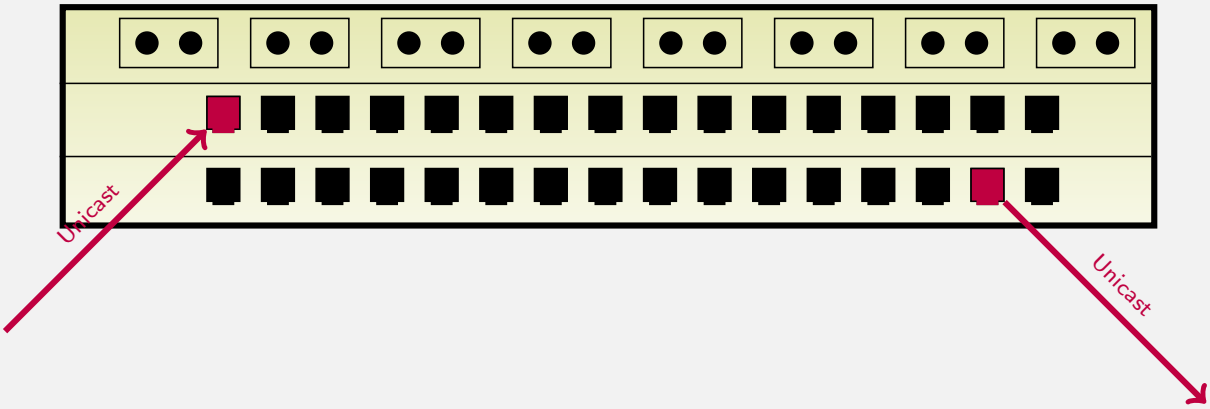
VLAN

Introduction

Virtual Local Area Network

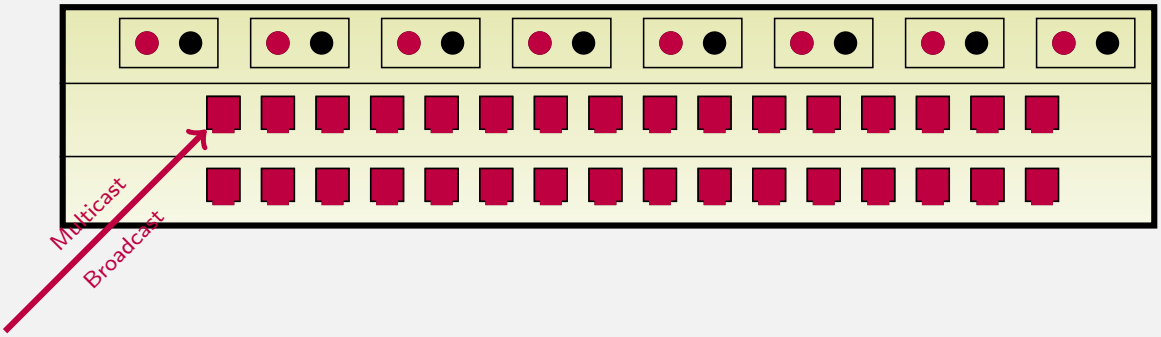
- LAN allows a simple management:
  - No address allocation procedures
  - Simple interconnection with bridges
- But suffers from some scalability and security problems
  - Address list must be construct for each port
  - Traffic cannot be filtered
  - Broadcast or Multicast frames flood the network
  - Cabling constraints: Moving one equipment from one network to another implies modifying the topology.
- VLAN solves this problem separating the physical infrastructure into logical (virtual) ones.
  - Physical: Links, switches
  - Logical: Configure switches to assign a virtual network to a group of ports.

VLAN on a switch

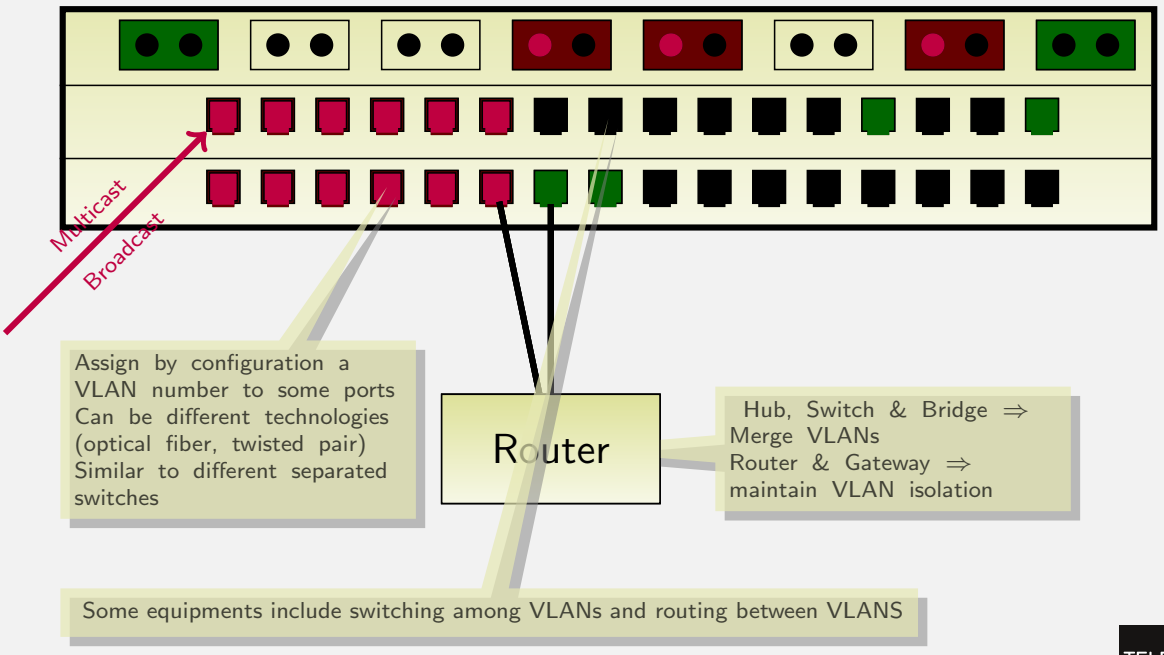


VLAN on a switch

VLAN ▶ Introduction



VLAN on a switch



Comments I

Les réseaux virtuels ou VLAN (*Virtual LAN*) constituent une alternative à la construction de réseaux de niveau 3 à l'aide de routeurs. Construire un réseau virtuel revient à avoir sur une même infrastructure physique (c,blage, équipements d'interconnexion), plusieurs réseaux de niveau 2 complètement indépendants. Les réseaux virtuels présentent plusieurs avantages :

- facilité de mise en œuvre : contrairement aux réseaux de niveau 3 où une gestion relativement stricte du plan d'adressage est nécessaire, les réseaux virtuels gardent la souplesse des réseaux de niveau 2. Des logiciels d'administration facilitent leur configuration ;
- confidentialité : sur un réseau de niveau 2, il est relativement difficile de filtrer les trafics, un équipement peut dialoguer avec n'importe quel autre, les ponts ne filtrant que les trames d'équipements situés sur un même réseau.
- Au niveau 3, l'attribution d'adresses liées à la topologie du réseau permet la mise en place de règles de filtrage dans les routeurs permettant la création de firewall. Comme le trafic entre les réseaux virtuels est isolé, il est possible de limiter les accès à certains équipements.
  Il est à noter que les stations devront posséder une adresse IP par réseau virtuel ;
- souplesse d'utilisation : il est facile de donner ou de retirer les accès aux différents réseaux virtuels de l'entreprise. Cela évite de modifier le c,blage dans les armoires de brassage.

Comments II

Un réseau virtuel peut être vu comme une réduction de la portée des trames en diffusion. Les transparents précédents montrent comment sont gérées les trames dans un commutateur. En règle générale, le trafic point-à-point est aiguillé uniquement vers le destinataire, cela permet des échanges simultanés entre plusieurs équipements. Par contre, quand une trame avec une adresse de diffusion (par exemple un broadcast ARP) est émise par un équipement, celle-ci est envoyée sur tous les ports du commutateur.

Dans l'exemple, les ports marqués en routage (port 3, 4 et 5 optiques et 1 à 6 des ports paires torsadées) peuvent être alloués au réseau virtuel 1et les ports marqués en vert au réseau virtuel 2. Le comportement est identique à l'utilisation de deux commutateurs différents. Une trame en diffusion (ou multicast) émise par un équipement du réseau virtuel 2 ne sera reçue que par les autres équipements de ce réseau virtuel. Pendant ce temps, les équipements des réseaux virtuels 1 peuvent continuer à dialoguer.

En conséquence, une trame émise par un équipement situé sur le port 1 ne pourra pas directement joindre un équipement situé sur le port 5. Un routeur devra être utilisé pour interconnecter les réseaux virtuels entre eux. La définition de réseaux virtuels nécessite plusieurs modifications au modèle de pontage établi par l'IEEE pour les réseaux de niveau 2, décrite dans les paragraphes suivants.

L'appartenance à un VLAN est généralement lié à la configuration des ports physiques des commutateurs. Il s'agit de la méthode la plus simple : elle associe aux ports du commutateur, un numéro de VLAN. Cette méthode est la plus s ˚re car l'appartenance à un réseau virtuel ne dépend pas de facteurs extérieurs au commutateur. Un utilisateur ne peut pas, en modifiant la configuration de sa machine, changer de réseau virtuel.
Les équipements terminaux ignorent la notion de VLAN. Ce sont les équipements d?interconnexion qui déterminent l'appartenance. Plusieurs méthodes de configuration des équipements d'interconnexion pour décrire cette appartenance à un réseau virtuel sont possibles.

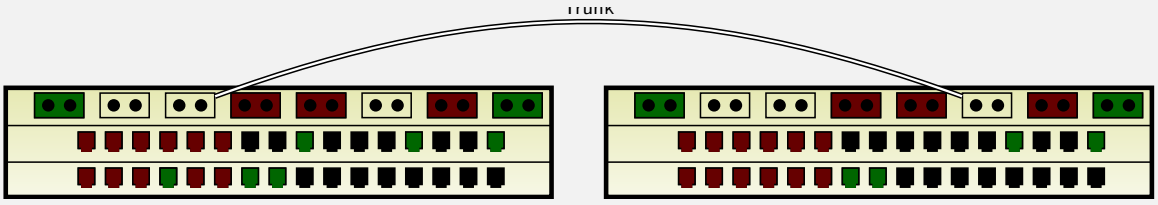
Comments III

- Manuelle: Cette méthode ne convient que lorsque le réseau virtuel n'est constitué que d'un seul commutateur. L'administrateur du réseau se connecte sur l'équipement et entre les fichiers de configuration. Le risque d'erreur est relativement important.
- Semi-automatique: Cette méthode se base sur les plates-formes d'administration SNMP. Généralement, ces outils ne fonctionnement que si tous les équipements d'interconnexion sont du même constructeur. Il peut y avoir la découverte automatique des commutateurs présents sur le réseau. L'administrateur peut ensuite attribuer un numéro de VLAN à chaque port. La configuration est ensuite stockée dans chaque commutateur.

VLAN

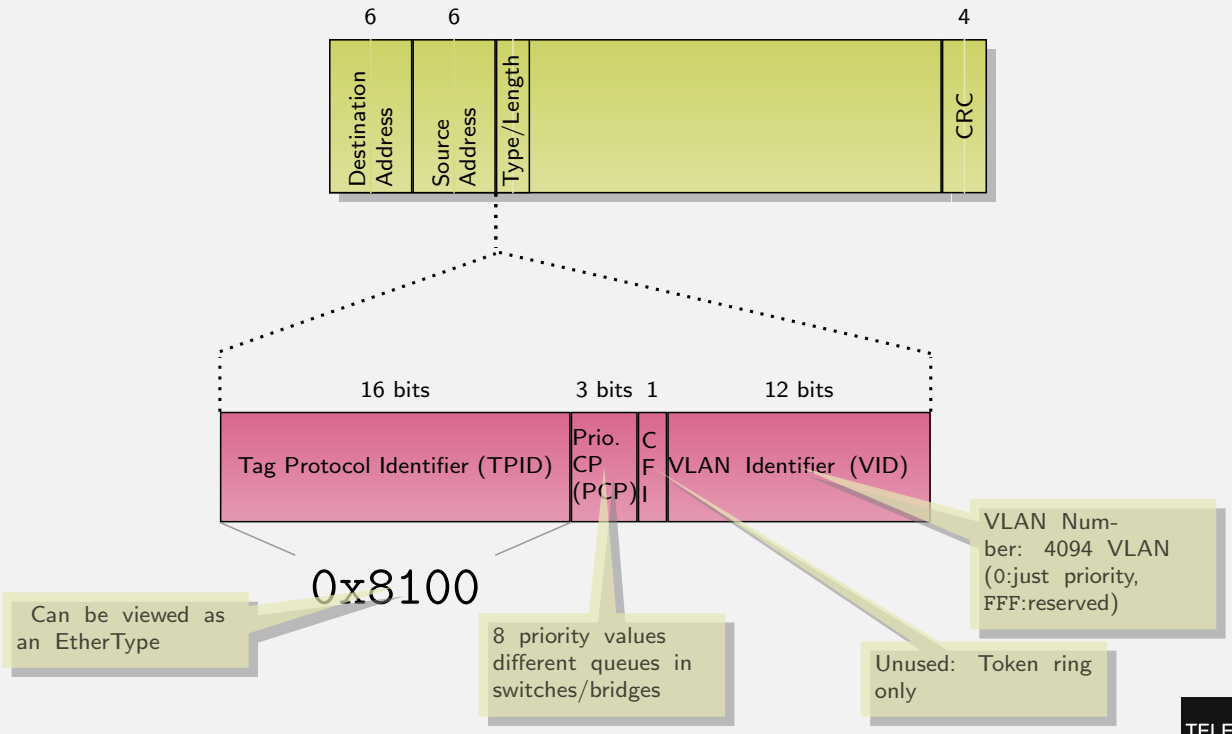
IEEE 802.1p/Q

Tags IEEE 802.1p/Q



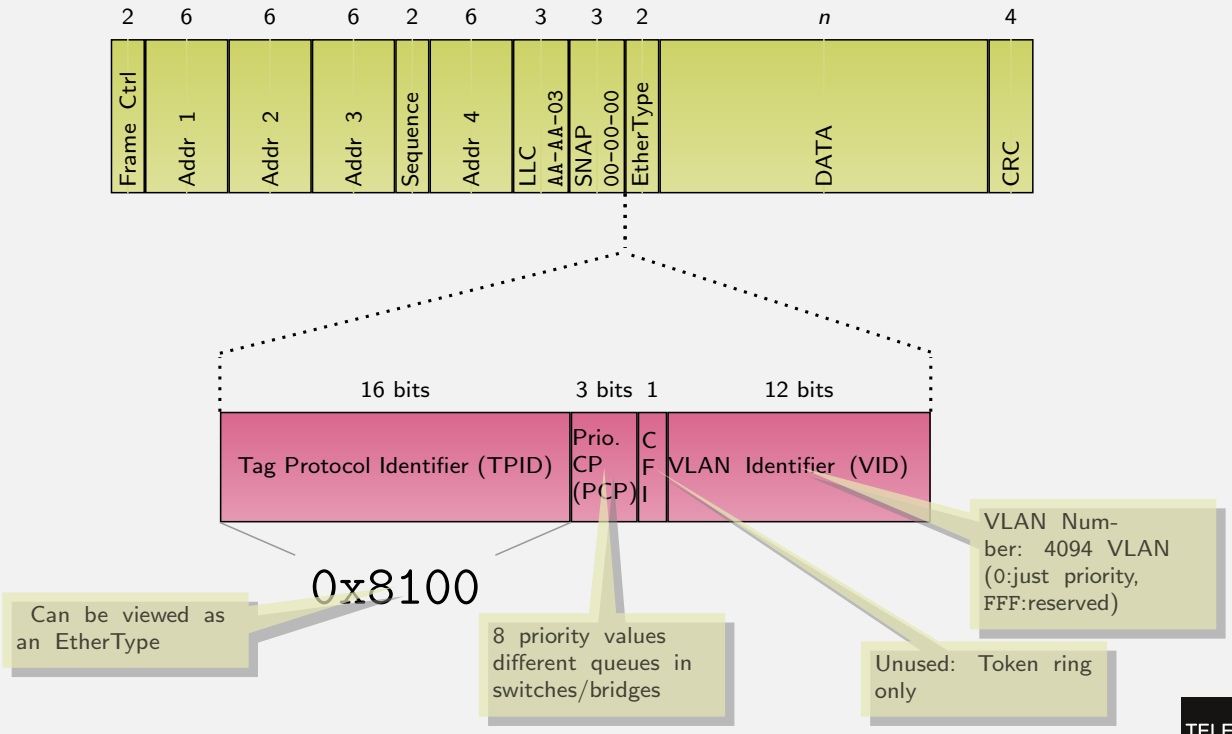
- When several switches are involved, VLAN value must be carried between equipments.
- Switches add a tag in the frame containing the VLAN number.
- IEEE 802.1p/Q defines tag format for VLAN and priority.

Ethernet vs IEEE 802.3

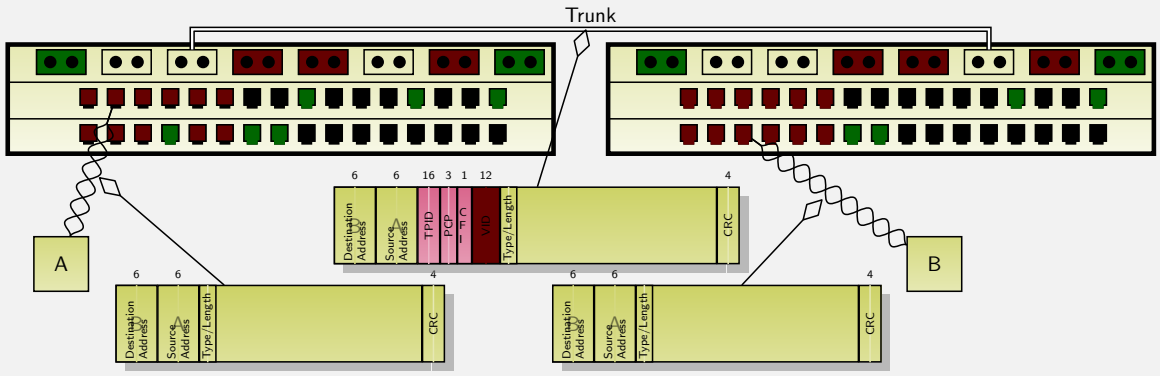


Ethernet vs IEEE 802.3

VLAN ▶ IEEE 802.1p/Q



Tags IEEE 802.1p/Q



- VLAN tag is added between switches:
  - Host generally do not tag frames
  - Maximum frame size can be exceeded when tag is added
    - not a problem with jumboframe (9 KB)
    - IEEE 802.3ac also allow Ethernet frame with up to 1522 Bytes
- There is a default VLAN without tagging, switch can be configured to associate internally a VLAN value.

Comments I

Quand le réseau comprend plusieurs commutateurs, il est nécessaire de pouvoir transporter l'information d'appartenance d'un équipement d'interconnexion à un autre. Ceci se fait en ajoutant une étiquette aux trames transportées indiquant le numéro du VLAN. Les équipements d'interconnexion ne doivent connaître que les appartenances locales aux réseaux virtuels. Cet étiquetage est souvent fait par les équipements d'interconnexion car les équipements terminaux émettent généralement des trames non étiquetées.

La norme IEEE 802.1p permet de définir un format commun pour les étiquettes indépendamment des constructeurs. Pour ce faire, l'en-tête des trames MAC doit être modifié pour pouvoir y insérer des informations supplémentaires, tout en garantissant une compatibilité avec les anciens équipements. Pour ceux-ci, les informations supplémentaires seront vues comme un protocole de niveau supérieur. Ainsi pour Ethernet, le champ protocole 0x8100 définit l'encapsulation IEEE 802.1p. Cette encapsulation se résume à ajouter une étiquette de deux octets appelée TCI (Tag Control Information). Elle peut être aussi interprétée comme l'ajout de deux champs : TPID (Tag Protocol Identifier) et TCI entre l'adresse de la source et le champ type/protocole de la trame MAC. Pour le Wi-Fi, cette encapsulation se fait en utilisant SNAP pour placer l'identificateur de protocole 0x8100.

Le champ TCI se décompose en trois parties :

- un champ priorité sur 3 bits ;
- un drapeau appelé CFI (Canonical Format Indicator) sur un bit. qui permet en théorie d'insérer des informations de routage par la source (liste des ponts à traverser), mais en pratique ce bit n'est pas utilisé.
- un champ VID (VLAN Identifier) permet de marquer l'appartenance de la trame à un VLAN particulier.

L'insertion de cette information peut être réalisée par la station émettrice de la trame ou par les équipements d?interconnexion (Hub, commutateurs, ponts, etc). Dans ce dernier cas :

- le checksum de la trame doit être recalculé ;

Comments II

- dans le cas des réseau IEEE 802.3, les octets de bourrage devenus inutiles peuvent être retirés.

Surtout dans le cas d'Ethernet et du protocole IEEE 802.3, la longueur des trames peut excéder la taille maximale autorisée (c'est-à-dire 1 500 octets pour les données et 18 octets pour l'en-tête). La norme IEEE 802.3ac propose d'étendre la taille maximale des trames à 1522 octets si le champ type/longueur contient le TPID. Mais cette longueur peut être incompatible avec d'anciens pilotes de carte mais comme ce format ne concerne que des équipements récents, le risque d'incompatibilité est très faible.

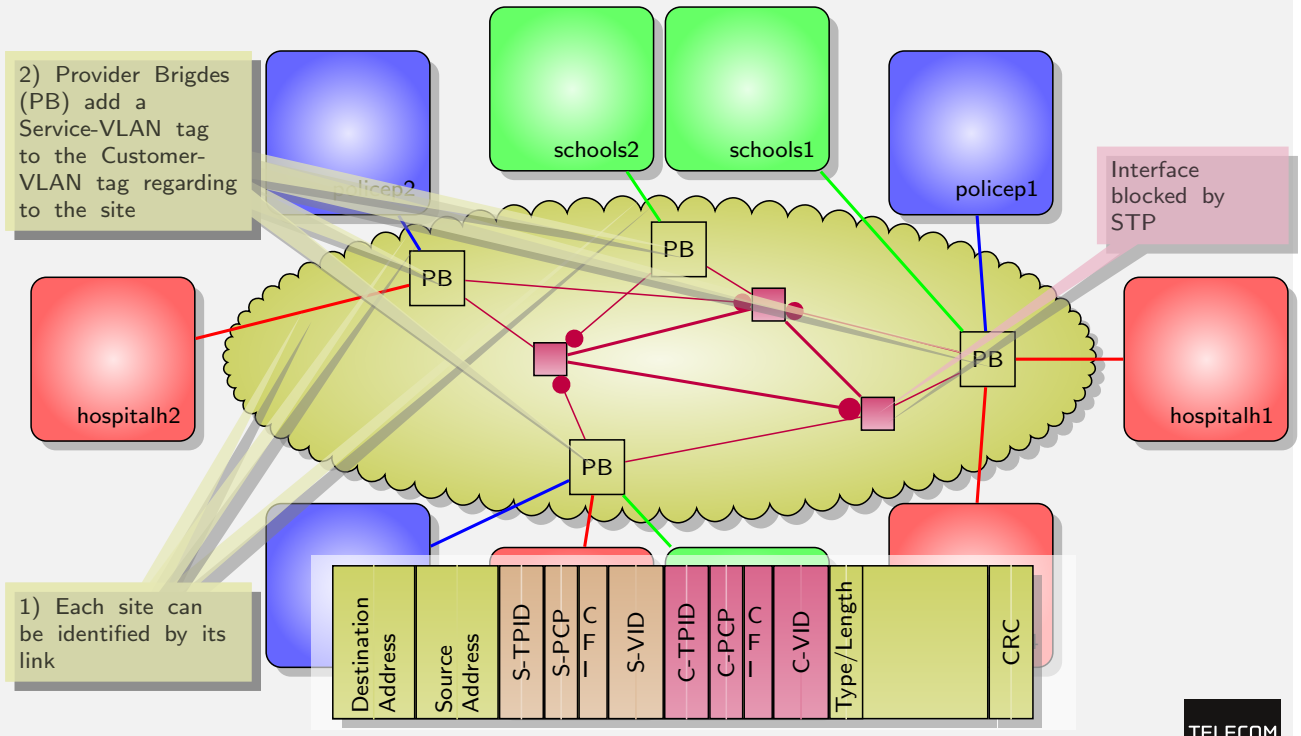
Metropolitan Networks

QinQ

Metropolitan Ethernet

- Switched networks with VLAN are particularly powerful:
  - frame switching is efficient
  - distance is no more a limitation (no CSMA/CD)
  - OPEX is reduced (equipments automatically configure themselves)
- But scalability and security are limited:
  - Switches must learn all the MAC address
  - An attacker may inject fake addresses
  - VLAN numbers must be globally unique
- QinQ allows to stack VLAN tag
- MACinMAC(IEEE 802.1ah) allows to encapsulate Ethernet frame into Ethernet frame.

IEEE 802.1ad (QinQ)



IEEE 802.1ad (QinQ)

- IEEE 802.1ad defines two categories of VLAN tags:
  - Customer VLAN (or C-VLAN) generated by the customer switches
  - Service VLAN (or S-VLAN) added by the provider.
    - S-VID are chosen by the provider to link a group of sites.
    - S-VID designates a customer.
- S-VLAN guaranty unicity between two customers
  - if two independent company select the same VLAN number, traffic will not be merged
- S-VLAN guaranty security
  - A company cannot access to another company traffic.
- S-LAN are defined by the value 88a8 in the S-TPID
- More simple to manage than Layer 2 VPN based on IP routing and MPLS.

Comments I

Avec le développement des réseaux métropolitains, il est nécessaire de pouvoir protéger des trafic venant d'une société. Les VLAN sont évidemment la solution la plus adaptée mais impliquent certaines contraintes en terme de numérotation. Il est possible pour l'opérateur de définir sur les liaisons trunk vers ses clients les numéros de VLAN autorisés ou non. Mais cela impose au client de numéroter ses VLAN dans une plage restreinte. Pour offrir plus de souplesse, le document IEEE 802.1ad propose d'empiler deux tags, seul le premier étant actif.

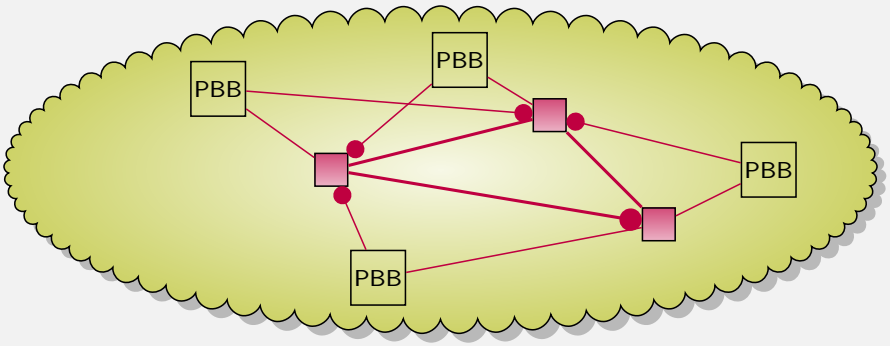
Le premier tag est appelé S-VLAN (pour Service VLAN) est purement interne au réseau de l'opérateur métropolitain. Il sert à désigner le client de l'opérateur. Le second tag est appelé C-VLAN (pour Customer VLAN) et est celui choisi par le client pour ses besoins. Il n'y a dons pas de risque de conflit entre deux clients puisque même si ils choisissent la même valeur de C-VLAN, la valeur du S-VLAN permettra de les différencier). Comme le S-VLAN est ajouté ou retiré par le commutateur en entrée ou en sortie de l'opérateur (PB :*Provider Bridge*), les clients n'ont pas la possibilité de le modifier. Cela renforce la sécurité du réseau car une société ne peut pas avoir accès au trafic d'une autre.

Cette solution de pontage sur de grandes distances est une alternative intéressante à des solutions de VPN (*Virtual Private Network*) basées sur du routage IP et sur l'utilisation de MPLS.

Metropolitan Networks

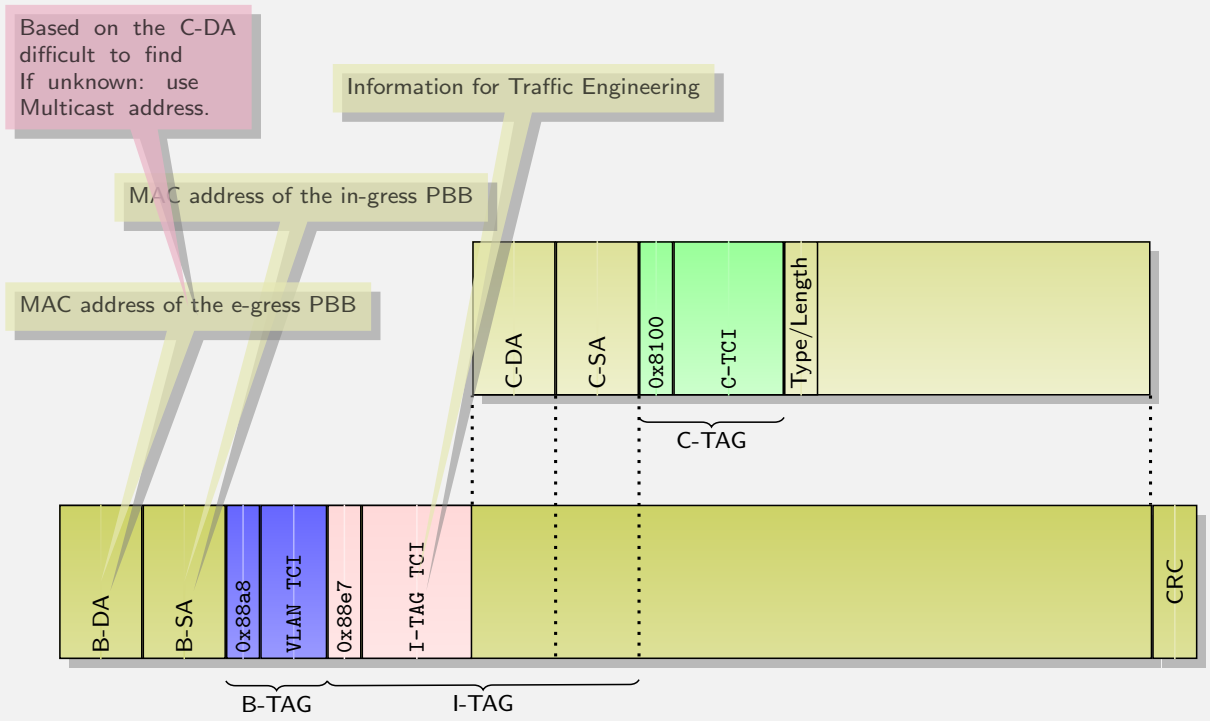
Mac In Mac

IEEE 802.1ah (MACinMAC)

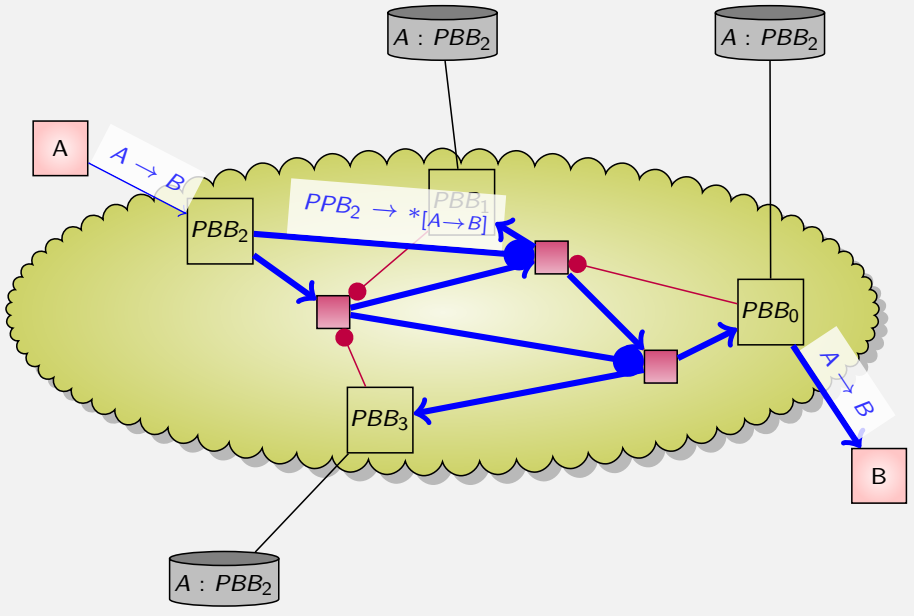


- Switches in the provider network have to memorize all the MAC addresses
  - require a lot of memory.
- an attacker forging the source mac address may:
  - slow down the network saturating core tables,
  - make some equipments unreachable using their MAC address.
- Provider Backbone Bridge adds an Ethernet header.

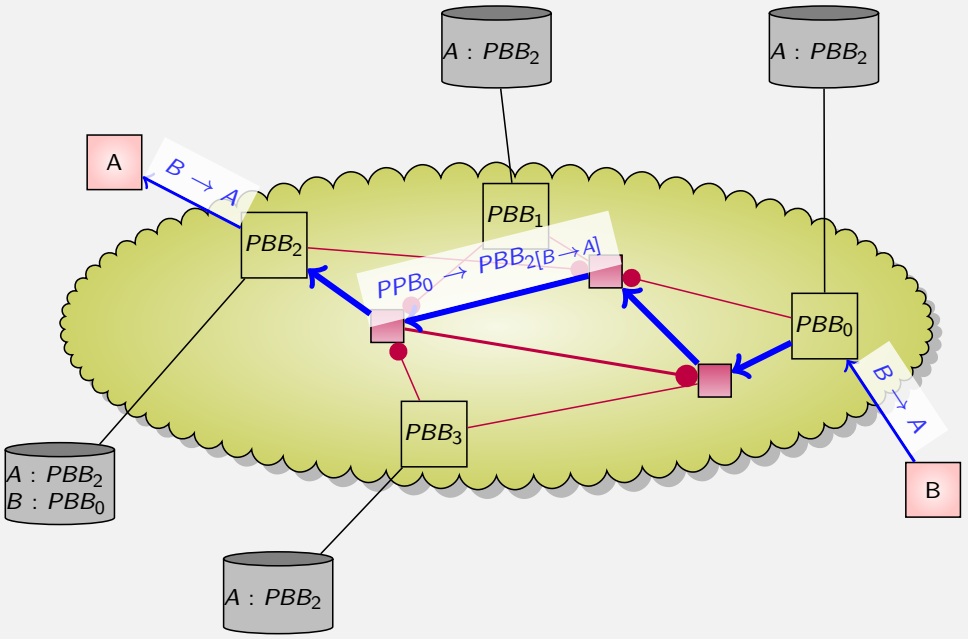
IEEE 802.1ah (MACinMAC)



IEEE 802.1ah (MACinMAC)



IEEE 802.1ah (MACinMAC)



Comments I

La norme IEEE 802.1ah permet d'encapsuler une trame Ethernet dans une autre. Cela permet de réduire le nombre d'entrées dans les tables de relayage des équipements du cœur de réseau. Comme pour la norme QinQ, deux niveaux de VLAN sont possible. le premier (appelé B-TAG) contient le VLAN ajouté par l'opérateur pour distinguer les utilisateurs. La norme prévoit d'insérer également d'autres informations dans un I-TAG qui serviront à la gestion de la qualité de service dans le réseaux de l'opérateur.
Comme il existe deux niveaux d'adresses, il faut pouvoir établir la correspondance entre l'adresse de destination contenue dans la trame du client et l'adresse MAC de destination du Provider Backbone Bridge (PBB) qui devra retirer l'encapsulation IEEE 802.1ah. Le moyen le plus simple consiste à construire une table dans les ponts en bordure du réseau de correspondance entre les adresses MAC du client et les adresses mac des PBB de sortie. Cet apprentissage se fait automatiquement en lisant les adresses sources au niveau du backbone et du client. Quand un pont n'a pas dans ses tables la correspondance, il utilise à la place une adresse de multicast. L'information est donc transmise à l'ensemble des ponts de bordure de l'opérateur. Cette trame en multicast a permis également à tous les ponts en bordure d'apprendre la localisation de la source. Quand le destinataire répond, le pont en bordure est capable de trouver l'adresse du pont de sortie du réseau de l'opérateur. Les mécanismes de commutation font que cette trame est directement et uniquement envoyé vers cette destination.

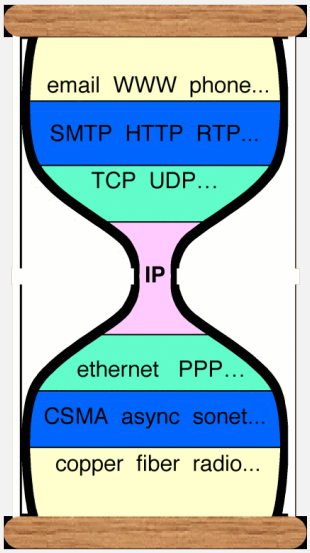
References

- http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4463774
- www.itg523.de/oeffentlich/08april/Knoll_Carrier_Ethernet.pdf

Concepts

Datagram

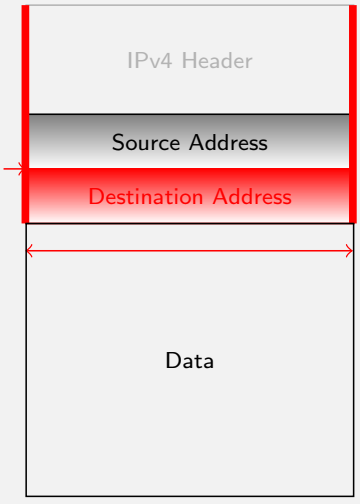
IP Layer



- IP is kept simple
  - Forwards packet towards destination
- IP on everything
  - Adapt IP protocol on every layer 2
- Everything on IP
  - Write applications to use IP layer (through L4: TCP, UDP)
- IP must facilitate network interconnection
  - Avoid ambiguities on addresses

W http://www.ietf.org/proceedings/01aug/slides/plenary-1/index.html Steve deering, Watching the Waist of the Protocol Hourglass, IETF 51, London

Destination Address Processing



The destination address must be easily accessible:

- Fixed location
- Fixed size
- Aligment in memory

RFC 791 (Sept 1981)

`Addresses are fixed length of four octets (32 bits)`

Facts on Addresses

Historical view

Historical facts

- 1983 : Research network for about 100 computers
- 1992 : Commercial activity
  - Exponential growth
- 1993 : Exhaustion of the class B address space
  - Allocation in the class C space
  - Require more information in routers memory
- Forecast of network collapse for 1998!
  - 1999 : Bob Metcalfe ate his Infoworld 1995 paper where he made this prediction

TELECOM Bretagne

Facts on Addresses

Emergency Measures

Emergency Measures: Better Addresses Management

RFC 1517 - RFC 1520 (Sept 1993)

- Ask the internet community to give back allocated prefixes (RFC 1917)
- Re-use class C address space
- CIDR (Classless Internet Domain Routing)
  - network address = prefix/prefix length
  - less address waste
  - recommend aggregation (reduce routing table length)
- Introduce private prefixes (RFC 1918)

TELECOM Bretagne

Facts on Addresses

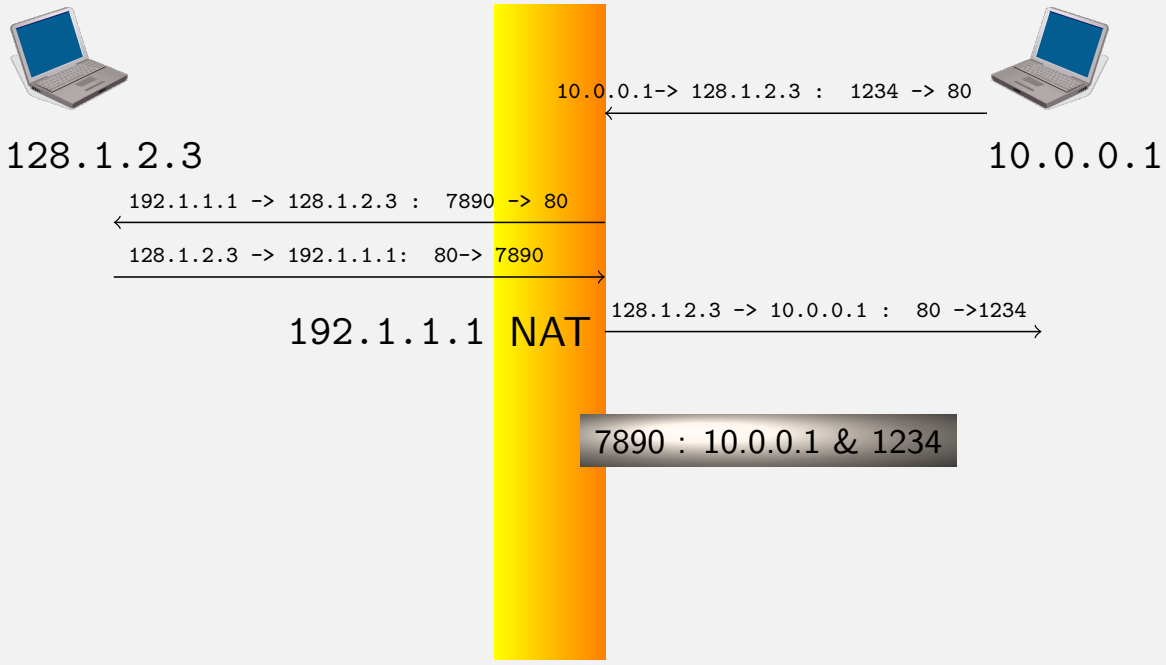
NAT

Emergency Measures: Private Addresses (RFC 1918 BCP)

- Allow private addressing plans
- Addresses are used internally
- Similar to security architecture with firewalls
- Use of proxies or NAT to go outside
  - RFC 1631 , RFC 2663 and RFC 2993
- NAPT is the most commonly used of NAT variations

TELECOM Bretagne

How NAT with Port Translation Works



NAT Impact

first consequence

The application does not know its public name.

second consequence

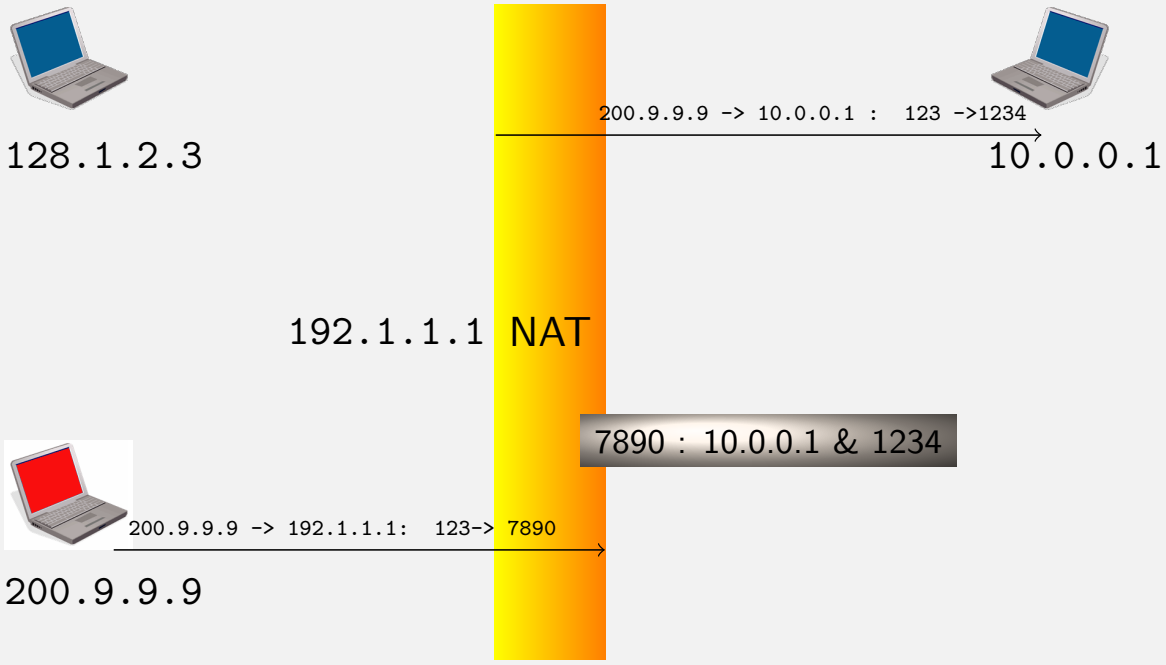
It is difficult to contact a NATed equipment from outside

- Security feeling
- Solutions for NAT traversal exist

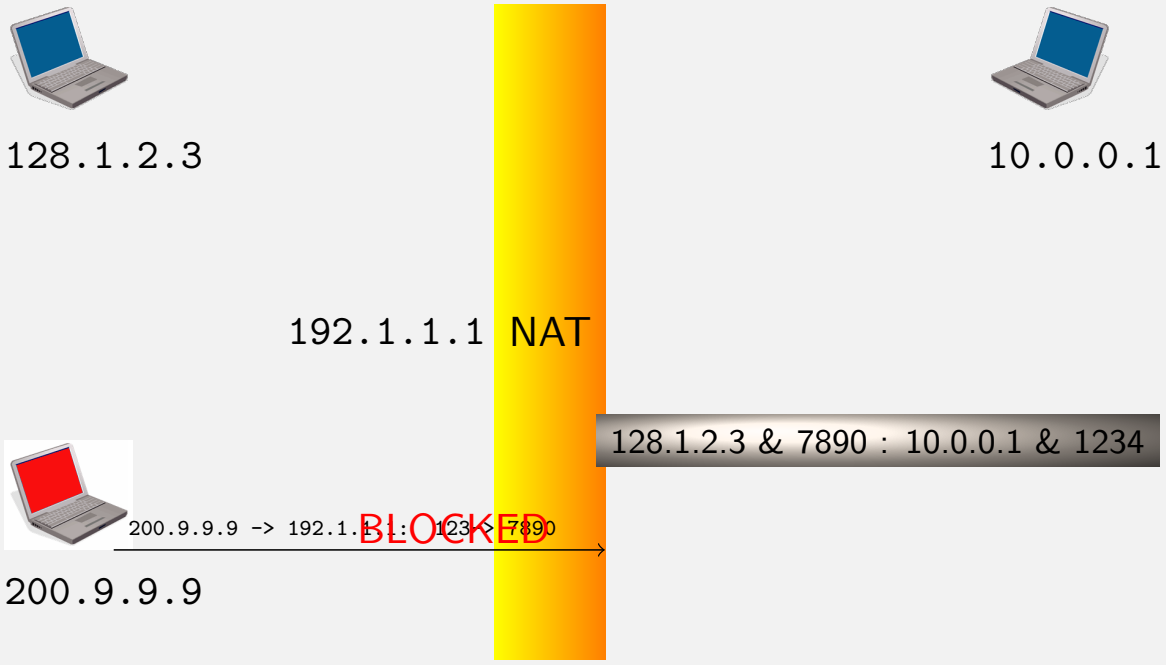
third consequence

There is no standardized behavior for NAT yet

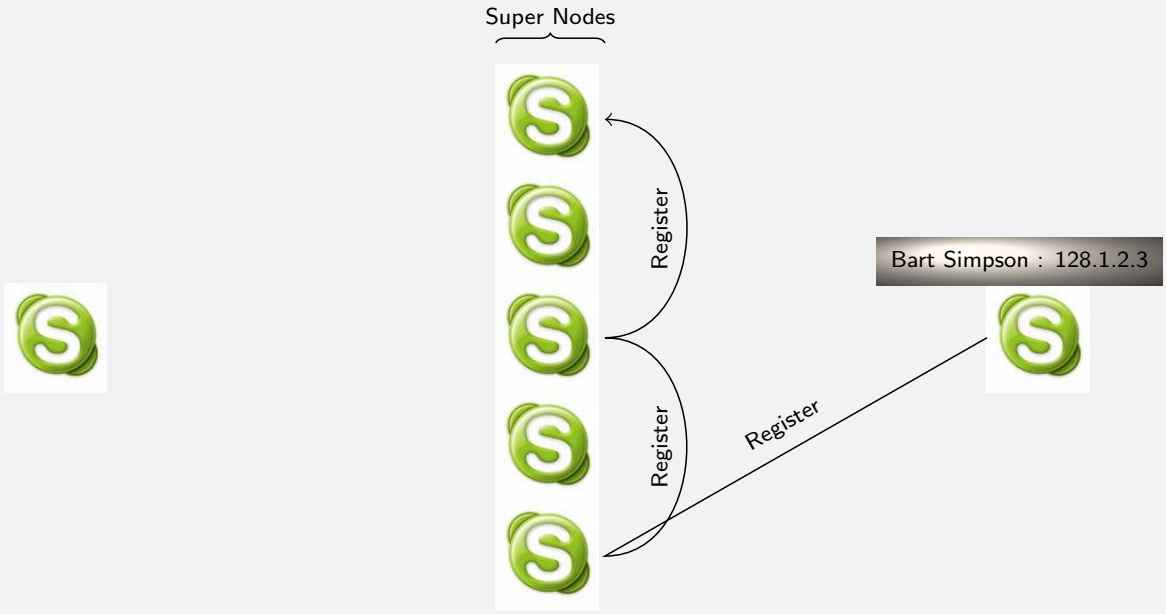
Conic NAT



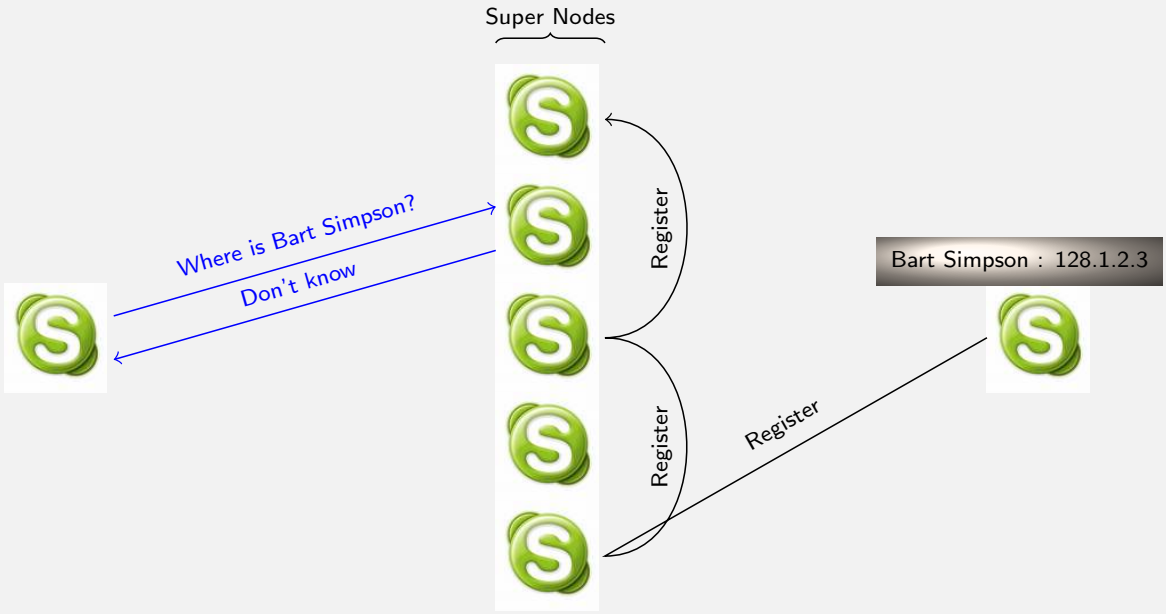
Restricted NAT



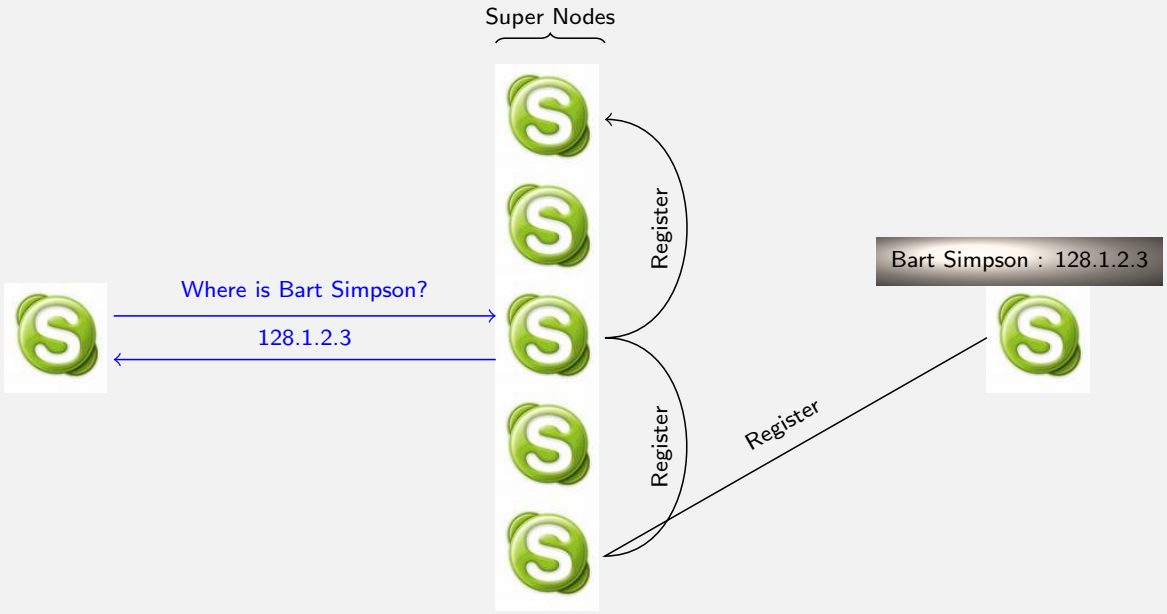
NAT Traversal and Peer To Peer



NAT Traversal and Peer To Peer

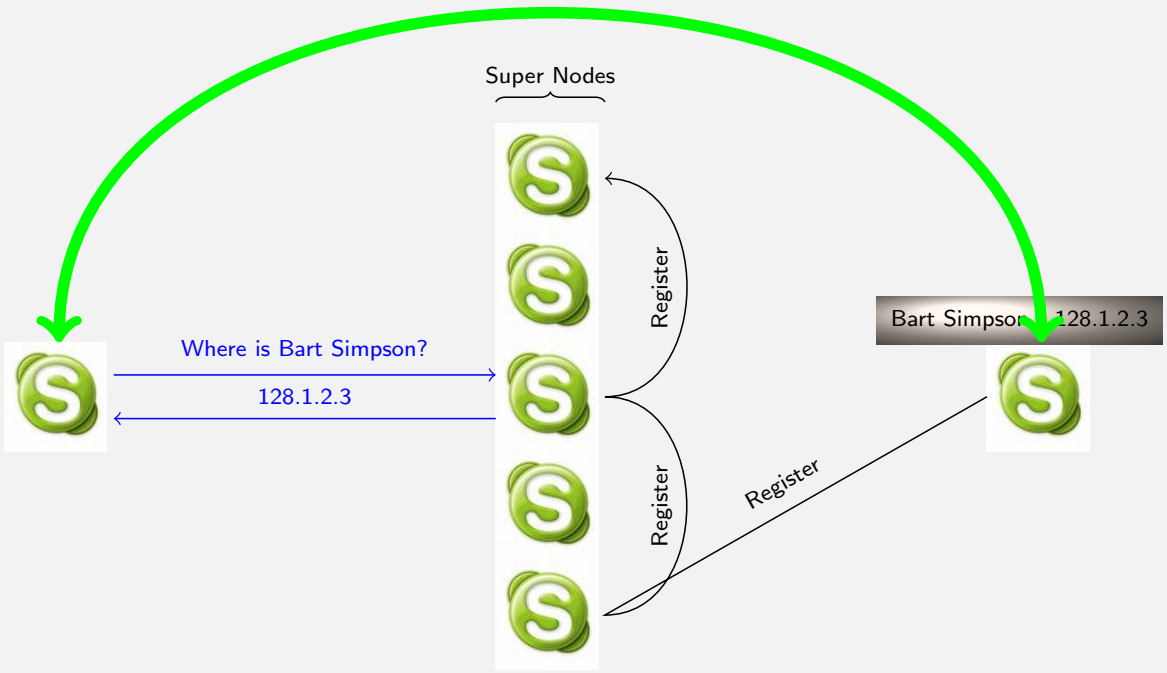


NAT Traversal and Peer To Peer

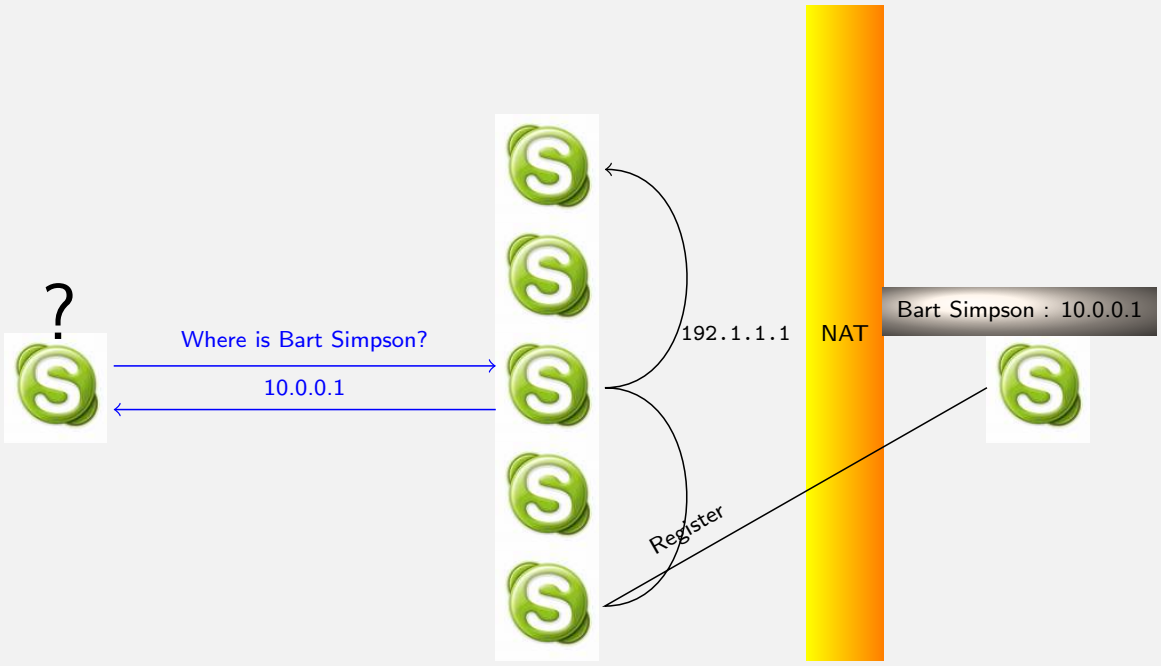


NAT Traversal and Peer To Peer

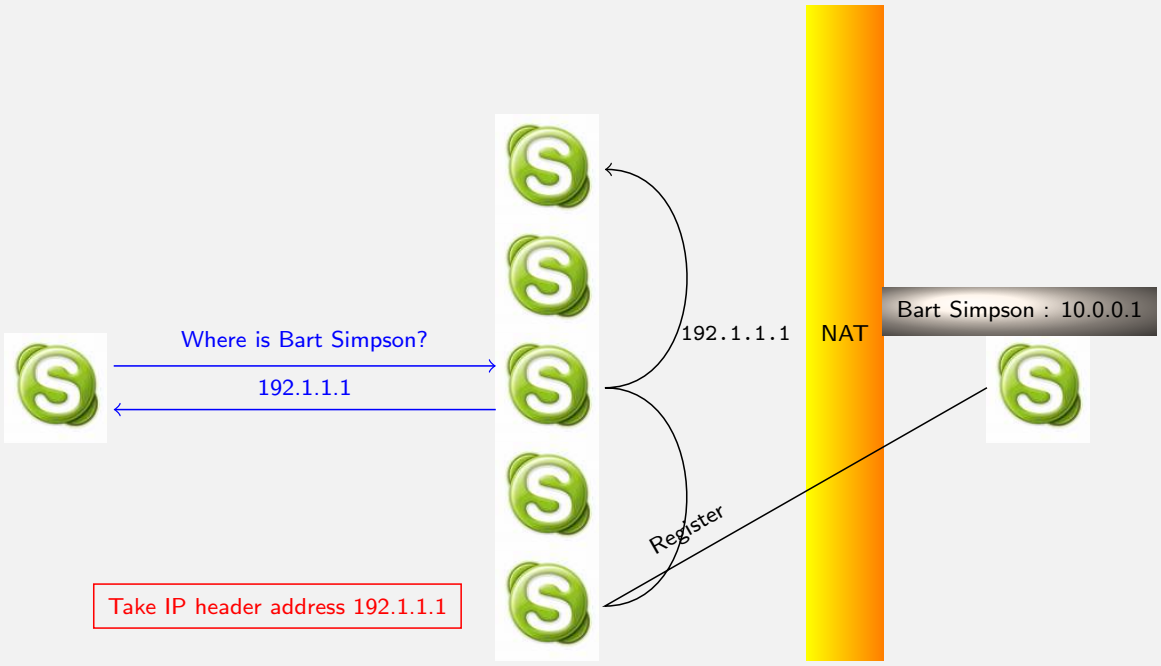
Facts on Addresses ▶ NAT



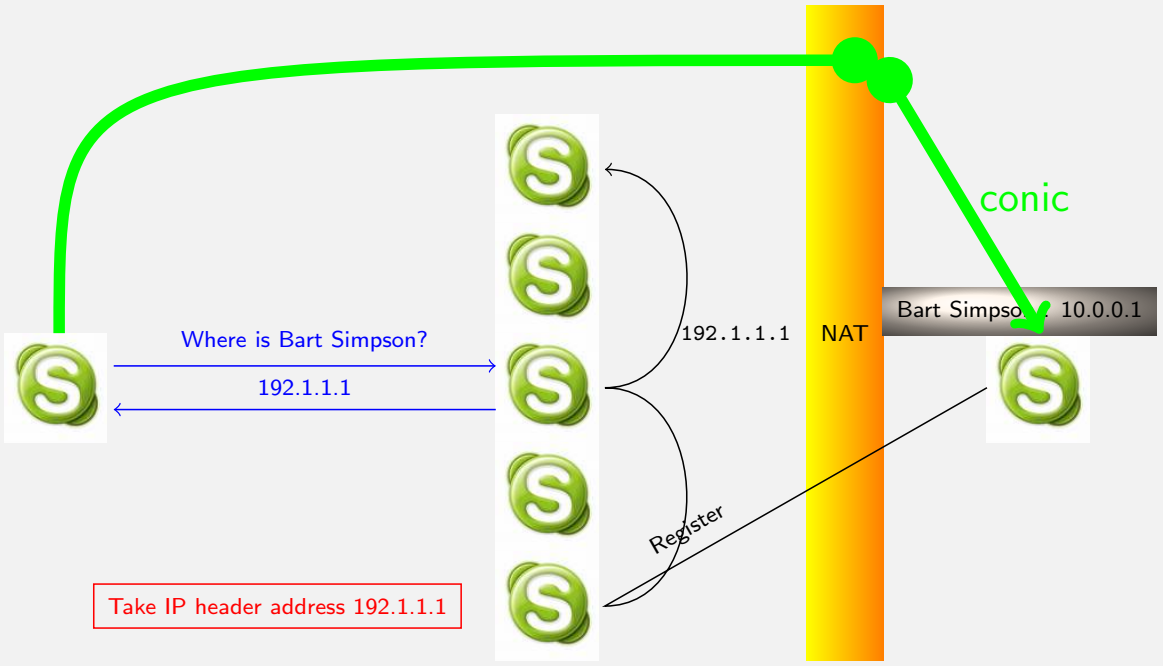
NAT Traversal and Peer To Peer



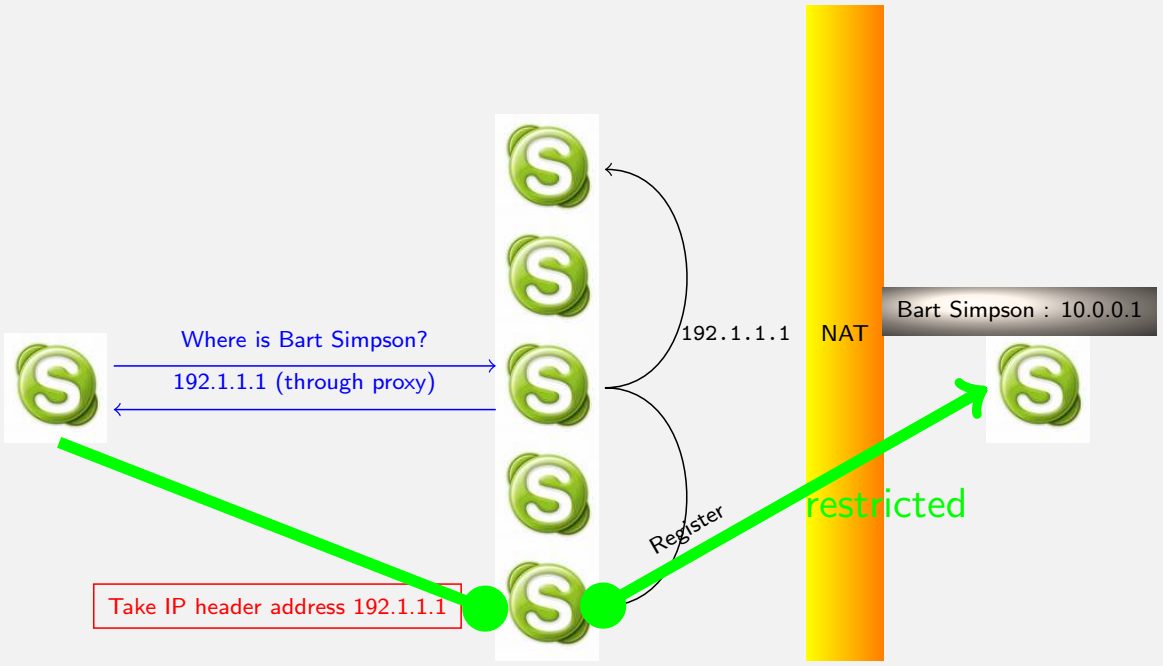
NAT Traversal and Peer To Peer



NAT Traversal and Peer To Peer



NAT Traversal and Peer To Peer



Facts on Addresses

Prefixes delegation

What Has Changed

Classful Addressing

1. Ensure uniqueness
2. Facilitate administrative allocation
   - One central entity

Class-Less (CIDR)

1. Facilitate administrative allocation (hierarchical)
   - Nowadays 5 regional entities
2. Facilitate host location in the network
3. Allocate the minimum pool of addresses

TELECOM Bretagne

CIDR Administrative Point of View

- A hierarchy of administrative registries
  - IANA/ICANN at the top
- 5 Regional Internet Registries (RIR)
  - APNIC (Asia Pacific Network Information Centre)
  - ARIN (American Registry for Internet Numbers)
  - LACNIC (Regional Latin-American and Caribbean IP Address Registry)
  - RIPE NCC (RÈseaux IP EuropÈens - Network Coordination Center)
    - Europe, Middle east.
  - AfriNIC (Africa)
- Providers get prefixes allocation from RIR

RIR Regions

Exhaustion of IPv4 Prefix Pool

- IANA Unallocated Address Pool Depleted: February, 1st 2011
  - See: http://www.nro.net/news/ipv4-free-pool-depleted
- RIR Unallocated Address Pool Exhaustion Forecasts: Start from May 2011
  - See: http://www.potaroo.net/tools/ipv4/
  - See als: http://www.ipv4depletion.com/

IPv6 Benefits

- Larger address space from 2^{32} to 2^{128}
  - Permanent address
- Stateless auto-configuration of hosts
  - Layer 3 "Plug & Play" Protocol
- Simple header $\Rightarrow$ Efficient routing
  - No checksum
  - No fragmentation by routers
  - Enhanced extension system
- end to end, but...
- ~~Quality of service~~
- ~~Better support of mobility~~
- ~~IPsec~~

Addresses

Notation

IPv6 addresses

F2C:544:9E::2:EF8D:6B7 F692:: A:1455::A:6E0 D:63:D::4:3A:55F B33:C::F2 7:5059:3D:C0::

9D::9BAC:B8CA:893F:80 1E:DE2:4C83::4E:39:F35:C875 2:: A:FDE3:76:B4F:D9D:: D6::

369F:9:F8:DBF::2 DD4:B45:1:C42F:BE6:75:: 9D7B:7184:EF::3FB:BF1A:D80 FE9::B:3

EC:DB4:B:F:F11::E9:090 83:B9:08:B5:F:3F:AF:B84 E::35B:8572:7A3:FB2 99:F:9:8B76::BC9

D64:07:F394::BDB:DF40:08EE:A79E AC:23:5D:78::233:84:8 F0D:F::F4EB:0F:5C7 E71:F577:ED:E:9DE8::

B::3 1D3F:A0AA:: 70:8EA1::8:D5:81:2:F302 26::8880:7 93:: F::9:0 E:2:0:266B::

763E:C:2E:1EB:F6:F4:14:16 E6:6:F4:B6:A888:979E:D78:09 9:754:5:90:0A78:A1A3:1:7 2:8::

97B:C4::C36 A40:7:5:7E8F:0:32EC:9A:D0 8A52::575 D::4CB4:E:2BF:5485:8CE 07:5::41 6B::A9:C

94FF:7B8::D9:51:26F 2::E:AE:ED:81 8241:: 5F97:: AD5B:259C:7DB8:24:58:552A:: 94:4:9FD:4:87E5::

5A8:2FF:1::CC EA:8904:7C:: 7C::D6B7:A7:B0:8B DC:6C::34:89 6C:1::5 7B3:6780:4:B1::E586

412:2:5E1:6DE5:5E3A:553:3:: 7F0:: B39::1:B77:DB 9D3:1F1:4B:3:B4E6:7681:09:D4A8 61:520::E0

1:28E9:0:095:DF:F2:: 1B61:4::1DE:50A 34BC:99::E9:9EFB E:EF:: BDC:672A:F4C8:A1::4:7:9CB7

C697:56AD:40:8:0::62

Don't Worry

Addresses are not random numbers. . . they are often easy to handle and even to memorize sometimes

Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:0db8:beef:0001:0000:0000:cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:db8:beef:1:0:0:cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Notation

Addresses ▶ Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:db8:beef:1::cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Notation

- Base format (a 16-octet Global IPv6 Address):
  - 2001:0db8:beef:0001:0000:0000:cafe:deca
- Compact Format:

2001:db8:beef:1::cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by ::

- an IPv4 address may also appear : ::ffff:192.0.2.1

Warning:

2001:db8:3::/40 is in fact 2001:db8:0003::/40 and not 2001:db8:0300::/40

Comments I

La représentation textuelle d'une adresse IPv6 se fait en découpant le mot de 128 bits de l'adresse en 8 mots de 16 bits séparés par le caractère :, chacun d'eux étant représenté en hexadécimal. Par exemple :
2001:0db8:0000:0000:0400:a987:6543:210f
Dans un champ, il n'est pas nécessaire d'écrire les zéros placés en tête :
2001:db8:0:0:400:a987:6543:210f
En outre plusieurs champs nuls consécutifs peuvent être abrégés par ´::'. Ainsi l'adresse précédente peut s'écrire comme suit :
2001:db8::400:a987:6543:210f
Naturellement, pour éviter toute ambiguïté, l'abréviation ´::[a] ne peut apparaître qu'une fois au plus dans une adresse. Les cas extrêmes sont l'adresse indéfinie (utilisée pour désigner les routes par défaut) à tous les bits à zéro et qui se note de manière compacte :
::
et l'adresse de bouclage (loopback) en IPv6, équivalent de l'adresse 127.0.0.1 en IPv4, dont tous les bits sont à zéro sauf le dernier et qui s'écrit :
::1
La représentation des préfixes IPv6 est similaire à la notation CIDR RFC 1519 utilisée pour les préfixes IPv4. Un préfixe IPv6 est donc représenté par la notation :
adresse-ipv6/longueur-du-préfixe-en-bits
Les formes abrégées avec ´::[a] sont autorisées.
2001:0db8:7654:3210:0000:0000:0000:0000/64 2001:db8:7654:3210:0:0:0:0/64 2001:db8:7654:3210::/64
Le seul piège de cette notation vient des longueurs de préfixes qui ne sont pas en frontière de ´:[a]. Ainsi le préfixe 3edc:ba98:7654:3::/56 équivaut en réalité à 3edc:ba98:7654:0000::/56 car il s'écrit 3edc:ba98:7654:0003::/56.
On peut combiner l'adresse d'une interface et la longueur du préfixe réseau associé en une seule notation.
2001:db8:7654:3210:945:1321:abA8:f4e2/64

Comments II

Ces représentations peuvent apparaître beaucoup plus complexes qu'avec IPv4, mais leur attribution répond à des règles strictes, ce qui favorise leur mémorisation.
Dans certains cas, une adresse (voire plusieurs adresses) IPv4 peut être contenue dans une adresse IPv6. Pour les faire ressortir, la notation classique d'IPv4 peut être utilisée au sein d'une adresse IPv6. Ainsi :
`::192.0.2.1` représente une adresse IPv6 composée de 96 bits à 0 suivit des 32 bits de l'adresse IPv4 `192.0.2.1`
Il est pourtant parfois nécessaire de manipuler littéralement des adresses IPv6. Le caractère ":" utilisé pour séparer les mots peut créer des ambigultés. C'est le cas avec les URL où il est aussi utilisé pour indiquer le numéro de port. Ainsi l'URL
`http://2001:db8:12::1:8000/`
pourrait aussi bien indiquer le port 8000 sur la machine ayant l'adresse IPv6 `2001:db8:12::1`, que la machine ayant l'adresse `2001:db8:12::1:8000` en utilisant le port par défaut (80). Pour lever cette ambigulté, le RFC 2732 propose d'inclure l'adresse IPv6 entre "[ ]". L'URL précédente s'écrirait :
`http://[2001:db8:12::1]:8000/`
ou
`http://[2001:DB8:12::1:8000]/`
suivant les cas. Cette représentation peut être étendue à d'autres domaines comme X-window ou au protocole de signalisation téléphonique SIP.

Is it enough for the future ?

- Address length
  - About $3.4x10^{38}$ addresses
  - 60 000 trillion trillion addresses per inhabitant on earth
  - Addresses for every grain of sands in the world
  - IPv4: 6 addresses per US inhabitant, 1 in Europe, 0.01 in China and 0.001 in India
- Justification of a fixed-length address

Warning:

- An address for everything on the network and not an address for everything
- No addresses for the whole life:
  - Depends on your position on the network
  - ISP Renumbering may be possible

Is it enough for the future ?

- Hop Limit:
  - Should not be a problem
  - Count the number of routers used to reach a destination
  - Growth will be in-width more than in-depth
- Payload Length
  - 64 Ko is not a current hard limit
  - Ethernet is limited to 1.5 Ko, evolution can use until 9Ko.
  - Use Jumbogram for specific cases

Addresses

Addressing scheme

Addressing scheme

- RFC 4291 defines current IPv6 addresses
  - loopback (::1)
  - link local (fe80::/10)
  - global unicast (2000::/3)
  - multicast (ff00::/8)
- Use CIDR principles:
  - Prefix / prefix length notation
  - 2001:db8:face::/48
  - 2001:db8:face:bed:cafe:deca:dead:beef/64
- Interfaces have several IPv6 addresses
  - at least a link-local and a global unicast addresses

Comments I

IPv6 reconnaît trois types d'adresses : unicast, multicast et anycast. Le premier de ces types désigne une interface unique. Un paquet envoyé à une telle adresse, sera donc remis à l'interface ainsi identifiée. Parmi les adresses unicast, on peut distinguer celles qui auront une portée globale, c'est-à-dire désignant sans ambigulté une machine sur le réseau Internet et celles qui auront une portée locale (lien ou site). Ces dernières ne pourront pas être routées sur l'Internet.
Une adresse de type multicast désigne un groupe d'interfaces qui en général appartiennent à des noeuds différents pouvant être situés n'importe oˇ dans l'Internet. Lorsqu'un paquet a pour destination une adresse de type multicast, il est acheminé par le réseau à toutes les interfaces membres de ce groupe.
Il faut noter qu'il n'y a plus d'adresses de type broadcast comme sous IPv4 ; elles sont remplacées par des adresses de type multicast qui saturent moins un réseau local constitué de commutateurs. L'absence de broadcast augmente la résistance au facteur d'échelle d'IPv6 dans les réseaux commutés.
Le dernier type, anycast, est une officialisation de propositions faites pour IPv4 RFC 1546 . Comme dans le cas du multicast, une adresse de type anycast désigne un groupe d'interfaces, la différence étant que lorsqu'un paquet a pour destination une telle adresse, il est acheminé à un des éléments du groupe et non pas à tous. C'est, par exemple, le plus proche au sens de la métrique des protocoles de routage. Cet adressage est principalement expérimental.

Une interface possèdera généralement plusieurs adresses IPv6. En IPv4 ce comportement est exceptionnel, il est

banalisé en IPv6.

Addresses

Address Format

Addressing Space Utilization


```
0000::/8 Reserved by IETF [RFC4291]
0100::/8 Reserved by IETF [RFC4291]
0200::/7 Reserved by IETF [RFC4048]
0400::/6 Reserved by IETF [RFC4291]
0800::/5 Reserved by IETF [RFC4291]
1000::/4 Reserved by IETF [RFC4291]
2000::/3 Global Unicast [RFC4291]
4000::/3 Reserved by IETF [RFC4291]
6000::/3 Reserved by IETF [RFC4291]
8000::/3 Reserved by IETF [RFC4291]
a000::/3 Reserved by IETF [RFC4291]
c000::/3 Reserved by IETF [RFC4291]
e000::/4 Reserved by IETF [RFC4291]
f000::/5 Reserved by IETF [RFC4291]
F800::/6 Reserved by IETF [RFC4291]
fc00::/7 Unique Local Unicast [RFC4193]
fe00::/9 Reserved by IETF [RFC4291]
fe80::/10 Link Local Unicast [RFC4291]
fec0::/10 Reserved by IETF [RFC3879]
ff00::/8 Multicast [RFC4291]
```

http://www.iana.org/assignments/ipv6-address-space

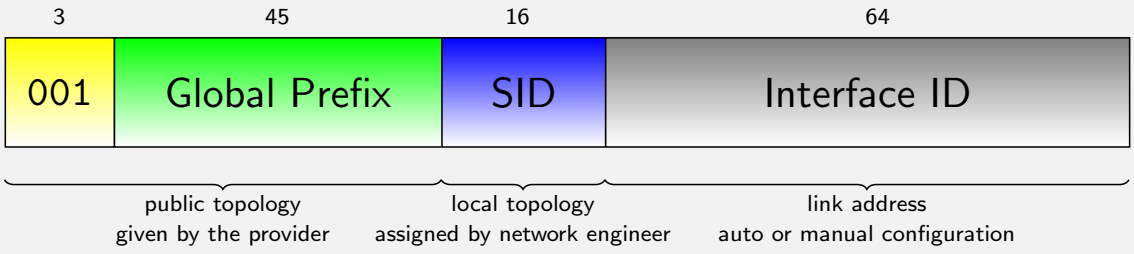
Comments I

Certains types d'adresses sont caractérisés par leur préfixe RFC 4291 . Le tableau suivant (source : `http://www.iana.org/assignments/ipv6-address-space`) donne la liste de ces préfixes. La plage ´réservée[a] du préfixe `0::/8` est utilisée pour les adresses spéciales (adresse indéterminée, de bouclage, mappée, compatible). On notera que plus de 70% de l'espace disponible n'a pas été alloué, ce qui permet de conserver toute latitude pour l'avenir.

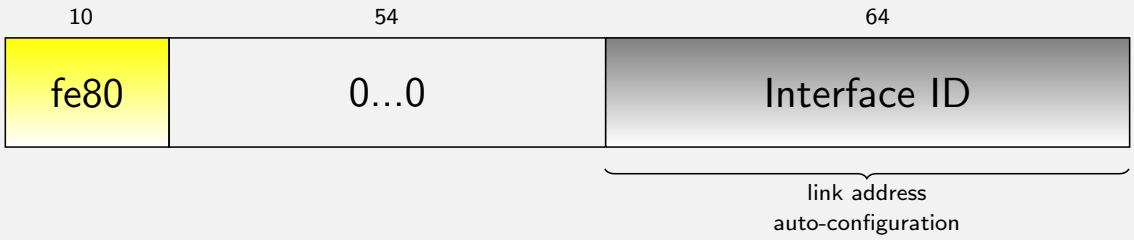
- Glogal Unicast: adresses point-à-point équivalent des adresses publics en IPv4
- Link-Local : utllisable uniquement sur le link (non routable), utilisée principalement pendant la période de bootstrap
- Multicast: equivalent aux classes D d'IPv4
- ULA: équivalent aux adresses privées en IPv4

Address Format

Global Unicast Address:



Link-Local Address:



Comments I

Ce plan, proposée dans le RFC 3587 , précise la structure d'adressage IPv6 définie dans le RFC 4291 en précisant les tailles de chacun des blocs. Il est géré de la même manière que CIDR en IPv4. Une adresse intègre trois niveaux de hiérarchie :

- une topologie publique (appelée "'Global Prefix"') codé sur 48 bits, allouée par le fournisseur d'accès;
- une topologie de site codé sur 16 bits (appelée "'Subnet ID"'). Ce champ permet de coder les numéros de sous réseau du site;
- un identifiant d'interface sur 64 bits (appelé "'Interface ID"') distinguant les différentes machines sur le lien.

Les adresses de type lien-local ("link local use address") sont des adresses dont la validité est restreinte à un lien, c'est-à-dire l'ensemble de interfaces directement connectées sans routeur intermédiaire : par exemple machines branchées sur un même Ethernet, machines reliées par une connexion PPP, ou extrémités d'un tunnel. Les adresses lien-local sont configurées automatiquement à l'initialisation de l'interface et permettent la communication entre noeuds voisins. L'adresse est obtenue en concaténant le préfixe `fe80::/64` aux 64 bits de l'Identifiant d'interface—identifiant d'interface. L'identifiant d"interface est généralement basé sur l'adresse MAC. Cela ne pose pas de problème de respect de le vie privée car, contrairement aux adresses globales, les adresses lien-local ne sortent jamais du réseau où elles sont utilisées.
Ces adresses sont utilisées par les protocoles de configuration d'adresse globale, de découverte de voisins ("neighbor discovery") et de découverte de routeurs ("router discovery"). Ce sont de nouveaux dispositifs, le premier supplantant en particulier le protocole ARP ("Address Resolution Protocol"), qui permettent pas à un réseau local de se configurer automatiquement. Elles sont également largement utilisées par les protocoles de routage soit pour l'échange de données (cf. RIPng, OSPFv3), soit dans les tables de routage puisque le champ prochain routeur est toujours un équipement directement accessible sur le lien.
Un routeur ne doit en aucun cas retransmettre un paquet ayant pour adresse source ou destination une adresse de type lien-local.

SID Values

- 16-bit length up to 65 535 subnets
  - Large enough for most companies
  - Too large for home network ?
  - May be a /56 or /60 GP will be allocated depending on the ISP
- There is no strict rules to structure SID:
  - sequencial : 1, 2, ...
  - use VLAN number
  - include usage to allow filtering, for instance, for a University:

4bits : Community	**8bits**	**4bits**
0 : Infrastructure	*Specific addresses*	
1 : Tests	*Specific addresses*	
6 : Point6	*Managed by Point6*	
8 : Wifi guests	*Specific addresses*	
A : Employees	Entity	Sub-Network
E : Students	Entity	Sub-Network
F : Other (Start-up, etc.)	*Specific addresses*	

Comments I

Il n'existe pas de règles pour allouer les identificateurs de sous-réseau au sein d'un site. Plusieurs techniques (non exclusives) peuvent être utilisées :
numéroter de manière incrémentale les sous-réseaux: 0001, 0002, ... Cette technique est simple a mettre en œuvre dans des réseaux expérimentaux, mais elle peut conduire à un plan d'adressage à plat difficile à mémoriser. Elle peut être utilisée par exemple pour un sous-réseau dédié aux serveur pour simplifier l'écriture et la mémorisation des adresses. utiliser le numéro de VLAN. Elle permet d'éviter de mémoriser plusieurs niveau de numérotation. séparer les types de réseaux et utiliser les chiffres de gauche pour les désigner. Cette technique permet de faciliter les règles de filtrage, tout en utilisant des règles appropriées pour à la gestion de ces sous-réseau pour la partie de droite. A titre d'exemple, le tableau suivant contient le plan de numérotation d'une université localisée sur plusieurs sites prenant en compte les différentes communautés d'utilisateurs :
Ainsi, le préfixe:

- 2001:DB8:1234::/52 servira pour la création de l'infrastructure, donc en particulier les adresses des interfaces des routeurs seront pris dans cet espace,
- 2001:DB8:1234:8000::/52 servira pour le réseau wifi des invités. La manière dont sont gérés les 12 bits restants du SID ne sont pas spécifiés,
- 2001:DB8:1234:E000::/52 servira pour le réseau des étudiants. L'entité représente la localisation géographique du campus. Dans chacun de ces campus, il sera possible d'avoir jusqu'à 16 sous-réseaux différents pour cette communauté.

Interface Identifier

Interface ID can be selected differently

- Derived from a Layer 2 ID (I.e. MAC address) :
  - for Link Local address
  - for Global Address : plug-and-play hosts
- Assigned manually :
  - to keep same address when Ethernet card or host is changed
  - to remember easily the address
    - 1, 2, 3, ...
    - last digit of the v4 address
    - the IPv4 address (for nostalgic system administrators)
    - ...

Interface Identifier

Interface ID can be selected differently

- Random value :
  - Changed frequently (e.g, every day, per session, at each reboot...) to guarantee anonymity
- Hash of other values (experimental) :
  - To link address to other properties
  - Public key
  - List of assigned prefixes
  - . . .

Comments I

Si initialement pour des raisons d'auto-configuration, l'identifiant d'interface devait toujours être dérivé de l'adresse de niveau 2, c'est de moins en moins le cas. Il existe plusieurs méthodes pour construire cette valeur de 64 bits:

- manuelle,
- basée sur l'adresse de niveau 2 de l'interface,
- aléatoire,
- cryptographique.

Manuel
Pour les serveurs les plus utilisé, il est préférable d'assigner manuellement des adresses aux interfaces, car dans ce cas l'adresse IPv6 est facilement mémorisable, et le serveur peut être accessible même si le DNS n'est pas actif. Il existe plusieurs techniques plus ou moins mnémotechniques :
* incrémenter l'identifiant d'interface à chaque nouveau serveur créé
2001:DB8:1234:1::1
2001:DB8:1234:1::2
...
* reprendre le dernier octet de l'adresse IPv4 comme identifiant d'interface. Par exemple si un serveur a comme adresse IPv4 ¡tt¿192.0.2.123¡/tt¿, son adresse IPv6 sera :
2001:DB8:1234:1::7B
ou plus simplement
2001:DB8:1234:1::123
* reprendre l'adresse IPv4 comme identifiant d'interface, bien que cela ait l'inconvénient de conduire à des adresses plus longues à taper :
2001:DB8:1234:1::192.0.2.123

Comments II

Dérivé de l'adresse de l'interface
L'avantage d'utiliser une adresse de niveau 2 pour construire un identifiant d'interface est que l'unicité de cette valeur est presque toujours assurée. En plus, cette valeur est stable tant que la carte réseau de la machine n'est pas changée. Par contre, ces valeurs sont difficilement mémorisables.
Les adresses lien-local sont construites en utilisant ce type d'identifiant. Par contre pour les adresses globales, il est conseillé de ne les utiliser que pour les machines client et de préférer les identifiant d'interface manuel pour les serveur.
Ces identifiants d'interface étant stable dans le temps, à chaque fois qu'un individu change de réseau, il change de préfixe, mais garde le même identifiant d'interface. il pourrait donc servir à tracer les déplacements d'un individu. Le risque est faible, car les cookies mis en place par les serveurs web sont bien plus efficaces, mais ils ne s'agit plus d'un problème réseau. Autre désavantage, comme les adresses MAC contiennent l'identification du matériel, il est possible d'indiquer à l'extérieur du réseau quel type de matériel est utilisé et donner des indications.
Si ces inconvénients sont jugés important par l'entreprise, l'identifiant d'interface pour les adresses globales peut être généré aléatoirement.
Valeur aléatoire
L'identifiant d'interface basé sur des adresses MAC, comme indiqué précédemment, pourrait poser des problèmes pour la vie privée. Il identifie fortement la machine d'un utilisateur, qui même s'il se déplace de réseau en réseau garde ce même identifiant. Il serait alors possible de traquer un individu utilisant un portable, chez lui, au bureau, lors de ses déplacements. Ce problème est similaire à l'identificateur placé dans les processeurs Pentium III.
Pour couper court à toute menace de boycott d'un protocole qui ´menacerait la vie privée[a], il a été proposé d'autres algorithmes de construction d'un identifiant d'interface basé sur des tirages aléatoires (voir RFC 3041).
Un utilisateur particulièrement méfiant pourrait valider ces mécanismes. L'identifiant d'interface est soit choisi aléatoirement, soit construit par un algorithme comme MD5 à partir des valeurs précédentes, soit tiré au hasard si l'équipement ne peut pas mémoriser d'information entre deux démarrages. Périodiquement l'adresse est mise dans

Comments III

l'état ´déprécié[a] et un nouvel identifiant d'interface est choisi. Les connexions déjà établies continuent d'utiliser l'ancienne valeur tandis que les nouvelles connexions utilisent la nouvelle adresse.
Cette solution a été adoptée par Microsoft. Dans Windows XP, l'interface possède deux adresses IPv6 globale. La première a un identifiant d'interface dérivé de l'adresse MAC. Elle sert aux applications attendant des connexions sur la machine (i.e. les applications serveur). Cette adresse est stable et peut être publiée dans le DNS. La seconde possède un identifiant d'interface tiré aléatoirement. Elle est changée tous les jours et sert aux applications client. Dans Windows Vista, ce comportement est généralisé car l'identifiant d'interface de l'adresse permanente est également issu d'un tirage aléatoire. Cela permet d'éviter de donner la marque de la machine ou le type de carte contenu dans les premiers octets de l'identifiant d'interface.
Bien entendu pour que ces mécanismes aient un sens, il faut que l'équipement ne s'enregistre pas sous un même nom dans un serveur DNS inverse ou que l'enregistrement de cookies dans un navigateur Web pour identifier l'utilisateur soit impossible.
En contre partie, il est plus difficile à un administrateur réseau de filtrer les machines puisque celles-ci changent périodiquement d'adresses.
Cryptographique
Encore un sujet de recherche
L'usage de ces adresses n'est pas encore généralisé. Shim6 pour la gestion de la multi-domiciliation ou SEND pour sécuriser la découverte de voisins y on recours.

Si un identifiant aléatoire permet de rendre beaucoup plus anonyme la source du paquet, des propositions sont

faites à l'IETF pour lier l'identifiant d'interface à la clé publique de l'émetteur du paquet. Le RFC 3972 définit le

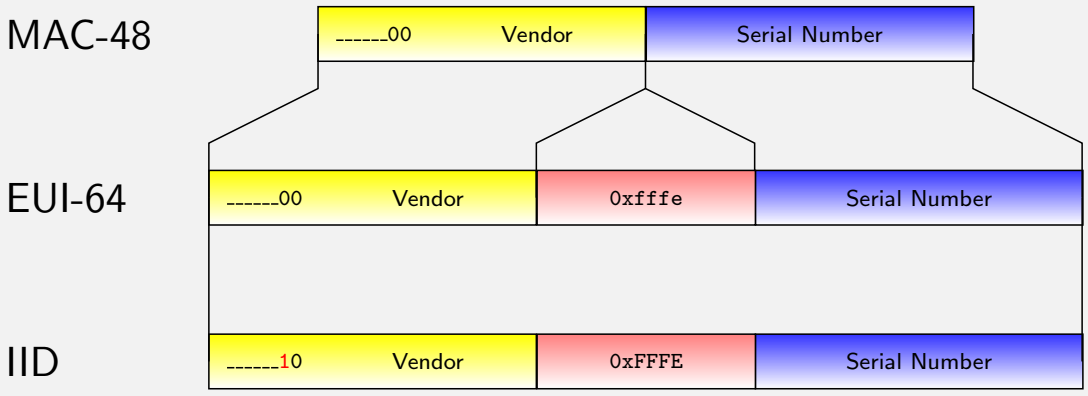
principe de création de l'identifiant d'interface (CGA : Cryptographic Generated Addresses) à partir de la clé

Comments IV

publique de la machine. Elles pourraient servir pour sécuriser les protocoles de découverte de voisins ou pour la gestion de la multi-domiciliation.

How to Construct an IID from MAC Address

- 64 bits is compatible with EUI-64 (i.e. IEEE 1394 FireWire, ...)
- IEEE propose a way to transform a MAC-48 to an EUI-64
- U/L changed for numbering purpose



- There is no conflicts if IID are manually numbered: 1, 2, 3, ...

Comments I

L'avantage d'utiliser une adresse de niveau 2 pour construire un identifiant d'interface est que l'unicité de cette valeur est presque toujours assurée. En plus, cette valeur est stable tant que la carte réseau de la machine n'est pas changée. Par contre, ces valeurs sont difficilement mémorisables.
Les adresses lien-local sont construites en utilisant ce type d'identifiant. Par contre pour les adresses globales, il est conseillé de ne les utiliser que pour les machines client et de préférer les identifiant d'interface manuel pour les serveur.
Ces identifiants d'interface étant stable dans le temps, à chaque fois qu'un individu change de réseau, il change de préfixe, mais garde le même identifiant d'interface. il pourrait donc servir à tracer les déplacements d'un individu. Le risque est faible, car les cookies mis en place par les serveurs web sont bien plus efficaces, mais ils ne s'agit plus d'un problème réseau. Autre désavantage, comme les adresses MAC contiennent l'identification du matériel, il est possible d'indiquer à l'extérieur du réseau quel type de matériel est utilisé et donner des indications.
Si ces inconvénients sont jugés important par l'entreprise, l'identifiant d'interface pour les adresses globales peut être généré aléatoirement.

EUI-64

L'IEEE a défini un identificateur global à 64 bits (format EUI-64) pour les réseaux IEEE 1394 (firewire) ou IEEE 802.15.4 (réseau de capteurs) qui vise une utilisation dans le domaine de la domotique. L'IEEE décrit les règles qui permettent de passer d'un identifiant MAC codé sur 48 bits à un EUI-64.
Il existe plusieurs méthodes pour construire l'identifiant : HorsTexte—Ordre de transmission—L'ordre des bits ne doit pas porter à confusion. Dans la représentation numérique des valeurs, le premier bit transmis est le bit de poids faible, c'est-à-dire le bit de droite. Ainsi sur le support physique le bit g, puis le bit u puis les bits suivants sont transmis.

Comments II

- Si une machine ou une interface possède un identificateur global IEEE EUI-64, celui-ci a la structure décrite figure Identificateur global IEEE EUI-64. Les 24 premiers bits de l'EUI-64, comme pour les adresses MAC IEEE 802, identifient le constructeur et les 40 autres bits identifient le numéro de série (les adresses MAC IEEE 802 n'en utilisaient que 24). Les 2 bits u (septième bit du premier octet) et g (huitième bit du premier octet) ont une signification spéciale :
  - u (Universel) vaut 0 si l'identifiant EUI-64 est universel,
  - g (Groupe) indique si l'adresse est individuelle (g = 0), c'est-à-dire désigne un seul équipement sur le réseau, ou de groupe (g = 1), par exemple une adresse de multicast.
- L'identifiant d'interface à 64 bits est dérivé de l'EUI-64 en inversant le bit u (cf. figure Identificateur d'interface dérivé d'une EUI-64). En effet, pour la construction des adresses IPv6, on a préféré utiliser 1 pour marquer l'unicité mondiale. Cette inversion de la sémantique du bit permet de garder la valeur 0 pour une numérotation manuelle, autorisant à numéroter simplement les interfaces locales à partir de 1.

Comments III

MAC-48
* Si une interface possède une adresse MAC IEEE 802 à 48 bits universelle (cas des interfaces Ethernet ou Wi-Fi). L'adresse est tout d'abord convertie en EUI-64, puis le bit u est mis à 1 comme dans le cas précédent. La figure ci-contre illustre ce processus.
Cas Particuliers
* Si une interface possède une adresse locale unique sur le lien, mais non universelle (par exemple le format d'adresse IEEE 802 sur 2 octets ou une adresse sur un réseau Appletalk), l'identifiant d'interface est construit à partir de cette adresse en rajoutant des 0 en tête pour atteindre 64 bits.

* Si une interface ne possède aucune adresse (par exemple l'interface utilisée pour les liaisons PPP), et si la machine n'a pas d'identifiant EUI-64, il n'y a pas de méthode unique pour créer un identifiant d'interface. La méthode conseillée est d'utiliser l'identifiant d'une autre interface si c'est possible (cas d'une autre interface qui a une adresse MAC), ou une configuration manuelle ou bien une génération aléatoire, avec le bit u positionné à 0. S'il y a conflit (les deux extrémités ont choisi la même valeur), il sera détecté lors de l'initialisation de l'adresse lien-local de l'interface, et devra être résolu manuellement.

Example : Mac / Unix

```
%ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet 127.0.0.1 netmask 0xff000000
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::216:cbff:febe:16b3%en1 prefixlen 64 scopeid 0x5
        inet 192.168.2.5 netmask 0xffffff00 broadcast 192.168.2.255
        inet6 2001:660:7307:6031:216:cbff:febe:16b3 prefixlen 64
        autoconf

        ether 00:16:cb:be:16:b3
        media: autoselect status: active
        supported media: autoselect
```

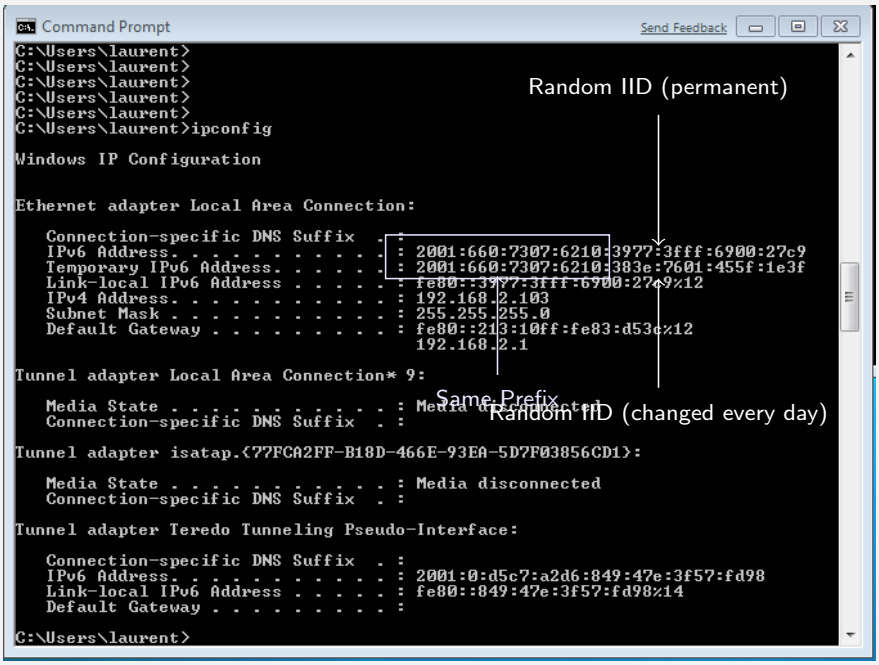
Comments I

L'interface Ethernet en1 possède une adresse IPv4 et deux adresses IPv6 :
La première adresse correspond à l'adresse lien-local. On retrouve l'identifiant d'interface qui suit le préfixe FE80::/64. A noter que l'on retrouve les octets de l'adresse MAC, sauf pour le premier octet qui est à 02 au lieu de 00 suite à l'inversion du bit ´universel/local[a]. A noter que la portée de l'adresse est indiquée par la chaîne de caractère %en1. La valeur scopeid indiquée à la fin de la ligne donne le numéro cette interface.
L'autre adresse correspond à une adresse globale dont le préfixe a éte attribués par l'opérateur :
- 2001 : une adresse unicast globale attribuée par les autorités régionales (cf. Familles d'adressage),
- 660 : est le préfixe attribué par RIPE-NCC au réseau Renater
- 7301 est attribué par Renater à Télécom-Bretagne,
- 6031 : est le numéro du réseau à l'intérieur de l'ENST Bretagne.

On voit ensuite l'adresse MAC qui a servi a construire les identifiants d'interface en mettant à 1 le second bit et en

ajoutant la séquence FFFE au milieu.

Windows 7



Comments I

Traditionnellement, la commande ipconfig permet de connaitre les paramètres des interfaces réseaux.
Ainsi sur cette exemple, l'interface vers le réseau local possède plusieurs adresses IPv6 :
* une adresse lien-local : `fe80::3977:3fff:6900:27c9%12`. Cette adresse contient la porté qui indique que l'interface sur ce système possède le numéro 12.
* une adresse globale permanente :`2001:8db:7307:6210:3977:3fff:6900:27c9` qui sera utilisée par les applications serveur tournant sur cette machine. Sous Vista et Seven, la partie identifiant d'interface est aléatoire comme dans cet exemple, tandis que sous XP, l'identifiant d'interface dérive de l'adresse MAC.
* une adresse globale temporaire: `2001:8db:7307:6210:383e:7601:455f:1e3f`. Les deux adresses globales partagent le même préfixe`2001:8db:7307:6210::/64`

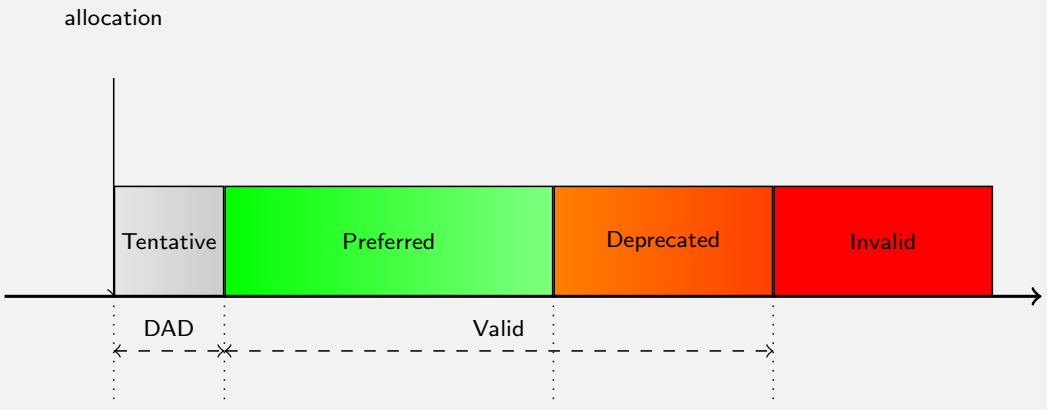
Il est également possible d'utiliser la commande `netsh` pour accéder aux configuration des interfaces et modifier les configurations :

```
 C:>netsh
netsh>interface ipv6
netsh interface ipv6>
```

Par exemple, pour enlever la configuration automatique des adresses à partir des annonces de routeur :

```
 C:>netsh
netsh>interface ipv6

netsh interface ipv6> set interface LAN routerdiscovery=disabled
```


Address Lifetime



Comments I

IPv6 généralisant le plan d'adressage CIDR, les préfixes restent dans tous les cas la propriété des opérateurs. Il ne peuvent plus être attribués "à vie" aux équipements. Pour faciliter la renumérotation d'une machine l'attribution d'une adresse à une interface est faite temporairement, les adresses IPv6 ne sont pas données mais prêtées. Une durée de vie est associée à l'adresse qui indique le temps pendant lequel l'adresse appartient à l'interface. Quand la durée de vie est épuisée, l'adresse devient invalide, elle est supprimée de l'interface et devient potentiellement assignable à une autre interface. Une adresse invalide ne doit jamais être utilisée comme adresse dans des communications. La valeur par défaut de la durée de vie d'une adresse est de 30 jours, mais cette durée peut être prolongée, ou portée à l'infini. L'adresse lien-local a une durée de vie illimitée.
La renumérotation d'une interface d'une machine consiste à passer d'une adresse à une autre. Lors d'une renumérotation, il n'est pas souhaitable de changer brusquement d'adresse, sinon toutes les communications TCP, qui l'utilisent comme identificateur de connexion, seraient immédiatement coupées. Ceci entraînerait des perturbations importantes au niveau des applications.

Pour faciliter cette transition, un mécanisme d'obsolescence est donc mis en place pour invalider progressivement une adresse. Ce mécanisme s'appuie sur la capacité d'affectation de plusieurs adresses valides à une même interface. Ensuite pour effectuer le choix de l'adresse à utiliser, un état est associé. Il indique dans quelle phase de sa durée de vie une adresse se situent vis à vis de l'interface. Le premier de ces états est qualifié de préféré : l'utilisation n'est aucunement restreinte. Peu avant son invalidation l'adresse passe dans un état de déprécié. Dans cet état, l'utilisation de l'adresse est déconseillée, mais pas interdite. L'adresse dépréciée ne doit plus être utilisée comme adresse de source pour les nouvelles communications (comme l'établissement de connexion TCP). Par

Comments II

contre l'adresse dépréciée peut encore servir d'adresse de source dans le cas des communications existantes. Les paquets reçus à une adresse dépréciée continuent à être remis normalement. A la durée de vie de validité d'un adresse, il est également associé une durée de vie pour son état préféré. La figure "'Etats successifs d'une adresse sur une interface"' représente les différents états que prend une adresse lorsqu'elle est allouée à une interface.

Addresses

Kind of addresses

Link-Local Scoped Addresses

- Global Address, the prefix designates the exit interface
- Link-Local address, the prefix is always fe80::/10
  - The exit interface is not defined
  - A %iface, can be added at the end of the address to avoid ambiguity
- Example:

```
Routing tables

Internet6:

Destination                     Gateway                         Flags      Netif Expire
default                         fe80::213:c4ff:fe69:5f49%en0    UGSc        en0
```


TELECOM Bretagne

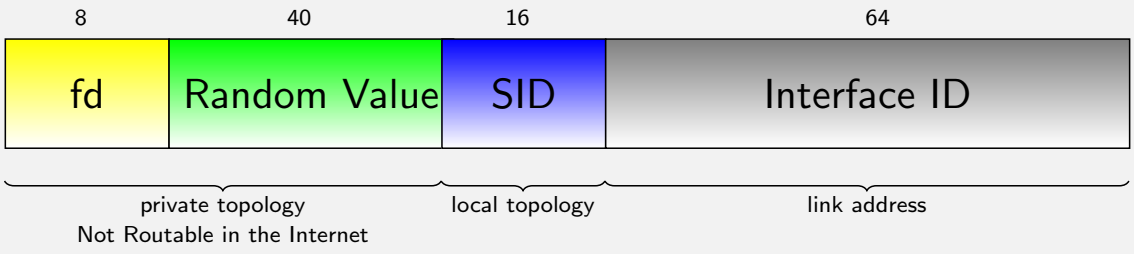
Comments I

Une adresse lien-local (ou multicast) n'indique pas intrinsèquement l'interface de sortie, puisque toutes les interfaces partagent le même préfixe fe80::/10. Il faut donc indiquer de manière explicite sur quelle interface doivent être émis les paquets. Sur certains systèmes d'exploitation (BSD, Mac OS, Windows), il est possible de la spécifier en ajoutant à la fin de l'adresse le nom de l'interface voulue, précédé du caractère "%". Sous Linux, un argument, généralement -I permet de la désigner.

Other kind of addresses : ULA (RFC 4193)

- Equivalent to the private addresses in IPv4
- But try to avoid same prefixes on two different sites:
  - avoid renumbering if two company merge
  - avoid ambiguities when VPN are used
- These prefixes are not routable on the Internet

Unique Local IPv6 Unicast Addresses:



http://www.sixxs.net/tools/grh/ula/ to create your own ULA prefix.

Comments I

Le RFC 4193 définit un nouveau format d'adresse unicast : les adresses uniques locales (ULA : Unique Local Address). Ces adresses sont destinées à une utilisation locale. Elles ne sont pas définies pour être routées dans l'Internet, mais seulement au sein d'une zone limitée telle qu'un site ou entre un nombre limité de sites. Les adresses uniques locales ont les caractéristiques suivantes :

- Prefixe globalement unique.
- Préfixe clairement définit facilitant le filtrage sur les routeurs de bordure.
- Permet l'interconnexion de sites sans générer de conflit d'adresse et sans nécessiter de renumérotation.
- Indépendantes des fournisseurs d'accès à l'Internet et ne nécessitent donc pas de connectivité.
- Pas de conflit en cas de routage par erreur en dehors d'un site.
- Aucune différences pour les applications, qui peuvent les considérer comme des adresses globales unicast standard.

Les adresses uniques locales sont créées en utilisant un identifiant global (Global ID) généré pseudo-aléatoirement. Ces adresses suivent le format suivant :

- Prefix (7 bits) : FC00::/7 préfixe identifiant les adresses IPv6 locales (ULA)
- L (1 bit) : Positionné à 1, le préfixe est assigné localement. La valeur 0 est réservée pour une utilisation future.
- Global ID (40 bits) : Identifiant global utilisé pour la création d'un préfixe unique (Globally Unique Prefix).
- Subnet ID (16 bits) : Identifiant d'un sous réseau à l'intérieur du site.

Comments II

- Interface ID (64 bits) : L'indentifiant d'interface tel que définit dans Identifiant d'interface.

Le site `http://www.sixxs.net/tools/grh/ula/` permet de créer et d'enregistrer son adresse ULA à partir d'une adresse MAC.

Generic Format:

- T (Transient) 0: well known address - 1: temporary address
- P (Prefix) 1 : assigned from a network prefix (T must be set to 1)
- R (Rendez Vous Point) 1: contains the RP address (P & T set to 1)
- Scope :
  - 1 - interface-local
  - 2 - link-local
  - 3 - reserved
  - 4 - admin-local
  - 5 - site-local
  - 8 - organisation-local
  - e - global
  - f - reserved

Comments I

Cette section décrit brièvement le système d'adressage multicast IPv6 et ne s'intéresse qu'aux adresses utilisées localement par les protocoles directements lié à IPv6 (Découverte de voisins, DHCPv6,...). Pour plus de détails sur le multicast en général, se reporter au chapitre Multicast. La figure Structure de l'adresse IPv6 Multicast donne le format de l'adresse IPv6 de multicast décrite dans le RFC 4291 .
Les adresses multicast IPv6 sont dérivées du préfixe FF00::/8. Le champ drapeaux de 4 bits est défini de la manière suivante :

- Seul le bit T (comme Transient) du champ drapeaux est initialement décrit dans le RFC 4291. La valeur 0 indique une adresse multicast bien connue gérée par une autorité. La valeur 1 indique une valeur temporaire.
- Les bits P et R sont décrits dans le RFC 3306 et le draft Internet sur embedded-RP (RFC 3956).
- Le bit de poids fort du champ drapeaux n'est pas encore attribué.

Le champ scope de l'adresse multicast IPv6 permet d'en limiter la portée (scope en anglais). En IPv4, la portée d'un paquet est limitée par le champ TTL (Time To Live), de même des préfixes peuvent être définis pour identifier des adresses à portée réduite. Les valeurs suivantes sont définies :

- 1 - interface-local : Les paquets ne sortent pas de la machine (équivalent du loopback en unicast), cette adresse sert pour la communication entre les applications.
- 2 - link-local : La portée se limite au réseau local, les paquets ne peuvent pas traverser les routeurs multicast. Cette valeur est utilisée en particulier par le protocole de découverte des voisins.
- 3 - réservé

Comments II

- 4 - admin-local
- 5 - site-local
- 8 - organisation-local
- E - global
- Les portées 0 et F sont réservées.

Some Well Known Multicast Addresses

```
ff02:0:0:0:0:0:0:1 All Nodes Address (link-local scope)
ff02:0:0:0:0:0:0:2 All Routers Address
ff02:0:0:0:0:0:0:5 OSPFIGP
ff02:0:0:0:0:0:0:6 OSPFIGP Designated Routers
ff02:0:0:0:0:0:0:9 RIP Routers
ff02:0:0:0:0:0:0:fb mDNSv6
ff02:0:0:0:0:0:1:2 All-dhcp-agents
ff02:0:0:0:0:1:ffxx:xxxx Solicited-Node Address
ff05:0:0:0:0:0:1:3 All-dhcp-servers (site-local scope)
```

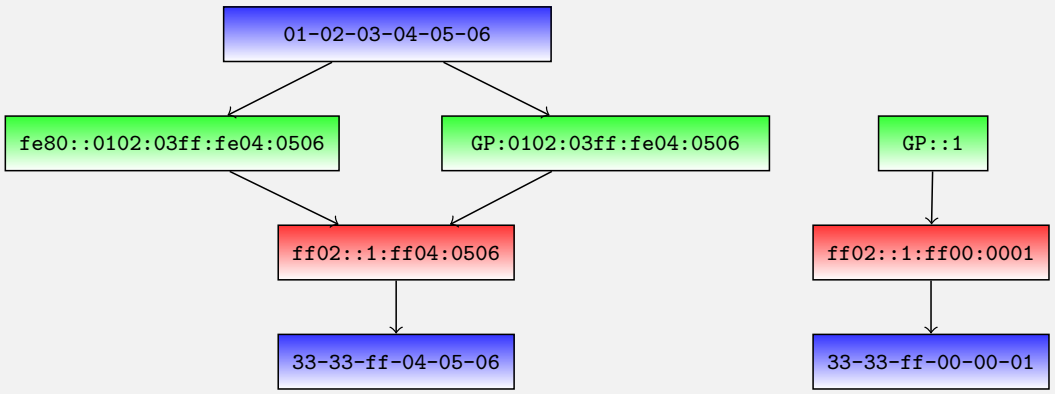
http://www.iana.org/assignments/ipv6-multicast-addresses

Comments I

`http://www.iana.org/assignments/ipv6-multicast-addresses` donne les adresses multicast définies.

Solicited Multicast Addresses

- Derive a Multicast Address from a Unicast Address
  - Widely used for stateless auto-configuration
  - Avoid the use of broadcast



Comments I

IPv6 interdit l'utilisation de la diffusion généralisée (Broadcast) lorsque le Multicast est disponible. Ainsi les protocoles comme Neighbor Discovery, chargés de faire le lien entre les adresses IPv6 et les adresses MAC (à l'instar d'ARP en IPv4) doivent utiliser une adresse de Multicast. Pour être plus efficace, au lieu d'utiliser l'adresse FF02::1 (tous les équipements sur le lien, l'utilisation des adresses de multicast sollicité permet de réduire considérablement le nombre d'équipements qui recevront la requête.
Le transparent montre comment l'on passe d'une adresse IPv6 unicast à une adresse de multicast sollicité. Il s'agit de prendre les 3 derniers octets de l'adresse unicast que l'on concatène avec le préfixe IPv6 multicast FF02::1:FF00::/96.
Dans l'exemple, les deux adresses dérivant d'une adresse MAC conduisent à la même adresse de multicast sollicité, tandis que la configuration manuelle d'une interface conduit à la construction d'une autre adresse de multicast sollicité. On peut noter que le risque que deux machines sur un lien aient la même adresse de multicast sollicité est très faible. Pour celle dérivant d'une adresse MAC, il faudrait que les 3 derniers octets soient identiques, ce qui est impossible chez un même constructeur et la probabilité d'avoir, sur un même lien, des cartes de deux constructeurs différents se terminant par les mêmes 3 derniers octets est très faible. Pour la numérotation manuelle des interfaces, une machine ayant l'adresse GP:::0100:0001 conduirait à construire la même adresse de multicast sollicité FF02::1:FF00:0001, mais cette numérotation manuelle des interfaces n'est pas logique.

L'exemple se poursuit par la transformation de l'adresse de Multicast au niveau IPv6 en adresse de multicast de

niveau 2. Elle est très spécifique à la technologie et à la manière dont est mis en ?uvre le multicast au niveau 2.

Pour les réseaux Ethernet (et dérivés comme le Wi-Fi), les 4 derniers octets de l'adresse multicast sollicité sont

ajoutés au préfixe 33-33.

Example

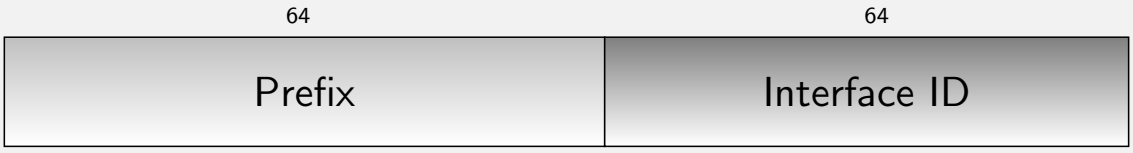

```
Vlan5 is up, line protocol is up
IPv6 is enabled, link-local address is fe80::203:fdff:fed6:d400
Description: reseau C5
Global unicast address(es):
  2001:660:7301:1:203:fdff:fed6:d400, subnet is 2001:660:7301:1::/64
Joined group address(es):
  ff02::1  <- All nodes
  ff02::2  <- All routers
  ff02::9  <- RIP
  ff02::1:ffd6:d400  <- Solicited Multicast
```

Comments I

Cet exemple montre la configuration des interfaces d'un routeur Cisco. Il possède une adresse Lien-Local `FE80::203:FDFF:FED6:D400` et une adresse globale toutes deux basées sur l'adresse MAC, l'adresse de multicast sollicité est donc la même pour ses deux adresses IPv6 `FF02::1:FFD6:D400`. Comme toute machine, il appartient au groupe `FF02::1`. Comme il s'agit d'un routeur, il s'est aussi inscrit à `FF02::2`. Le fait que le protocole de routage RIP soit utilisé, le fait également appartenir au groupe `FF02::9`.

Anycast

- In the same addressing space as unicast
- No way to distiguish them
- Two anycast families:
  - Same prefix on Internet
    - same a IPv4 anycast for DNS or 6to4
  - Same address on the link
    - Must avoid DAD
    - Some IID values are reserved
    - All IID bits to 1 except last byte
    - Only `0x7E` Mobile Home Agent
  - May more addresses with Wireless Sensor Network ?
    - Temperature, presence...



Comments I

Le principe sous-jacent est relativement simple : au lieu d'envoyer un paquet à une interface déterminée, on envoie ce paquet à une adresse (anycast) qui représente un ensemble de machines dans un domaine bien défini. Une adresse anycast permet de désigner un service par une adresse bien connue, de cette manière, il n'est pas nécessaire d'interroger un serveur pour connaître la localisation d'un équipement.
Il existe deux méthodes pour utiliser l'adressage multicast. La première consiste à définir un préfixe "virtuel" qui sera diffusé à plusieurs endroit dans l'Internet. Cette méthode est déjà très utilisée en IPv4 pour accéder à certains serveurs. Ainsi plusieurs serveurs de la racine du DNS (voir http://www.root-servers.org) partagent la même adresse et donc le même préfixe. Ce préfixe est annoncé par plusieurs réseaux. Pour le DNS, cela présente plusieurs avantages :

- La résolution se fait a plus près de l'utilisateur, ce qui réduit les temps de réponse,
- Les attaques par déni de service sont plus difficiles, car il faut que toutes les attaques viennent d'une même zone, sinon elles sont réparties sur plusieurs serveurs.
- Le nombre d'adresses à connaître par les résolveurs est limité.
- Le mécanismes de transition 6to4 utilise également cette technique. Le préfixe 192.88.99.0/24 correspondant à des extremités de tunnels (192.88.99.1) et donnant accès à l'Internet v6 est annoncé a plusieurs endroits dans le réseau. Cela permet de préconfigurer les équipements avec cette adresse et d'offrir à l'utilisateur l'extrémité de tunnel la plus proche. En cas de disparition de celle-ci, le routage de l'Internet aiguillera automatiquement vers un autre équipement tunnelier.

La figure donne la structure de l'adresse anycast dans le cas d'un anycast sur le lien. On y retrouve une partie préfixe et une partie identifiant anycast. La partie préfixe est la même que celle utilisée pour les adresses unicast. Contrairement aux structures d'adresses vue précédemment, la longueur de ce préfixe n'est pas spécifiée, car une adresse anycast doit s'adapter aussi bien aux plans d'adressage actuels (où la longueur est généralement de 64 bits) qu'aux futurs plans qui pourraient avoir des tailles différentes.

Anycast on prefix : Example from Renater


```
#traceroute6 2001:500:2f::f
traceroute6 to 2001:500:2f::f (2001:500:2f::f) from 2001:660:7301:3103:223:6cff:fe97:
30 hops max, 12 byte packets
1 2001:660:7301:3103::1 4.774 ms 1.198 ms 2.764 ms
2 2001:660:7301:3036::1 3.364 ms 2.215 ms 1.417 ms
3 vl856-gi9-9-rennes-rtr-021.noc.renater.fr 2.892 ms 6.794 ms 2.195 ms
4 te4-1-caen-rtr-021.noc.renater.fr 7.706 ms 5.1 ms 4.193 ms
5 te4-1-rouen-rtr-021.noc.renater.fr 6.527 ms 6.296 ms 6.661 ms
6 te0-0-0-1-paris1-rtr-001.noc.renater.fr 8.702 ms 10.26 ms 8.696 ms
7 F-root-server.sfinx.tm.fr 8.495 ms 8.607 ms 8.664 ms
8 f.root-servers.net 8.738 ms 9.171 ms 8.702 ms
```

Anycast on prefix : Example from Hawaï

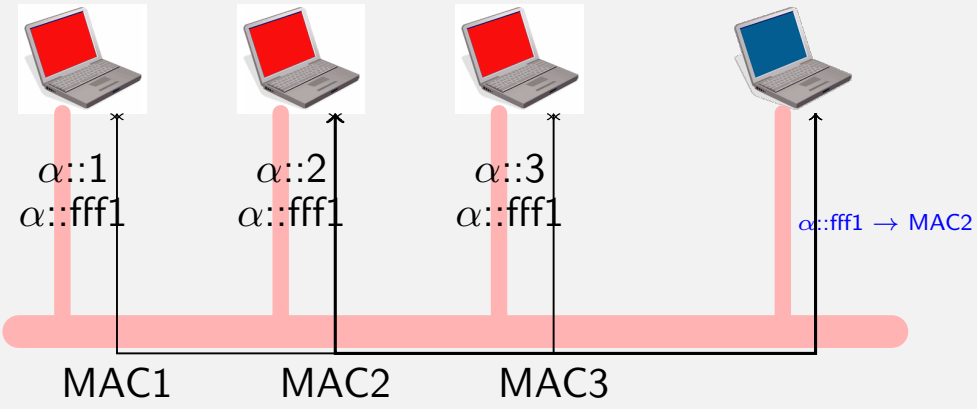

```
#traceroute6 2001:500:2f::f
traceroute6 to 2001:500:2f::f (2001:500:2f::f) from 2001:1888:0:1:2d0:b7ff:fe7d:bed6,
64 hops max, 12 byte packets
1 apapane-fe0-0-1 1.169 ms 0.970 ms 0.947 ms
2 r1.mdtnj.ipv6.att.net 121.159 ms 121.737 ms 121.378 ms
3 bbr01-p1-0.nwrk01.occaid.net 130.468 ms 129.640 ms 130.845 ms
4 bbr01-g1-0.asbn01.occaid.net 131.372 ms 131.596 ms 131.421 ms
5 bbr01-g1-0.atln01.occaid.net 144.937 ms 144.550 ms 144.834 ms
6 bbr01-p1-0.dlls01.occaid.net 166.709 ms 196.177 ms 165.983 ms
7 dcr01-p1-5.lsan01.occaid.net 138.437 ms 138.690 ms 138.544 ms
8 bbr01-g0-2.irvn01.occaid.net 138.552 ms 137.956 ms 137.649 ms
9 dcr01-g1-2.psdn01.occaid.net 137.629 ms 138.030 ms 141.332 ms
10 bbr01-f1-5.snfc02.occaid.net 138.501 ms 138.511 ms 137.483 ms
11 exit.sf-guest.sfo2.isc.org 147.941 ms 144.929 ms 145.956 ms
12 f.root-servers.net 139.063 ms 139.715 ms 142.571 ms
```


Comments I

Ce mécanisme fonctionne de la même manière avec les adresses IPv6. L'exemple suivant l'illustre avec les adresses IPv6 d'un serveur DNS. Vu depuis la France et le réseau de Renater, la route pour atteindre cette machine est la suivante:
Comme l'indique ce traceroute, la machine est localisée au sfinx (noeud d'échange de trafic Internet situé en France).
Un traceroute vers la même destination depuis une machine localisée aux Etats-Unis (Hawaï) donne le résultat suivant:

Le serveur se trouve dans les locaux de l'ISC.

OnLink Anycast: Example



RFC 2526 Anycast values, all bit of IID set to 1 except last 8 bits:

- 0x7F: reserved
- 0x7E: Home Agent (Mobile IP)
- 0x00 to 0x7D: reserved

http://www.iana.org/assignments/ipv6-anycast-addresses

Anycast on prefix : Example from Hawaï


```
# ifconfig en3
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet6 fe80::223:6cff:fe97:679c
inet 192.168.103.177 netmask 0xffffff00 broadcast 192.168.103.255
inet6 2001:660:7301:3103:223:65ff:fe97:679c prefixlen 64 autoconf
ether 00:23:6c:97:67:9c
media: autoselect status: active
supported media: autoselect
# ifconfig en3 inet6 2001:660:7301:3103:FF::FF anycast
# ifconfig en3
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet6 fe80::223:6cff:fe97:679c
inet 192.168.103.177 netmask 0xffffff00 broadcast 192.168.103.255
inet6 2001:660:7301:3103:223:65ff:fe97:679c prefixlen 64 autoconf
inet6 2001:660:7301:3103:ff::ff prefixlen 64 anycast
ether 00:23:6c:97:67:9c
media: autoselect status: active
supported media: autoselect
```

Comments I

Le deuxième type d'adresses anycast correspond à l'allocation de plusieurs adresses identiques sur le même lien. Cette méthode est plus complexe à gérer, car normalement, IPv6 va tester qu'une adresse n'est pas déjà présente gr,ce au mécanisme de détecttion d'adresses dupliquées (DAD). Comme rien ne distingue une adresse anycast d'une adresse unicast, il faut désactiver la détection d'adresses dupliquées en précisant anycast lors de l'affectation de l'adresse à l'interface.

Si le concept anycast est simple dans son principe, son implémentation est autrement délicate. En outre, ce concept n'est encore qu'un sujet de recherche. Enfin un autre argument, de taille, explique cette prudence : il n'y a eu aucune expérience grandeur nature analogue au projet Mbone pour le multicast. Peut être que les réseaux de capteurs pourraient permettre une utilisation a plus grande échelle de l'anycast. Ainsi, à une question du genre "quelle est la température de la pièce?" le premier capteur pouvant répondre à la question enverra la réponse. Les autres capteurs pouvant demeurer silencieux.

Comme les adresses de type sont allouées dans le même espace d'adressage, elles sont créées en attribuant une même adresse unicast à des noeuds distincts, chacun des noeuds devant être configuré pour que l'adresse ainsi attribuée soit de type anycast (sinon on aurait une adresse unicast dupliquée). La seule manière de différencier une adresse anycast d'une adresse multicast est de regarder la partie identifiant anycast qui diffère d'un identifiant d'interface. Ainsi la séquence binaire composée uniquement de 0 a été la première valeur retenue. Elle permet d'identifier un des routeurs du lien. Le RFC 2526 définit des règles de construction d'autres identifiants anycast sur un lien en réservant les 128 identifiants d'interface locaux les plus élevés. Cela permet d'éviter les conflits avec une numérotation manuelle qui partent des identifiants les plus petits vers les plus élevés. Le tableau Allocation des identifiants Anycast donne l'allocation des identifiants utilisés.

On peut remarquer que l'usage des l'anycast n'est pas généralisé puisqu'une seule valeur a été définie et correspond à l'agent mère pour IPv6 mobile. Cela pourrait changer avec l'introduction d'IPv6 dans les réseaux de capteurs. En effet, un des intérêt de l'adressage anycast est de désigner une fonction plus qu'une machine particulière. Avec les

Comments II

réseaux de capteur, on s'interesse plus à une information qu'à l'équipement qui a permis de l'obtenir, par exemple si des capteurs sont interrogés sur la température d'une pièce, le premier à donner l'information peut être suffisant.

Avec les adresses anycast, plusieurs équipements sur un lien physique possèdent une même adresse IPv6. Or il ne s'agit pas d'envoyer la même information à tous ces équipements comme en multicast, mais d'en choisir un seul. Une possibilité consiste à utiliser le mécanisme de découverte de voisins (cf. Découverte de voisins) pour trouver l'association entre l'adresse IPv6 et l'adresse MAC.
La figure page 212 illustre ce fonctionnement. La station A envoie un message de sollicitation de voisin pour déterminer l'adresse MAC de l'équipement. Trois serveurs reçoivent cette requête et répondent. La station A prendra une de ces réponses et dialoguera en point-à-point avec l'équipement choisi.

Comme le correspondant peut varier soit en cas de modification du routage ou de la table de résolution d'adresse, il ne peut pas y avoir d'états. L'Anycast doit être limité à des requêtes simples de type question/réponse ou la gestion de tunnels (bien que cela puisse entraîner un déséquencement). De même les protocoles de niveau 4 orientés connexion comme TCP ne peuvent pas être utilisés.

Protocol

IPv6 Header

IPv6 Packet : Simpler

Definition

- IPv6 header follows the same IPv4 principle:
  - fixed address size ... but 4 times larger
  - alignment on 64 bit words (instead of 32)
- Features not used in IPv4 are removed
- Minimum MTU 1280 Bytes
  - If L2 cannot carry 1280 Bytes, then add an adaptation layer such as AAL5 for ATM or 6LoWPAN (RFC 4944) for IEEE 802.15.4.

Goal :

- Forward packet as fast as possible
- Less processing in routers
- More features at both ends

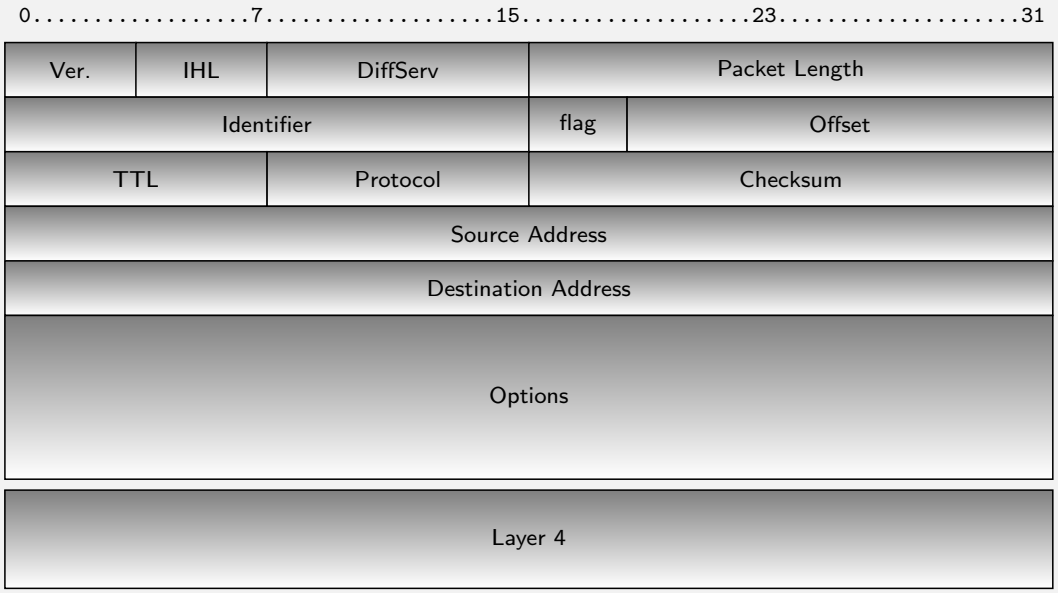
Comments I

Hormis la modification de la taille des adresses, ce qui conduit à une taille d'en-tête de 40 octets (le double de l'en-tête IPv4 sans les options), le protocole IP a subi un toilettage reprenant l'expérience acquise au fil des ans avec IPv4. Le format des en-têtes IPv6 est simplifié et permet aux routeurs de meilleures performances dans leurs traitements :

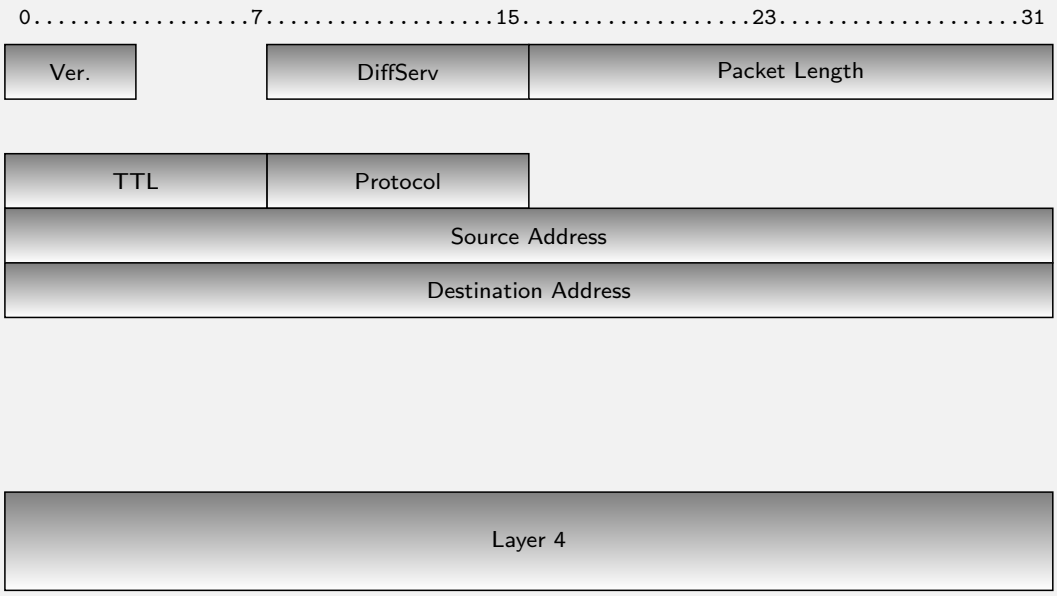
- La taille des adresses a été multipliée par 4.
- Les champs sont alignés sur des mots de 64 bits, ce qui optimise leur traitement, surtout avec les nouvelles architectures à 64 bits.
- La taille minimale des MTU : Maximum Transmission Unit est de 1 280 octets. Le choix de 1 280 comme MTU minimal en IPv6 permet le tunnelage de paquets IPv6. En effet, la taille de 1 500 octets est généralement admise car elle correspond à la valeur imposée par Ethernet. La majorité des autres réseaux offrent une taille supérieure. Pour les réseaux ne le permettant pas, une couche d'adaptation (comme avec les couches d'adaptation AAL d'ATM) ou 6LoWPAN avec les réseaux de capteurs (comme IEEE 802.15.4) devra être mise en oeuvre pour pouvoir transporter les paquets IPv6.

L'idée est de retirer du cœur de réseau les traitements compliqués. Les routeurs ne font que forwarder les paquets vers la destination, les autres traitements (fragmentation, ...) seront fait par l'émetteur du paquet.

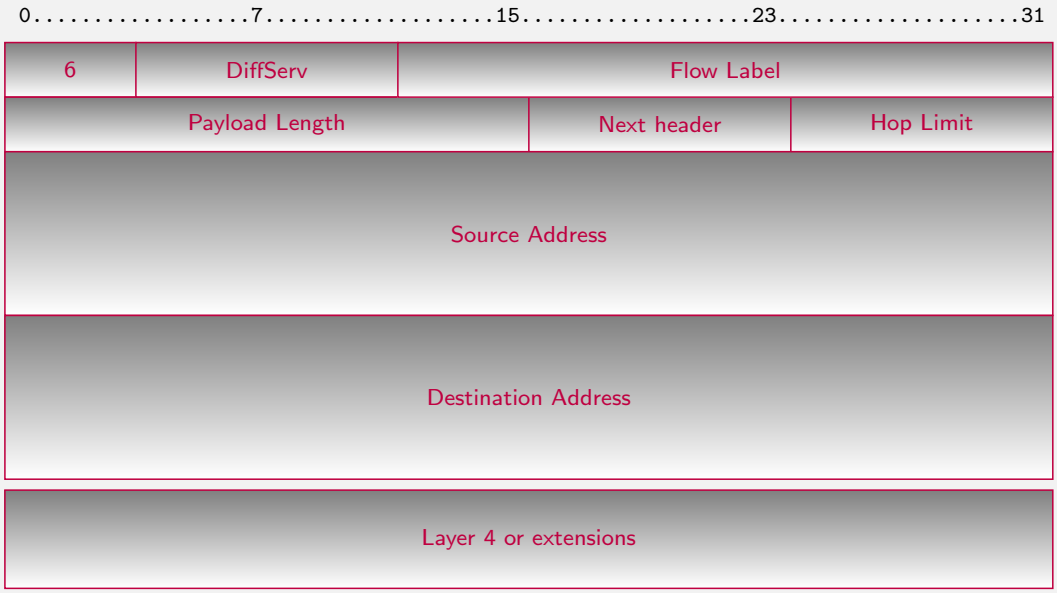
IPv6 Header



IPv6 Header



IPv6 Header



Comments I

La taille des en-têtes est fixe. Le routeur peut facilement déterminer oô commence la zone de données utiles. En IPv4 les options n'étaient pas utilisées car mal mises en œuvre dans les routeurs, ce qui fait que très peu de paquets en contenait. Pour rendre plus efficace des ajouts de traitements supplémentaires, IPv6 repose sur des extensions qui peuvent être vu comme des protocoles de niveau supérieur.

La fonction de fragmentation a été retirée des routeurs. Les champs qui s'y reportent (identification, drapeau, place du fragment) ont été supprimés. Normalement les algorithmes de découverte du PMTU(Path MTU) évitent d'avoir recours à la fragmentation. Si celle-ci s'avère nécessaire, une extension est prévue.

L'en-tête ne contient plus le champ checksum, qui devait être ajusté par chaque routeur en raison de la décrémentation du champ durée de vie. Par contre, pour éviter qu'un paquet dont le contenu est erroné – en particulier sur l'adresse de destination – ne se glisse dans une autre communication, tous les protocoles de niveau supérieur doivent mettre en ?uvre un mécanisme de checksum de bout en bout incluant un pseudo-en-tête qui prend en compte les adresses source et destination. Le checksum d'UDP, facultatif pour IPv4, devient ainsi obligatoire. Pour ICMPv6, le checksum intègre le pseudo-en-tête, alors que pour ICMPv4, il ne portait que sur le message ICMP.

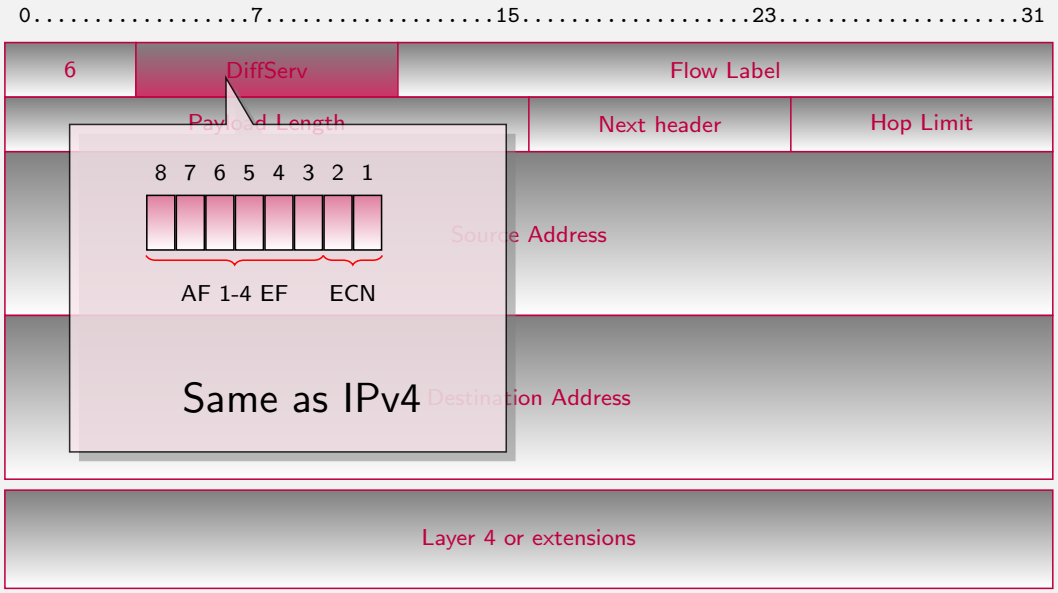
Les champs TTL ont été renommé en Hop Limit et le champ Protocol est renommé en Next Header.

Un champ Flow Label a été ajouté au paquet.

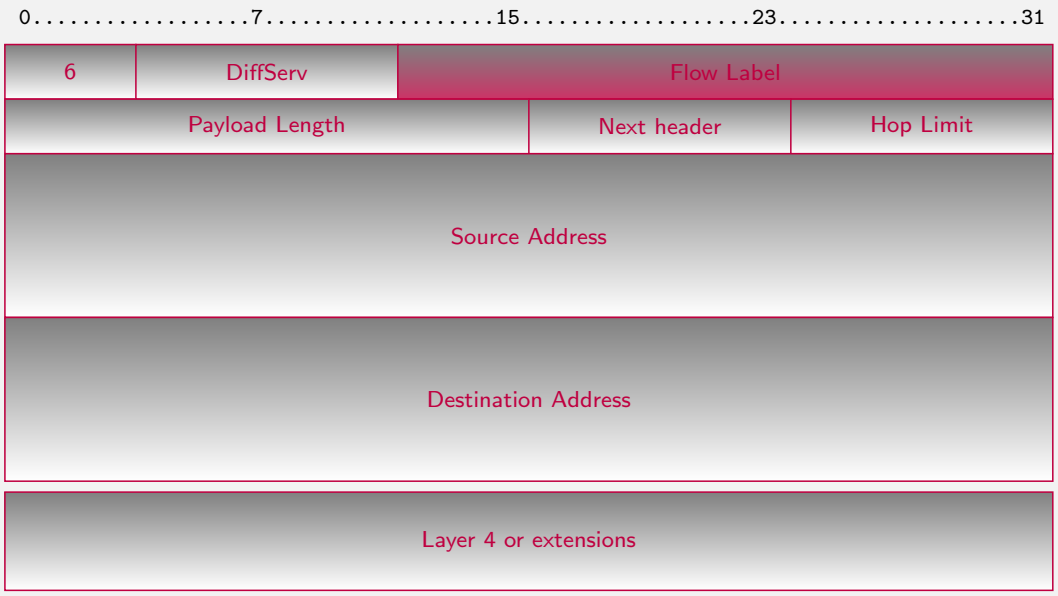
L'en-tête contient moins de champs, donc on a un traitement simplifié dans le routeur. La taille de l'en-tête IPv6

n'est que le double de l'en-tête IPv4, bien que les adresses soient quatre fois plus grande.

DiffServ & Flow Label



DiffServ & Flow Label



Flow Label (RFC 3697): Why?

- A flow is a sequence of packets that should receive specific non-default handling from the network
  - For instance : 5-tuple of the same source/destination address/port and transport protocol values
- The Flow Label field is designed to enable classification of packets belonging to a specific flow
  - Without the flow label the classifier must use transport next header value and port numbers
    - Less efficient (need to parse the option headers)
    - May be impossible (fragmentation or IPsec ESP)
- A flow is a unique identifier (for the source)
  - Flow label + source address is unique
  - Reduce processing time by 2-4 times in IPv4 and 3-6 times in standard IPv6
  - After a silence of 120 seconds, packets are not assumed to belong to the same flow

Flow Label: Why not?

- When IPv6 started to be designed:
  - RSVP was forecast to allow end-to-end micro flows reservation.
  - Short path for ATM VC
- With DiffServ, aggregation inside provider network is preferred:
  - Macro flows (e.g. from an ingress to an egress router)
  - Different sources, possibility of no common bits in the header
- Providers do not have to look at flow label to identify macro-flows
  - Only hosts have to do it, and they ´†have time†ª to look at port numbers
  - Firewalls cannot use Flow Label to take decisions

Comments I

Le champ classe de trafic est aussi appelé dans les paquets IPv4 octet DiffServ (DS), il prend la place du champ ToS, initialement défini dans la spécification d'IPv4 (cf. figure Format de l'octet classe de trafic). Le champ DS est découpé en deux parties. Le sous-champ DSCP (DiffServ Code Point) contient les valeurs des différents comportements. Les deux derniers bits du champ sont actuellement non utilisés, mais devraient servir aux routeurs pour indiquer un risque de congestion en combinaison avec l'algorithme RED (Random Early Detection).

L'Internet différencié permet aux fournisseurs d'accès de gérer différemment les congestions qui surviennent dans le réseau. Sans différenciation, les paquets ont la même probabilité de rejet. Avec la différenciation, plusieurs classes de trafic seront définies. Les paquets appartenant aux classes les plus élevées ont une probabilité de rejet plus faible. Bien entendu pour que l'introduction de telles classes de service soit efficace, il faut offrir des tarifications différentes pour chacune des classes et des mécanismes de contrôle pour vérifier qu'un utilisateur n'utilise pas que les classes les plus élevées ou qu'il dépasse son contrat.

L'intérêt principal de la différenciation de services est qu'elle ne casse pas le modèle initial de l'Internet (version 4 ou version 6). Les flux sont toujours traités en "Best Effort" même si certains sont plus "Best" que d'autres. Il n'y a aucune garantie qu'un trafic d'une classe de service haute arrive à destination, mais la probabilité est plus importante. L'autre intérêt des classes de service vient de la possibilité d'agrégation des flux. La classe d'appartenance est indiquée dans l'en-tête du paquet. Les applications peuvent marquer le paquets en fonction de paramètres locaux (trafic du directeur de la société, flux multimédia interactif,...). Le fournisseur d'accès qui récupère le trafic n'a plus à se préoccuper des applicatifs, il vérifie que le trafic d'une classe ne dépasse pas le contrat préalablement établi.

Dans le c?ur de son réseau, les routeurs prennent en compte les différentes classes. Le fournisseur d'accès devra également passer des accords avec les autres opérateurs pour pouvoir faire transiter les flux avec un traitement approprié. Cet aspect de dimensionnement de réseau et de négociation d'accords d'échange est au coeur du métier d'opérateur.

Comments II

Pour l'instant deux types de comportement sont standardisés :

- Assured Forwarding : Ce comportement définit quatre classes de services et trois priorités suivant que l'utilisateur respecte son contrat, le dépasse légèrement ou est largement en dehors. Les classes sont donc choisies par l'utilisateur et restent les mêmes tout au long du trajet dans le réseau. La priorité, par contre, peut être modifiée dans le réseau par les opérateurs en fonction du respect ou non des contrats.
- Explicit Forwarding : Ce comportement est comparable à un circuit à débit constant établi dans le réseau. Le trafic est mis en forme à l'entrée du réseau, en retardant l'émission des paquets qui sont hors contrat.

En plus de ces comportements, l'octet DS a gardé, pour des raisons de compatibilité avec les équipements existants, les valeurs du bit ToS qui étaient le plus fréquemment utilisées. En particulier, la valeur 0xE0 correspond à la classe de contrôle du réseau (Network Control). Elle est utilisée dans des mises en oeuvre d'IPv6 pour l'émission de certains paquets ICMPv6.

Ce champ contient un numéro unique choisi par la source, qui a pour but de faciliter le travail des routeurs et la mise en oeuvre des fonctions de qualité de service comme RSVP. Cet indicateur peut être considéré comme une marque à un contexte dans le routeur. Le routeur peut alors faire un traitement particulier : choix d'une route, traitement en "temps-réel" de l'information.
Avec IPv4, certains routeurs, pour optimiser le traitement, se basent sur les valeurs des champs adresses de la source et de destination, numéros de port de la source et de destination et protocole pour construire un contexte. Ce contexte sert à router plus rapidement les paquets puisqu'il évite de consulter les tables de routage pour chaque paquet. Ce contexte est détruit après une période d'inactivité.
Avec IPv6, cette technique est officialisée. Le champ identificateur de flux peut être rempli avec une valeur aléatoire qui servira à référencer le contexte. La source gardera cette valeur pour tous les paquets qu'elle émettra pour cette application et cette destination. Le traitement est optimisé puisque le routeur n'a plus à consulter cinq

Comments III

champs pour déterminer l'appartenance d'un paquet. De plus si une extension de confidentialité est utilisée, les informations concernant les numéros de port sont masquées aux routeurs intermédiaires.

Le groupe de travail MPLS (Multi Protocol Label Switching). L'identificateur de flux d'IPv6 n'est plus utilisé, mais un en-tête spécifique est introduit entre l'encapsulation de niveau 2 et celle de niveau 3. L'identificateur de flux n'a plus à être modifié en cours de transmission. Cette évolution clarifie l'utilisation du protocole RSVP (Reservation Protocol) qui peut se baser sur cette valeur, identique tout au long du chemin, pour identifier un flux.

En fait, l'utilisation de l'étiquette de flux est très floue, les micro-flux, c'est-à-dire de flux applicatifs, ne sont pas vus dans le coeur du réseau pour des raisons de scalabilité, de plus MPLS a repris la notion de routage spécifique en fonction d'une étiquette. Pour l'instant ce champ peut être vu comme réservé et son utilisation pourra être mieux spécifiée dans le futur.

Protocol

IPv6 Extensions

Extensions

- Seen as a L4 protocol
- Processed only by destination
  - Except Hop-by-Hop processed by every router
  - Equivalent of option field in IPv4
- No size limitation
- Several extensions can be linked to reach L4 protocol
- Processed only by destination
  - Destination (mobility)
  - Routing (loose source routing, mobility)
  - Fragmentation
  - Authentication (AH)
  - Security (ESP)

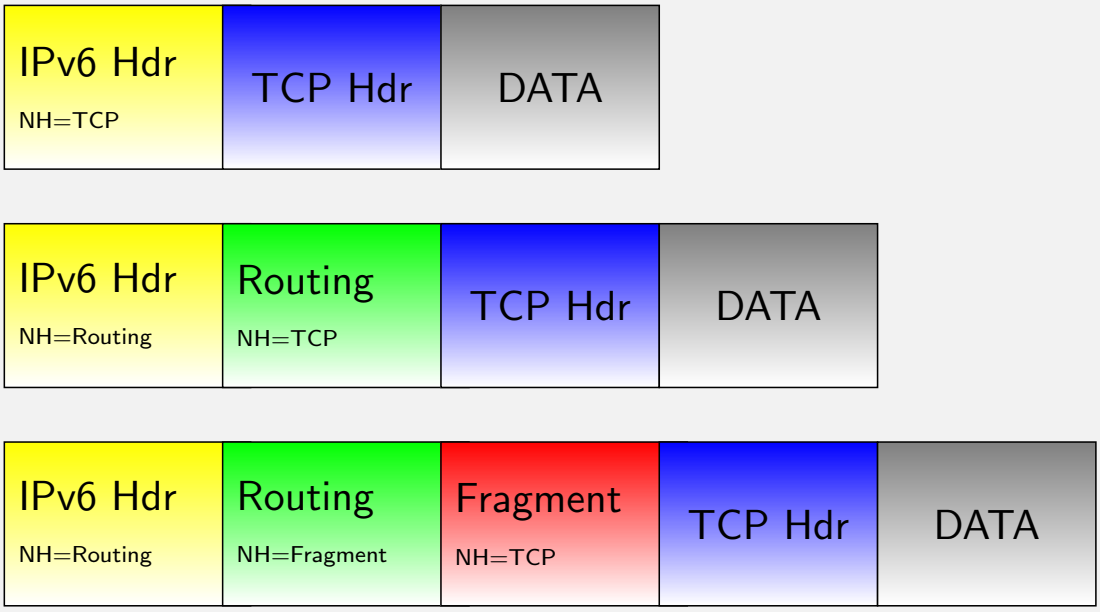
Comments I

Les extensions peuvent être vues comme un protocole 3.5 (entre la couche 3 et la couche 4). En effet, à part l'extension de proche-en-proche, qui est traitée par tous les routeurs traversés, les autres extensions ne sont traitées que par le destinataire du paquet (i.e. celui spécifié dans le champ adresse de destination du paquet IPv6).

Si d'un point de vue théorique les extensions sont supérieurs aux options d'IPv4, dans la réalité très peu sont

utilisées à grande échelle et restent du domaine de la recherche.

Extensions in packets



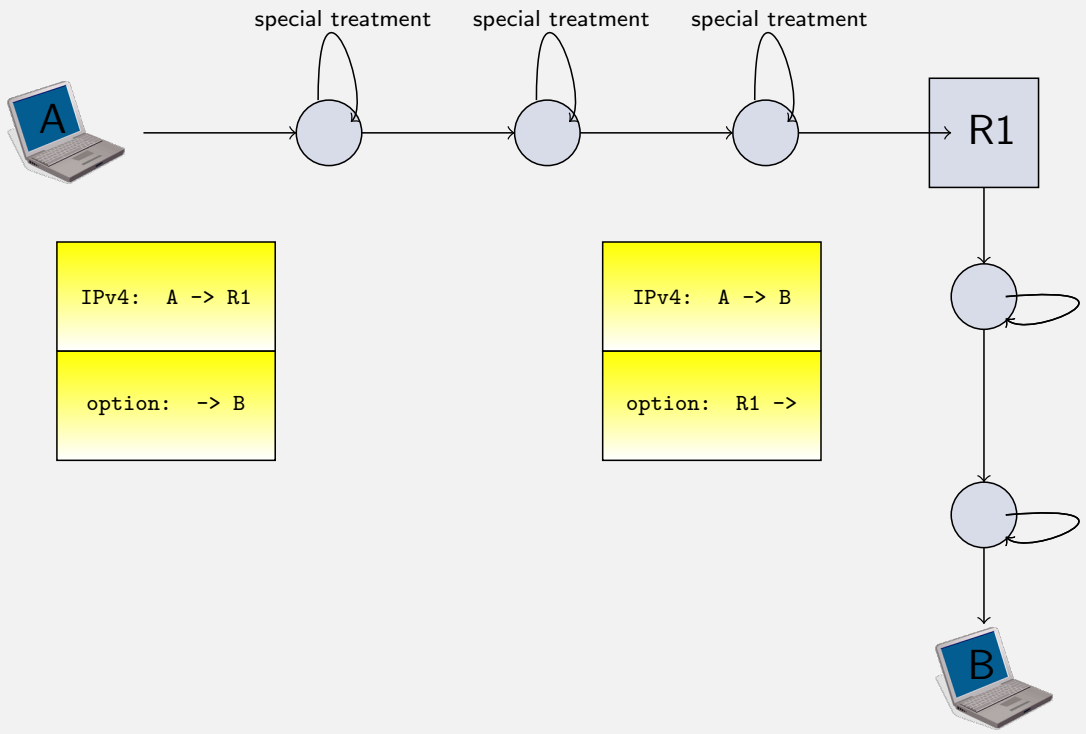
Comments I

Cette figure montre la souplesse avec laquelle plusieurs extensions peuvent être chaînées. Chaque extension contient dans son en-tête un champ en-tête suivant et longueur. Le premier paquet ne contient pas d'extension, le champ en-tête suivant pointe sur TCP. Le second paquet contient une extension de routage qui pointe sur TCP. Dans le dernier paquet, une extension de fragmentation est ajoutée après celle de routage.

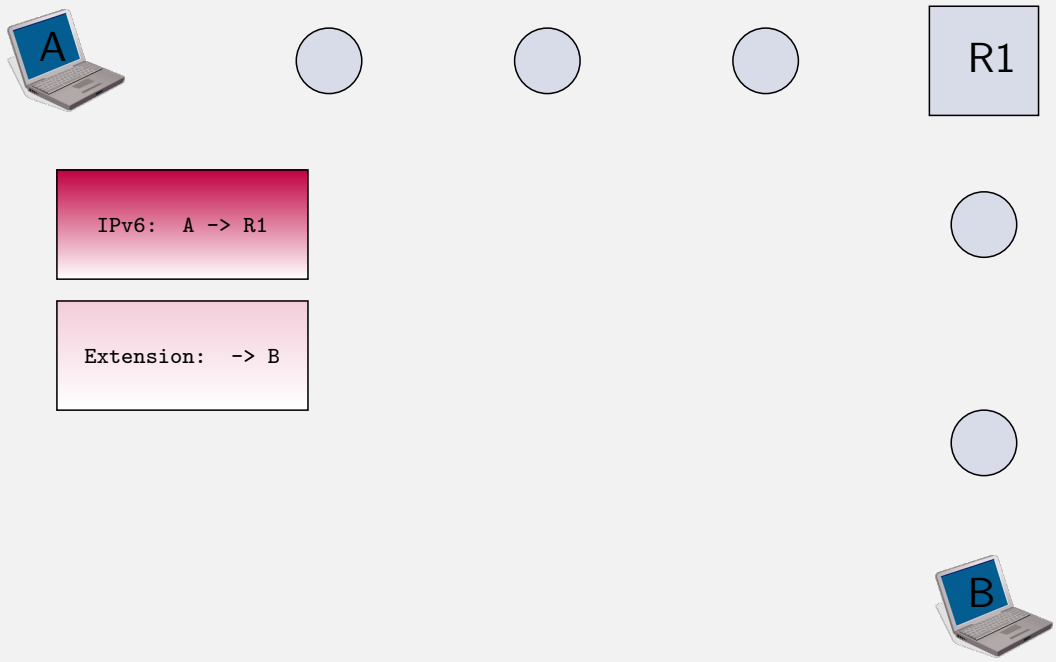
Si cet enchaînement d'extension offre beaucoup plus de souplesse que les options d'IPv4, il rend difficile la lecture des numéros de port, il faut en effet lire tout l'enchaînement d'extension pour arriver au protocole de niveau 4.

Ceci a servi de justification au l'identificateur de flux qui permettait de refléter au niveau 3 un flux particulier et évitait de dérouler l'enchaînement. Bien entendu, les pare-feux devront aux numéros de ports.

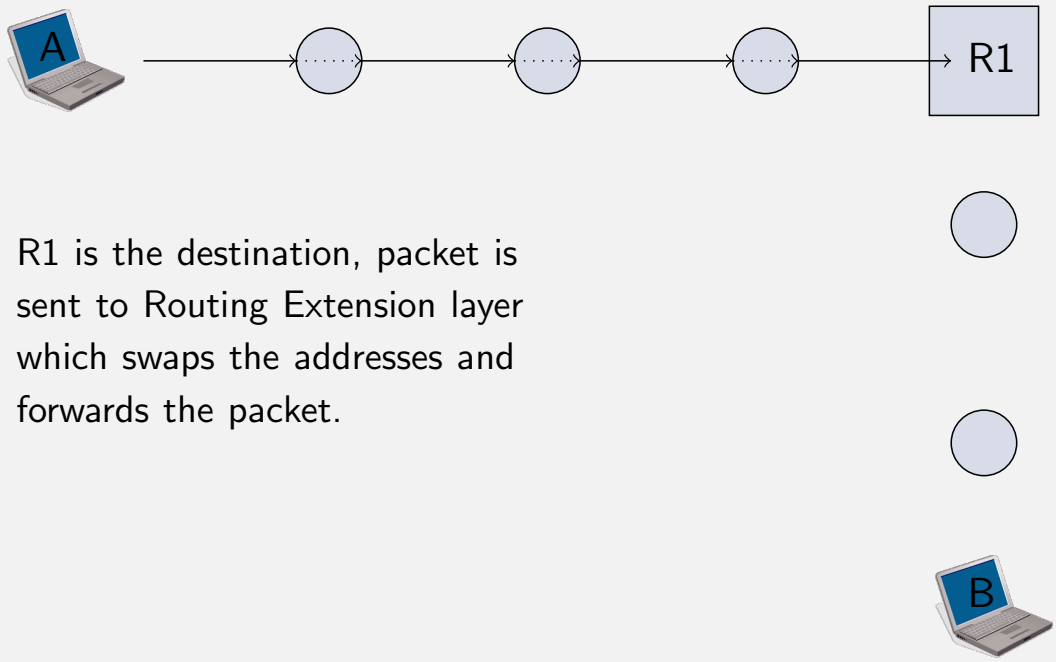
Extension Superiority



Extension Superiority

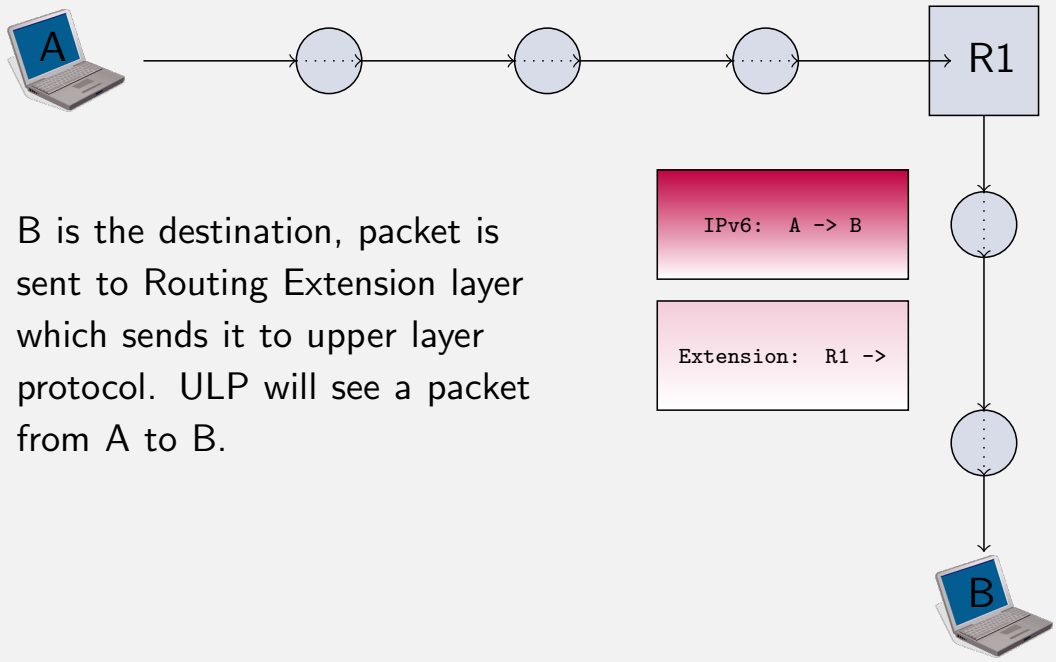


Extension Superiority



R1 is the destination, packet is sent to Routing Extension layer which swaps the addresses and forwards the packet.

Extension Superiority



B is the destination, packet is sent to Routing Extension layer which sends it to upper layer protocol. ULP will see a packet from A to B.

Comments I

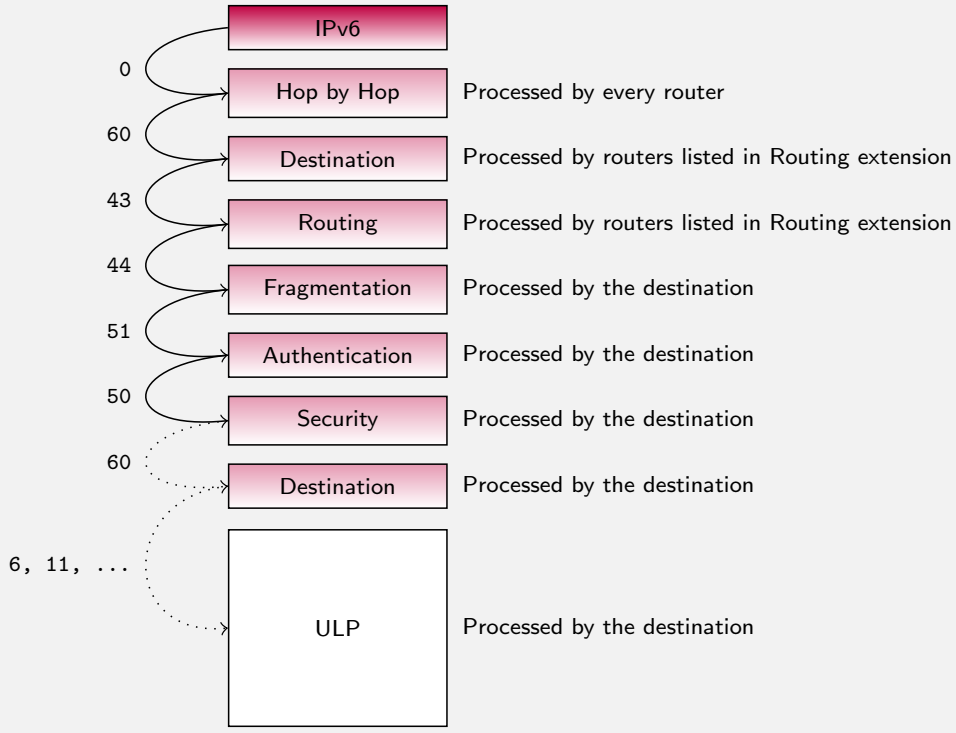
Cet exemple permet de souligner les problèmes d'utilisation des options dans IPv4, d'illustrer la notion de tunnel et le concept de transmission multicast.

La solution (cf. figure Traitement de l'option LSR en IPv4) consiste à émettre le paquet avec l'option de routage libéral par la source (loose source routing). Le paquet est destiné au routeur R1, qui permute l'adresse de destination avec celle contenue dans le champ option. Le paquet franchissant les routeurs entre A et R1 puis R1 et B sera retardé à cause de la présence du champ option. Avec IPv4, les options sont obligatoirement prises en compte par tous les routeurs intermédiaires. Ceux-ci, pour des raisons de performance, privilégient les paquets sans option. De plus, par construction, la longueur du champ option est limitée à 40 octets, ce qui limite l'emploi simultané de plusieurs options.

Avec IPv6 la philosophie est différente comme le montre la figure "Traitement avec l'extension de routage IPv6". Un paquet normal à destination de R1 est envoyé dans le réseau et est traité normalement par les routeurs intermédiaires. R1 reconnait son adresse et le passe à la couche supérieur qui traite l'extension de routage. Cette couche inverse les adresses et réémet le paquet vers la nouvelle destination.

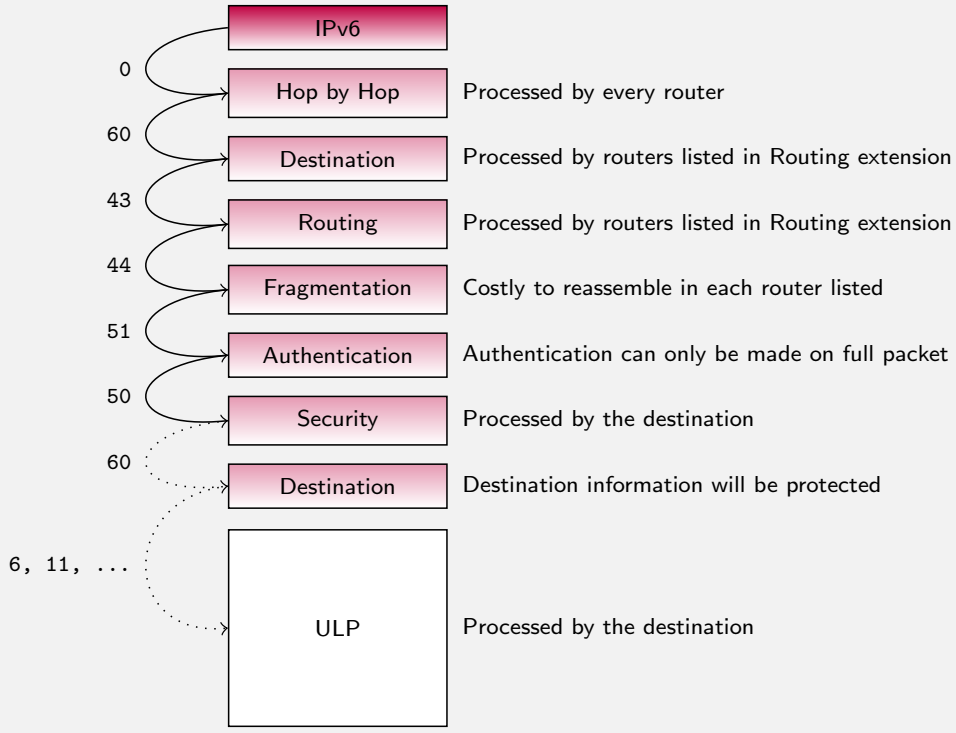
Il faut noter que cet exemple est purement théorique, car le

Extension Order is Important



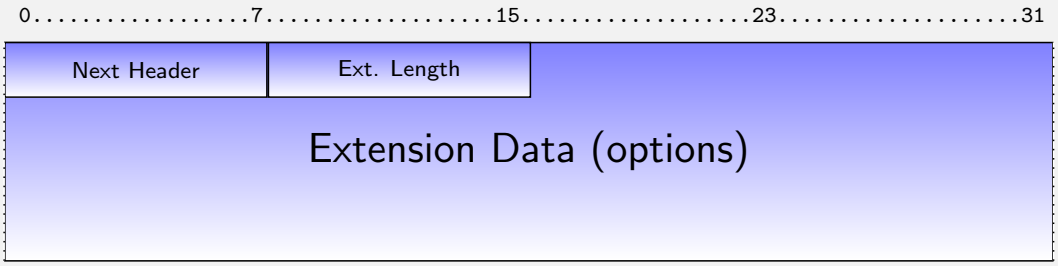
TELECOM Bretagne

Extension Order is Important



TELECOM Bretagne

Extensions Generic Format



- Next Header: Save values as in IPv6 packets
- Length: numbers 64-bit long words for variable length extensions (0 for fixed length fragmentation extension)
- Data: options (Hop by hop, Destination) or specific format

Comments I

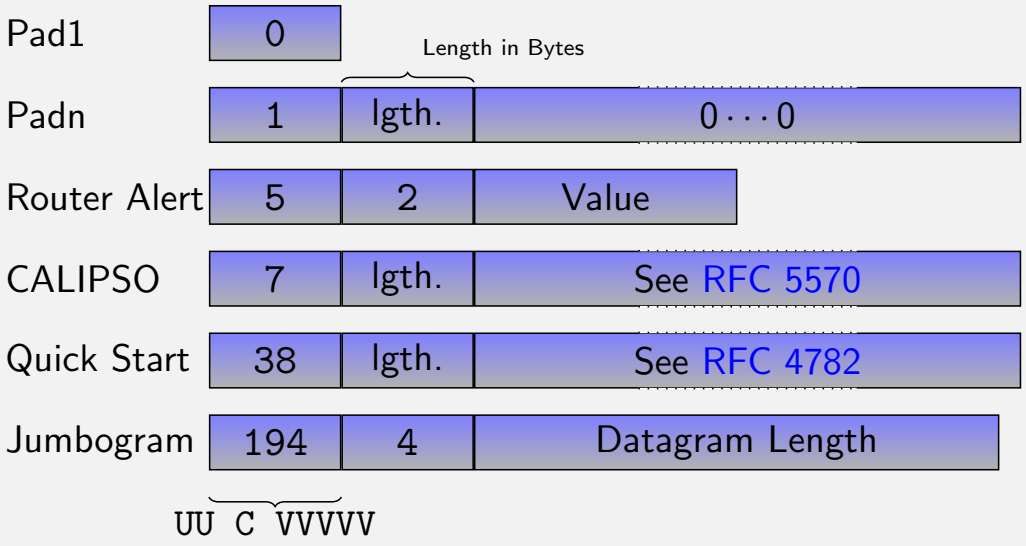
Toutes les extensions sont construites suivant le même modèle. L'extension commence par un champ Next Hop qui indique quel sera la nature de l'encapsulation suivante, comme pour l'en-tête IPv6.

Le deuxième champ contient la longueur de l'extension, généralement en mot de 64 bits. Pour l'extension de fragmentation qui a une longueur fixe, la valeur est 0.

La partie données peut être structurée en options (comme les extensions de proche-en-proche ou de destination) ou avoir un format spécifique.

Hop by Hop (NH=0)

- Always first position
- Composed of options:

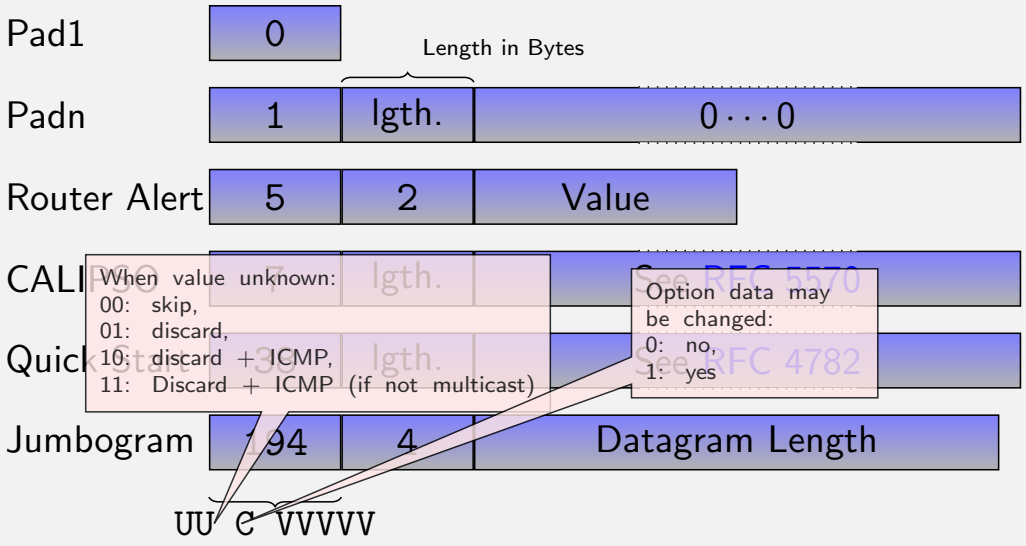


TELECOM Bretagne

Hop by Hop (NH=0)

Protocol ▶ IPv6 Extensions

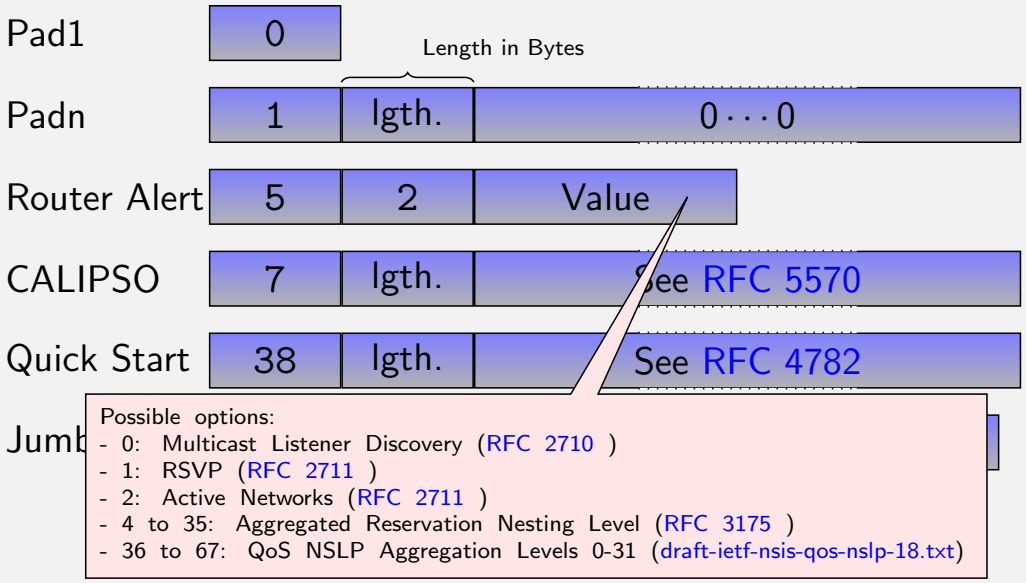
- Always first position
- Composed of options:



TELECOM Bretagne

Hop by Hop (NH=0)

- Always first position
- Composed of options:



Comments I

Cette extension (en anglais : hop-by-hop) se situe toujours en première position et est traitée par tous les routeurs que le paquet traverse. Le type associé (contenu dans le champ d'en-tête en-tête suivant de l'en-tête précédent) est 0 et le champ longueur de l'extension contient le nombre de mots de 64 bits moins 1.
L'extension est composée d'options. Pour l'instant, seules quatre options, dont deux de bourrage, sont définies (cf. Format des options IPv6). Chaque option est une suite d'octets. Le premier octet est un type, le deuxième (sauf pour l'option 0) contient la longueur de l'option moins 2. Les deux premiers bits de poids fort du type définissent le comportement du routeur quand il rencontre une option inconnue :

- 00 : le routeur ignore l'option ;
- 01 : le routeur rejette le paquet ;
- 10 : le routeur rejette le paquet et retourne un message ICMPv6 d'inaccessibilité ;
- 11 : le routeur rejette le paquet et retourne un message ICMPv6 d'inaccessibilité si l'adresse de destination n'est pas multicast.

Le bit suivant du type indique que le routeur peut modifier le contenu du champ option (valeur à 1) ou non (valeur à 0).
Les quatre options de proche-en-proche sont :

- Pad1 (type 0). Cette option est utilisée pour introduire un octet de bourrage.
- Padn (type 1). Cette option est utilisée pour introduire plus de 2 octets de bourrage. Le champ longueur indique le nombre d'octets qui suivent.

Comments II

Les options de bourrage peuvent sembler inutiles avec IPv6 puisqu'un champ longueur pourrait en donner la longueur exacte. En fait les options de bourrage servent à optimiser le traitement des paquets en alignant les champs sur des mots de 32, voire 64 bits ; le RFC 2460 discute en annexe de la manière d'optimiser le traitement tout en minimisant la place prise par les options.

L'option Router Alert (RFC 2711) demande à un routeur d'examiner le contenu des données qu'il relaie (Router Alert existe également en IPv4, RFC 2113). En principe, le processus de relayage (recopier le paquet sur une interface de sortie en fonction de l'adresse destination et des tables de routage) doit être le plus rapide possible. Mais pour des protocoles comme la gestion des groupes de multicast avec MLD (Multicast Listener Discovery) ou la signalisation des flux avec RSVP, tous les routeurs intermédiaires doivent tenir compte des données. L'émetteur envoie les données à la destination, mais s'il précise l'option Router Alert, les routeurs intermédiaires vont analyser les données, voire modifier leur contenu avant de relayer le paquet. Ce mécanisme est efficace puisque les routeurs n'ont pas à analyser le contenu de tous les paquets d'un flux. Le type de l'option vaut 5. Il commence par la séquence binaire 00, puisqu'un routeur qui ne connaît pas cette option doit relayer le paquet sans le modifier. Le champ valeur de l'option contient :

- 0 : pour les messages du protocole MLD de gestion des groupes multicast ;
- 1 : pour les messages RSVP ;
- 2 : pour les réseaux actifs ;
- 4 à 35 : niveau d'imbrication de réservation pour RSVP
- 36 à 67 : niveau d'imbrication de réservation pour NSIS

Comments III

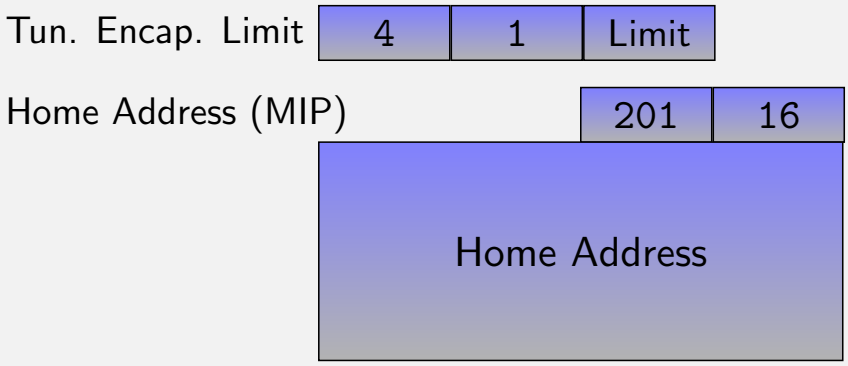
L'option CALIPSO permet de donner un degré de confidentialité au paquet transporté. Elle est décrite dans le RFC 5570 , mais doit être limité a un intranet, car l'utilisation de l'extension Hop-By-Hop nuit a l'efficacité du relayage des paquets.

L'option Demarrage Rapide (Quick Start) de manière expérimentale par le RFC 4782 . Elle permet aux applications de collaborer avec les routeurs pour déterminer le débit auquel l'application peut commencer à émettre.

Jumbogramme (type 194 ou 0xc2, RFC 2675). Cette option est utilisée quand le champ longueur des données du paquet IPv6 n'est pas suffisant pour coder la taille du paquet. Cette option est essentiellement prévue pour la transmission à grand débit entre deux équipements. Si l'option jumbogramme est utilisée, le champ longueur des données utiles dans l'en-tête IPv6 vaut 0. Noter que le type commence par la séquence binaire 11, ce qui permet au routeur ne traitant pas les jumbogrammes d'en informer la source. Celle-ci pourra réémettre l'information sans utiliser cette option.

les autres valeurs sont réservées.

Destination (NH=60)



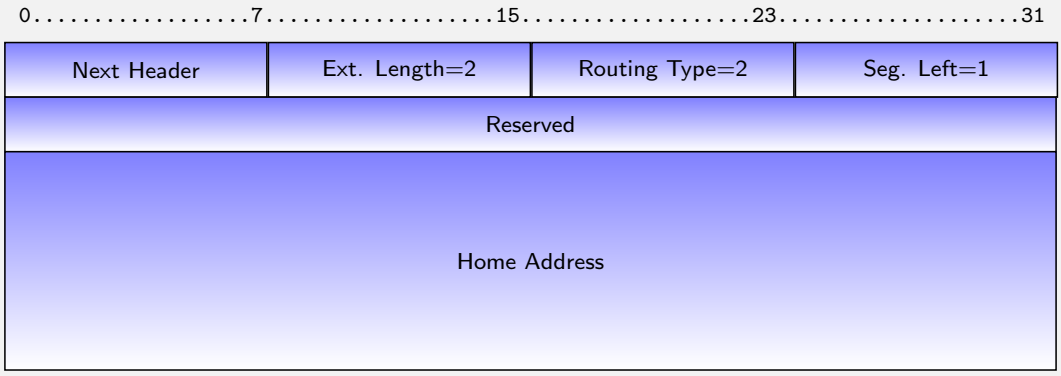
- Tunnel Encapsultation Limit (RFC 2473): the maximum number of nested encapsulations of a packet. When it reaches 0, the packet is discard and an ICMPv6 message is sent.
- Home Address (RFC 3775): Contains the Home Address of the sender (IPv6 header contains the Care-of Address).

Comments I

Cette extension, dont le format est identique à l'extension de proche-en-proche (contient des options qui sont traitées par l'équipement destinataire. Le RFC 2460 définissant IPv6 ne définit que les options de bourrage Pad1 et Padn. Les autres options sont définies dans d'autres RFC ou encore expérimentales. Les valeurs:

- 4 : "Tunnel Encapsulation Limit" [RFC 2473]: Contient le nombre de fois maximum qu'un paquet peut être encapsulé dans les tunnels. La valeur est décrémentée a chaque fois qu'un nouveau tunnel est ajouté. Si la valeur atteint 0, le paquet est détruit et un message ICMPv6 est émis.
- 201 (0xC9): contient l'adresse sur le réseau mère ("Home Address") [RFC 3775] utilisée pour l'optimisation de la mobilité. L'en-tête IPv6 contient dans le champ adresse de la source, l'adresse sur le réseau visité ("Care-of Address"). Cette option est utilisée pour éviter qu'un opérateur ne rejette un paquet dont l'adresse de la source ne correspond pas à la plage de valeur qu'il a attribué au site. Le récepteur remplace l'adresse de la source de l'en-tête IPv6 par celle contenue dans cette option.

Routing (NH=43)



Comments I

Dans IPv4, le routage peut être strict (le routeur suivant présent dans la liste doit être un voisin directement accessible) ou libéral (loose) (un routeur peut utiliser les tables de routage pour joindre le routeur suivant servant de relais). Dans IPv6, seul la spécification d'un changement d'adresse au dernier lien est spécfié. En effet, le routage strict était initialement mis en place surtout pour des raisons de sécurité. La source devait être absolument s˚re du chemin pris par les paquets. Cette utilisation a maintenant disparu du réseau. Le routage par la source libéral pouvait conduire à une duplication de paquets dans le réseau et a été supprimé dans les dernière spécifications. Cette amplification du trafic permettant de réaliser des attaques par déni de service. Ainsi si dans la liste des routeurs a traverser, on met une liste R1, R2, R1, R2, ... le paquet fera du ping pong entre ces deux routeurs, comme l'explique le RFC 5095.

Le seul format de routage existant est le type 2 (appelé RH2, pour Routing Header type 2) comme le montre la figure "Format de l'extension routage". Il sert pour la mobilité. Son rôle est inverse de l'option Home Address de l'extension Destination. Quand un paquet est émis vers un noeud mobile, l'adresse dans le paquet IPv6 contient l'adresse du réseau visité, et l'adresse permanente est stockée dans l'extension RH2. Le noeud mobile reÁoit le paquet IPv6, traite l'extension et par conséquent remplace l'adresse de destination par la Home Address. Le paquet est ensuite transmis au niveau 4 qui n'a pas la notion des changements d'adresses du n?ud.

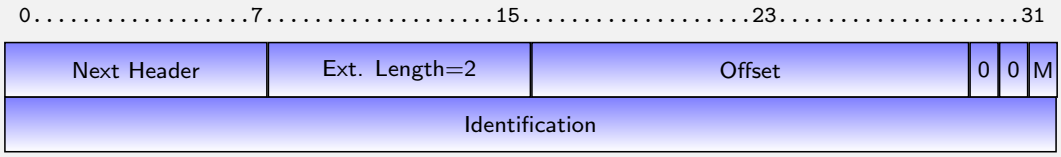
Le slide donne le format de l'extension de routage par la source :
- Le champ longueur de l'en-tête indique le nombre de mots de 64 bits qui composent l'extension. Pour l'extension de type 0, cela correspond au nombre d'adresses présentes dans la liste, multiplié par 2. Dans l'en-tête du type 2, il est fixé à 2 car une seule adresse est possible.
- Le champ type indique la nature du routage.

Comments II

- Le routage par la source, de type 0 est spécifié a été déprécié (cf RFC 5095) pour les possibilité amplification du trafic expliqué précédemment. Dans la description initiale, le champ longueur pouvait contenir un nombre quelconque d'adresses de routeurs intermédiaire. Le draft-manral-ipv6-rh4-00.txt aujourd'hui expiré proposait de borner le nombre d'adresses à 4.
- Le type 1 correspond à un adressage expérimental (Nimrod) testé au début d'IPv6, il est également abandonné.
- Le type 2 correspond à la mobilité, décrit ci dessus.

- Le nombre de segments restant est décrémenté après la traversée d'un routeur. Il indique le nombre d'équipements qui doivent encore être traversés. Il permet de trouver l'adresse qui devra être substituée. Pour RH2, il est forcement à 1.
Les 32 bits suivants sont inutilisés pour préserver l'alignement sur 64 bits du premier mot et avoir ainsi la suite des adresses IPv6 sur ces mêmes frontières.

Fragmentation (NH=44)



- Compared to IPv4, it is equivalent to DF=1
- A Router never fragments packets but sends an ICMPv6 message ("Packet Too Big") with the expected size
- The Sender either uses the fragmentation extension or adapts TCP segments

Comments I

La fragmentation telle qu'elle est pratiquée dans IPv4 n'est pas très performante. Initialement, elle servait à rendre transparente les limitations physiques des supports de transmission. Dans IPv4 quand un routeur ne peut pas transmettre un paquet à cause de sa trop grande taille et si le bit DF (don't fragment) est à 0, il découpe l'information à transmettre en fragments. Or le réseau IP étant un réseau à datagramme, il n'y a pas de possibilité de contrôler les fragments. Deux fragments successifs peuvent prendre deux chemins différents et par conséquent seul le destinataire peut effectuer le réassemblage. En conséquence, après la traversée d'un lien impliquant une fragmentation, le reste du réseau ne voit passer que des paquets de taille réduite.
Il est plus intéressant d'adapter la taille des paquets à l'émission. Ceci est fait en utilisant les techniques de découverte du MTU (voir Mécanisme de découverte du PMTU (RFC 1981)). En pratique une taille de paquets de 1 500 octets est presque universelle.
Il existe pourtant des cas oô la fragmentation est nécessaire. Ainsi une application telle que NFS sur UDP suppose que la fragmentation existe et produit des messages de grande taille. Comme on ne veut pas modifier ces applications, la couche réseau d'IPv6 doit aussi être capable de gérer la fragmentation. Pour réduire le travail des routeurs intermédiaires, la fragmentation se fera chez l'émetteur et le réassemblage chez le récepteur.
Le format de l'extension de fragmentation est donné dans le slide précédent. La signification des champs est identique à celle d'IPv4 :

- Le champ place du fragment indique lors du réassemblage oô les données doivent être insérées. Ceci permet de parer les problèmes dus au déséquencement dans les réseaux orientés datagrammes. Comme ce champ est sur 13 bits, la taille de tous les segments, sauf du dernier, doit être multiple de 8 octets.
- Le bit M s'il vaut 1 indique qu'il y aura d'autres fragments émis.
- Le champ identification permet de repérer les fragments appartenant à un même paquet initial. Il est différent pour chaque paquet et recopié dans ses fragments.

Comments II

- Le bit DF (don't fragment) n'est plus nécessaire puisque, si un paquet est trop grand, il y aura rejet du paquet par le routeur.

Dans IPv4, la valeur d'une option était codée de manière à indiquer au routeur effectuant la fragmentation si elle devait être copiée dans les fragments. Dans IPv6, l'en-tête et les extensions qui concernent les routeurs intermédiaires (pour l'instant proche-en-proche, routage par la source) sont recopiées dans chaque fragment.

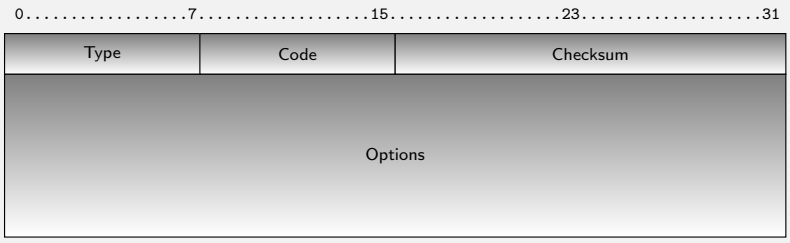
Protocol

ICMPv6

ICMPv6

- ICMPv6 is different from ICMP for IPv4 (RFC 4443)
  - IPv6 (or extension): 58
- Features are extended and better organized
- Never filter ICMPv6 messages blindly, be careful to what you do (see RFC 4890)

Format :



Precision

type code nature of the message ICMPv6
code specifies the cause of the message ICMPv6
mandatory *checksum* used to verify the integrity of ICMP packet

ICMPv6 : Two Functions

- Error occurs during forwarding ($value < 128$)

1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem

- Management Applications ($value > 128$)

128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertissement
135	Neighbor Solicitation
136	Neighbor Advertissement
137	Redirect

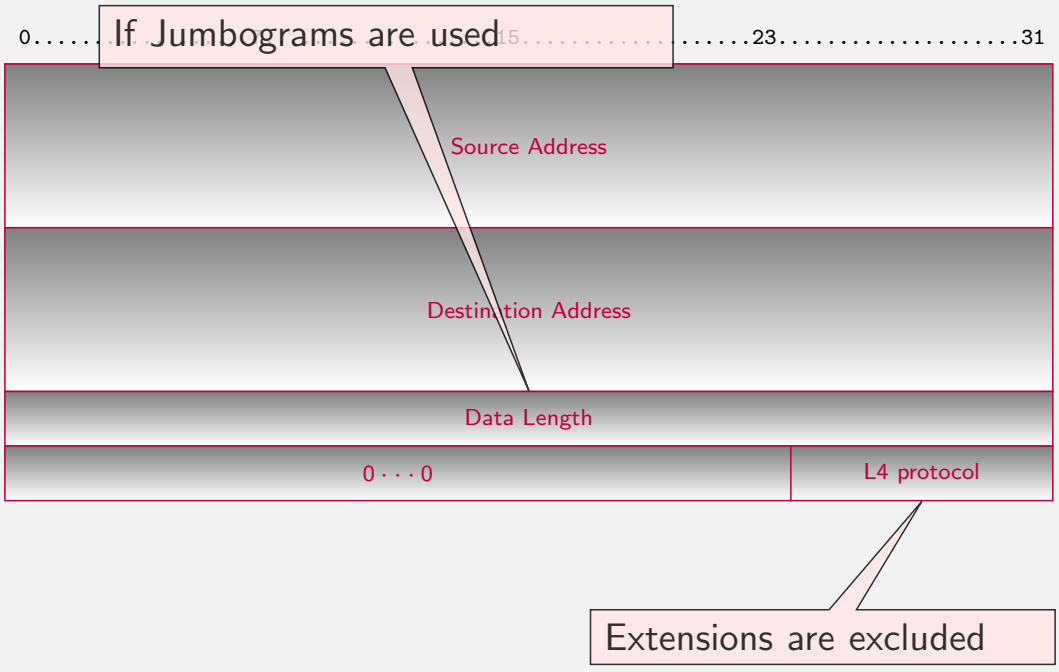
Comments I

Le protocole de contrôle d'IP a été revu. Dans IPv4, ICMP (Internet Message Control Protocol) sert à la détection d'erreurs (par exemple : équipement inaccessible, durée de vie expirée,...), au test (par exemple ping), à la configuration automatique des équipements (redirection ICMP, découverte des routeurs). Ces trois fonctions ont été mieux définies dans IPv6. De plus ICMPv6 (RFC 2463) intègre les fonctions de gestion des groupes de multicast (MLD : Multicast Listener Discovery) qui sont effectuées par le protocole IGMP (Internet Group Message Protocol) dans IPv4. ICMPv6 reprend aussi les fonctions du protocole ARP utilisé par IPv4.
Le protocole se voit attribuer le numéro 58. Le format générique des paquets ICMPv6 est donné figure Format générique d'un message ICMP :
Le champ type code la nature du message ICMPv6. Contrairement à IPv4 oô la numérotation ne suivait aucune logique, les valeurs inférieures à 127 sont réservées aux messages d'erreur. Les autres valeurs réservées aux messages d'information, parmi lesquels se trouvent ceux utilisés par le protocole découverte des voisins (neighbor discovery) pour la configuration automatique des équipements. Le champ code précise la cause du message ICMPv6. Le champ checksum permet de vérifier l'intégrité du paquet ICMP. Ce champ est calculé avec le pseudo-en-tête décrit au chapitre Checksum au niveau transport. Les messages ICMPv6 de compte rendu d'erreur contiennent dans la partie données le paquet IPv6 ayant provoqué l'erreur. Pour éviter des problèmes de fragmentation puisqu'il est difficilement envisageable de mettre en ?uvre la découverte du MTU, la longueur du message ICMPv6 est limitée à 1 280 octets et par conséquent le contenu du paquet IPv6 peut être tronqué.

Contrairement à une pratique couramment répandue en IPv4, il ne faut jamais filtrer les messages ICMPv6 (en particulier Paquet trop grand) car cela peut avoir des conséquences néfastes sur le bon fonctionnement du réseau.

Impact on Layers 4

Pseudo Header



Comments I

Parmi les différences existant entre les datagrammes IPv4 et IPv6, il y a la disparition du checksum dans les en-têtes IP. Cette somme de contrôle était utilisée pour vérifier la validité de l'en-tête du paquet traité. En IPv4, il est nécessaire de la vérifier et de l'ajuster lors de chaque retransmission par un routeur, ce qui entraîne une augmentation du temps de traitement du paquet.

Cette somme ne vérifie que l'en-tête IPv4, pas le reste du paquet. Aujourd'hui les supports physiques sont de meilleure qualité et savent détecter les erreurs (par exemple, Ethernet a toujours calculé sa propre somme de contrôle ; PPP, qui a presque partout remplacé SLIP, possède un CRC). L'intérêt de la somme de contrôle a diminué et ce champ a été supprimé de l'en-tête IPv6.

Le checksum sur l'en-tête IPv6 n'existant plus, il faut quand même se prémunir des erreurs de transmission. En particulier, une erreur sur l'adresse de destination va faire router un paquet dans une mauvaise direction. Le destinataire doit donc vérifier que les informations d'en-tête IP sont incorrectes pour éliminer ces paquets. Dans les mises en oeuvre des piles de protocoles Internet, les entités de niveau transport remplissent certains champs du niveau réseau. Il a donc été décidé que tous les protocoles au-dessus d'IPv6 devaient utiliser une somme de contrôle intégrant à la fois les données et les informations de l'en-tête IPv6. La notion de pseudo-en-tête dérive de cette conception. Pour un protocole comme TCP qui possède une somme de contrôle, cela signifie modifier le calcul de cette somme. Pour un protocole comme UDP qui possède une somme de contrôle facultative, cela signifie modifier le calcul de cette somme et le rendre obligatoire.

IPv6 a unifié la méthode de calcul des différentes sommes de contrôle. Celle-ci est calculée sur l'ensemble formé de la concaténation d'un pseudo-en-tête et du paquet du protocole concerné. L'algorithme de calcul du checksum est celui utilisé en IPv4. Il est très simple à mettre en ?uvre et ne demande pas d'opérations compliquées. Il s'agit de faire la somme en complément à 1 des mots de 16 bits du pseudo-en-tête, de l'en-tête du protocole de transport, et des données, puis de prendre le complément à 1 du résultat.

Comments II

Il faut noter que les informations contenues dans le pseudo-en-tête ne seront pas émises telles quelles sur le réseau.

Le champ "en-tête suivant" du pseudo-en-tête ne reflète pas celui qui sera émis dans les paquets puisque les extensions ne sont pas prises en compte dans le calcul du checksum. Ainsi, si l'extension de routage est mise en ?uvre, l'adresse de la destination est celle du dernier équipement. De même le champ longueur est sur 32 bits pour contenir la valeur de l'option jumbogramme, si celle-ci est présente.

Layer 4 protocols

IPv6 is almost transparent for Layer 4 protocol, except:

- Jumbogram impact:
  - UDP: if Jumbogram are used and $length > 65535 \Rightarrow$ UDP.length $= 0$ and use Jumbogram length
  - TCP: Use PMTU if $Length > 65535$
- UDP-Light: For multimedia flow a bit error is less important than a packet loss. UDP-light is used to not include UDP payload in L4 Checksum.
- SCTP: during session initialisation, IPv4 and IPv6 addresses are exchanged.

Comments I

Les modifications apportées aux protocoles de niveau 4 UDP et TCP sont minimes. L'un des pré-requis à la mise en ?uvre d'IPv6 était de laisser en l'état aussi bien TCP (Transmission Control Protocol) qu'UDP (User Datagram Protocol). Ces protocoles de transport sont utilisés par la très grande majorité des applications réseau et l'absence de modification facilitera grandement le passage de IPv4 à IPv6.

La principale modification à ces protocoles concerne le checksum. Comme il a été précisé Checksum au niveau transport, il a été adapté au format de paquet IPv6 et englobe le pseudo-en-tête. De plus, pour UDP, le checksum qui était facultatif en IPv4, devient obligatoire.
Un autre changement au niveau des protocoles de niveau 4 concerne la prise en compte de l'option jumbogramme de l'extension proche-en-proche. Le RFC 2675 définit le comportement de UDP et de TCP quand les jumbogrammes sont utilisés. En effet, les en-têtes de ces messages contiennent eux aussi un champ longueur codé sur 16 bits et par conséquent insuffisant pour coder la longueur du jumbogramme :

Pour le protocole UDP, si la longueur des données excède 65 535 octets, le champ longueur est mis à 0. Le récepteur détermine la longueur des données par la connaissance de la taille dans l'option jumbogramme.

Le protocole TCP pose plus de problèmes. En effet, bien que les messages TCP ne contiennent pas de champ longueur, plusieurs compteurs sont codés sur 16 bits.

- Le champ longueur de la fenêtre de réception ne pose pas de problème depuis que le RFC 1323 a défini l'option TCP window scale qui donne le facteur multiplicatif qui doit être appliqué à ce champ.
- ¿ l'ouverture de connexion, la taille maximale des segments (MSS) est négociée. Le RFC 2675 précise que si cette taille doit être supérieure à 65 535, la valeur 65 535 est envoyée et le récepteur prend en compte la longueur déterminée par l'algorithme de découverte du MTU.

Comments II

- Pour l'envoi de données urgentes avec TCP, on utilise un bit spécifique de l'en-tête (bit URG) ainsi que le champ "pointeur urgent". Ce dernier sert à référencer la fin des données à traiter de manière particulière. Trois cas peuvent se présenter :
  - Le premier, qui est identique à IPv4, est celui oô le pointeur indique une position de moins de 65 535.
  - Le second se produit lorsque le déplacement est supérieur à 65 535 et supérieur ou égal à la taille des données TCP envoyées. Cette fois-ci, on place la valeur 65 535 dans le champ "pointeur urgent" et on continue le traitement normal des paquets TCP.
  - Le dernier cas intervient quand le pointeur indique un déplacement de plus de 65 535 qui est inférieur à la taille des données TCP. Un premier paquet est alors envoyé, dans lequel on met la valeur 65 535 dans le champ "pointeur urgent". L'important est de choisir une taille de paquet de manière à ce que le déplacement dans le second paquet, pour indiquer la fin des données urgentes, soit inférieur à 65 535.
- Il existe d'autres propositions pour faire évoluer TCP. Il faut remarquer que le travail n'est pas de même ampleur que pour IP. En effet, TCP est un protocole de bout-en-bout, la transition vers une nouvelle génération du protocole peut se faire par négociation entre les deux extrémités. Pour IP, tous les routeurs intermédiaires doivent prendre en compte les modifications.

Comments III

UDP-lite permet de remonter aux couches supérieures des données erronées pendant leur transport. Si dans un environnement informatique, une erreur peut avoir des conséquences relativement grave quant à l'intégrité des données et il est normal de rejeter ces paquets, or, la plupart des décodeurs de flux multimédias sont capables de supporter un certains nombre d'erreurs binaires dans un flux de données. Pour améliorer la qualité perÁue par l'utilisateur, il est donc préférable d'accepter des paquets erronés plutôt que de rejeter un bloc complet d'information.

En IPv4, l'utilisation du checksum UDP étant optionnelle (la valeur 0 indique que le checksum n'est pas calculé), UDP peut être utilisé pour transporter des flux multimédia. Avec IPv6, l'utilisation du checksum a été rendue obligatoire puisque le niveau 3 n'en possède pas. Pour éviter qu'un paquet comportant des erreurs ne puisse pas être remonté aux couche supérieures, le protocole UDP-lite a été défini RFC 3828. Les modifications sont minimes par rapport à UDP. Le format de la trame reste le même, seule la sémantique du champ longueur est changée. Avec UDP, ce champ est inutile puisqu'il est facilement déduit du champ longueur de l'en-tête IP. UDP-lite le transforme en champ couverture du checksum. Si la longueur est 0, UDP-lite considère que tout le checksum couvre tout le paquet. La valeur 8 indique que seul l'en-tête UDP est protégé par le checksum (ainsi qu'une partie de l'en-tête IP gr,ce au pseudo-header). Les valeurs comprises entre 1 et 7 sont interdites car le checksum UDP-lite doit toujours couvrir l'en-tête. Une valeur supérieure à 8 indique qu'une partie des données sont protégées. Si la couverture est égale à la longueur du message on se retrouve dans un cas compatible avec UDP.

Le protocole SCTP (Stream Control Transmission Protocol) RFC 2960 est fortement lié au protocole IPv6. SCTP est un protocole de niveau 4 initialement conÁu pour transporter des informations de signalisation. La fiabilité est donc un prérequis important et la gestion de la multi-domiciliation est prise en compte. L'idée est de permettre aux deux équipements terminaux d'échanger à l'initialisation de la connexion (appelée dans le standard association), l'ensemble de leurs adresses IPv4 et IPv6. Chaque équipement choisi une adresse privilégiée pour émettre les

données vers l'autre extrémité et surveille périodiquement l'accessibilité des autres adresses. Si l'équipement n'est plus accessible par l'adresse principale, une adresse secondaire sera choisie.

SCTP permet une transition douce d'IPv4 vers IPv6 puisque l'application n'a plus à se préoccuper de la gestion des adresses. Si les deux entités possèdent une adresse IPv6, celle-ci sera privilégiée. De plus, SCTP peut servir de brique de base à la gestion de la multi-domiciliation IPv6. En effet, avec TCP une connexion est identifiée par ses adresses. Si une adresse n'est plus accessible, le fait d'en changer peut conduire à la coupure de la connexion. Il faut avoir recours à des superfuges, comme la mobilité IP pour maintenir la connexion. SCTP brise ce lien entre la localisation de l'équipement et l'identification des associations.

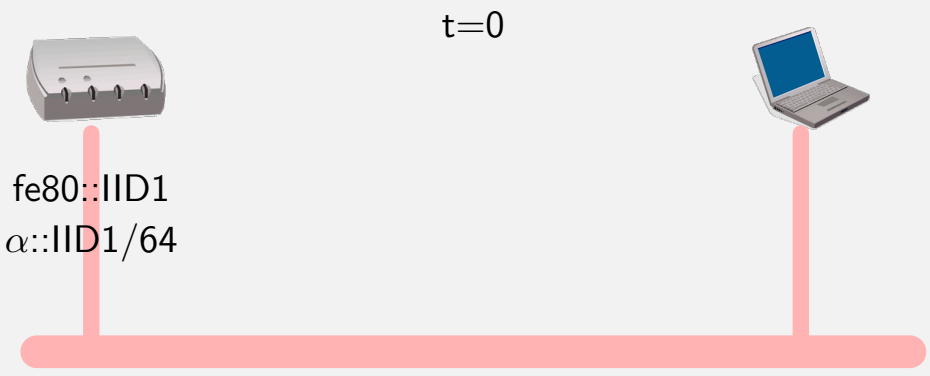
Associated Protocols & Mechanisms

Neighbor Discovery

Neighbor Discovery (RFC 4861)

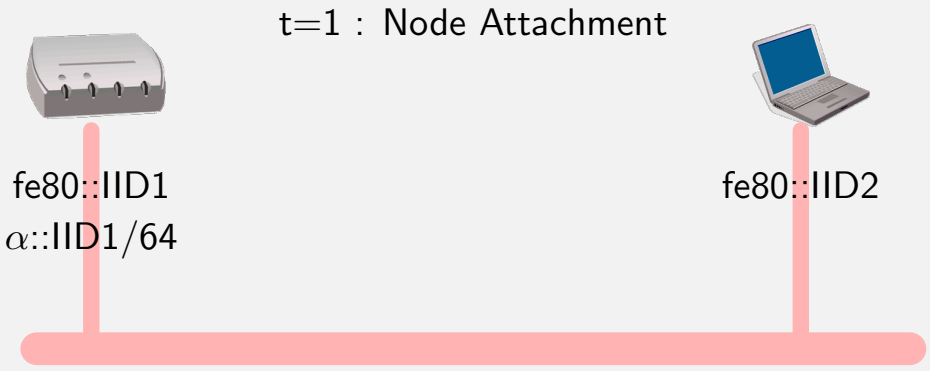
- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

Stateless Auto-configuration: Basic Principles



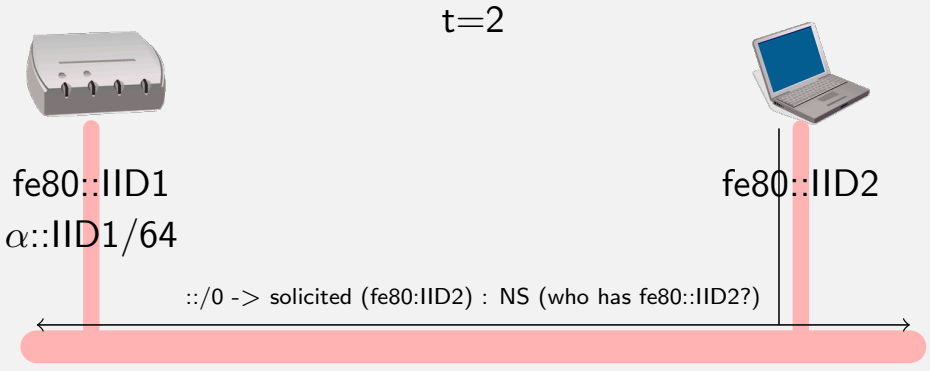
Time t=0: Router is configured with a link-local address and manually configured with a global address (α::/64 is given by the network administrator)

Stateless Auto-configuration: Basic Principles



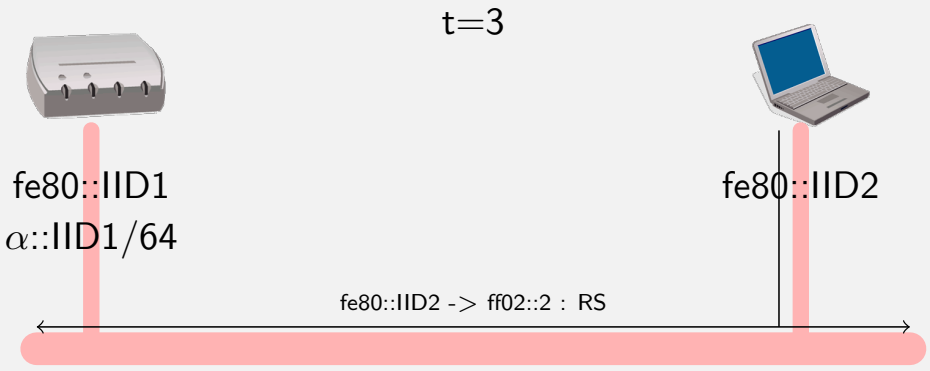
Host constructs its link-local address based on the interface MAC address

Stateless Auto-configuration: Basic Principles



Host does a DAD (i.e. sends a Neighbor Solicitation to query resolution of its own address (tentative): no answers means no other host has this value).

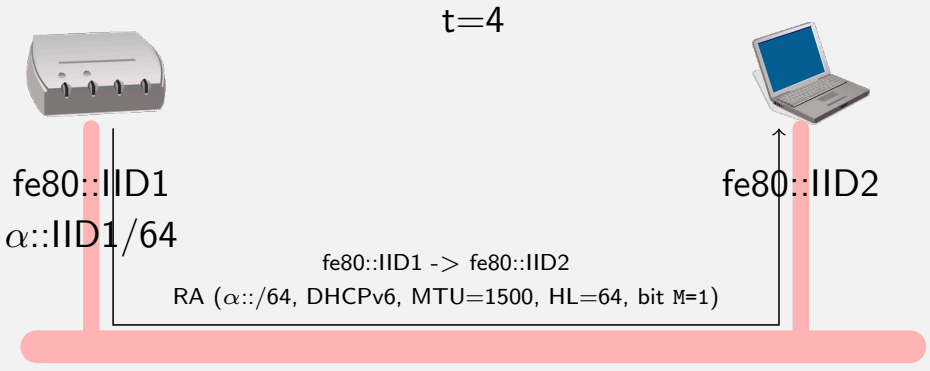
Stateless Auto-configuration: Basic Principles



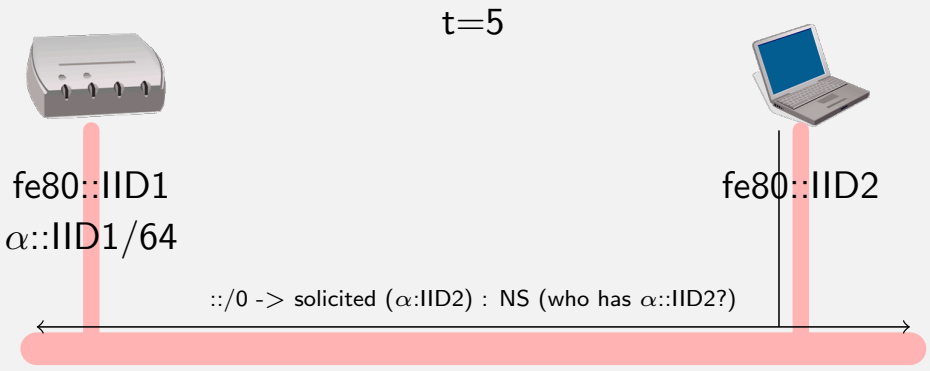
Host sends a Router Solicitation to the Link-Local All-Routers Multicast group using the newly link-local configured address

Associated Protocols & Mechanisms ▶ Neighbor Discovery

Stateless Auto-configuration: Basic Principles

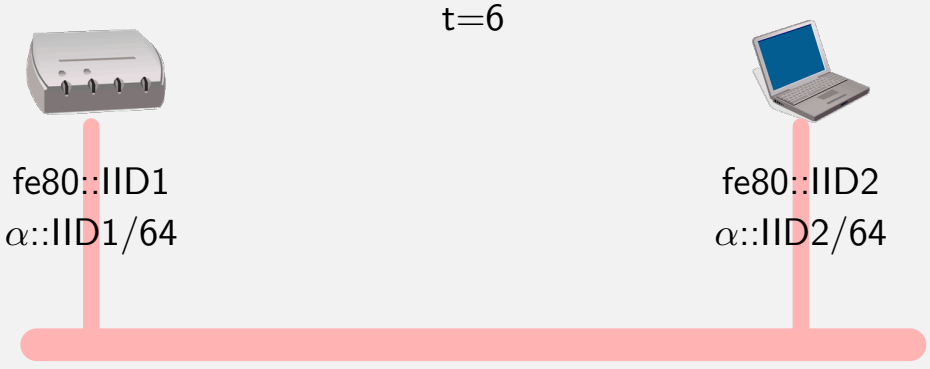


Router directly answers the host using Link-local addresses. The answer may contain a/several prefix(es). Router can also mandate hosts to use DHCPv6 to obtain prefixes (statefull auto-configuration) and/or other parameters (DNS servers...): Bit $\texttt{M} = 1$.



Host does a DAD (i.e. sends a Neighbor Solicitation to query resolution of its own global address: no answers means no other host as this value).

Stateless Auto-configuration: Basic Principles
Associated Protocols & Mechanisms ▶ Neighbor Discovery



Host sets the global address and takes answering router as the default router.

Comments I

Traditionnellement, la configuration d'une interface réseau d'une machine demande une configuration manuelle. C'est un travail souvent long et source d'erreurs. Avec IPv6, cette configuration est automatisée, introduisant par là-même des caractéristiques de fonctionnement immédiat (plug and play) à l'interface réseau. La configuration automatique signifie qu'une machine obtient toutes les informations nécessaires à sa connexion à un réseau local IP sans aucune intervention humaine. Dans le cas idéal, un utilisateur quelconque déballe son nouvel ordinateur, le connecte au réseau local et le voit fonctionner sans devoir y introduire des informations de "spécialiste". Nous allons maintenant étudier l'autre aspect de l'autoconfiguration de IPv6 qui est l'autoconfiguration d'adresses. Celle-ci a pour objectif :

- l'acquisition d'une adresse quand une machine est attachée à un réseau pour la première fois ;
- la possibilité d'attribuer d'autres préfixes, voire de renuméroter une machine.

Le processus d'autoconfiguration d'adresse d'IPv6 comprend la création d'une adresse lien-local, l'attachement aux groupes de multicast sollicités, la vérification de l'unicité de l'adresse lien-local et la construction d'adresses unicast globales.
Le rÙle du routeur est important dans l'autoconfiguration. Il dicte à la machine, par des bits (cf. Annonce du routeur) de l'en-tête du message d'annonce de routeurs, la méthode à retenir et fournit éventuellement les informations nécessaires à sa configuration. Le bit M (Managed address configuration) mis à 1 indique que l'équipement ne doit pas construire lui-même l'adresse à partir de son identifiant d'interface et des préfixes reçus, mais doit explicitement demander son adresse auprès d'une application d'un serveur d'adresses. Le bit O (Other stateful configuration) indique que l'équipement doit interroger le serveur de configuration pour obtenir des paramètres autre que l'adresse. L'algorithme de la procédure d'autoconfiguration d'adresse se décompose de la manière suivante :
La toute première étape consiste à créer l'adresse lien-local. Une fois l'unicité de cette adresse vérifiée, la machine est en mesure de communiquer avec les autres machines du lien. La machine doit chercher à acquérir un message d'annonce du routeur pour déterminer la méthode d'obtention de l'adresse unicast globale. S'il y a un routeur sur le lien, la machine doit appliquer la méthode indiquée par le message d'annonce de routeurs, à savoir :

Comments II

- l'autoconfiguration sans état,
- l'autoconfiguration avec état.

En l'absence de routeur sur le lien, la machine doit essayer d'acquérir l'adresse unicast globale par la méthode d'autoconfiguration avec état. Si la tentative échoue, c'est terminé. Les communications se feront uniquement sur le lien avec l'adresse lien-local. La machine n'a pas une adresse avec une portée qui l'autorise à communiquer avec des machines autres que celles du lien.

t=0 Le routeur est configuré avec une adresse locale et une adresse globale. Le routeur est aussi autoriser à participer au protocole de découverte de voisins.

t=1 à l'initialisation de son interface, la machine construit un identifiant pour l'interface qui doit être unique au lien. Cet identifiant utilise l'adresse EUI-64. Le principe de base de la création d'adresse IPv6 est de marier un préfixe avec l'identifiant. L'adresse lien-local est créée en prenant le préfixe lien-local (fe80::/64) qui est fixé. L'adresse ainsi constituée est encore interdite d'usage. Elle possède un état provisoire car la machine doit vérifier l'unicité de cette adresse sur le lien au moyen de la procédure de détection d'adresse dupliquée. Si la machine détermine l'adresse lien-local n'est pas unique, l'autoconfiguration s'arrête et une intervention manuelle est nécessaire. Une fois que l'assurance sur l'unicité de l'adresse lien-local est obtenue, l'adresse provisoire devient une adresse valide pour l'interface. La première phase de l'autoconfiguration est achevée.

Comments III

t=2 Pour vérifier l'unicité des adresses lien-local ou unicast, les machines doivent exécuter un algorithme de Détection d'Adresse Dupliquée (DAD) avant de les utiliser. L'algorithme utilise les messages ICMPv6 sollicitation d'un voisin et annonce d'un voisin. Si une adresse déjà en service est découverte, elle ne pourra être attribuée à l'interface. L'autoconfiguration s'arrête et une intervention humaine devient obligatoire. Une adresse est qualifiée de "provisoire" pendant l'exécution de l'algorithme DAD et ce jusqu'à la confirmation de son unicité. Une adresse provisoire est assignée à une interface uniquement pour recevoir les messages de sollicitation et d'annonce d'un voisin. Les autres messages reçus sont ignorés. L'algorithme DAD consiste à envoyer un message sollicitation d'un voisin avec dans le champ adresse de la cible l'adresse provisoire. Afin de distinguer l'algorithme DAD de celui de découverte des voisins, le paquet IPv6 contenant un message de sollicitation d'un voisin a comme adresse de source l'adresse indéterminée. Trois cas se présentent :

- Un message annonce d'un voisin est reçu : l'adresse provisoire est utilisée comme adresse valide par une autre machine. L'adresse provisoire n'est pas unique et ne peut être retenue.
- Un message sollicitation d'un voisin est reçu dans le cadre d'une procédure DAD; l'adresse provisoire est également une adresse provisoire pour une autre machine. L'adresse provisoire ne peut être utilisée par aucune des machines.

Rien n'est reçu au bout d'une seconde (valeur par défaut) : l'adresse provisoire est unique, elle passe de l'état de provisoire à celle de valide et elle est assignée à l'interface. A noter que cet algorithme n'offre pas une fiabilité absolue, notamment lorsque le lien est coupé.

Comments IV

t=3 L'autoconfiguration sans état (RFC 2462) ne demande aucune configuration manuelle des machines, une configuration minimum pour les routeurs et aucun serveur supplémentaire. Elle se sert du protocole ICMPv6 et peut fonctionner sans la présence de routeurs. Elle nécessite cependant un sous-réseau à diffusion. Cette méthode ne s'applique que pour les machines et ne peut être retenue pour la configuration des routeurs. Le principe de base de l'autoconfiguration sans état est qu'une machine génère son adresse IPv6 à partir d'informations locales et d'informations fournies par un routeur. Le routeur fournit à la machine les informations sur le sous-réseau associé au lien, il donne le préfixe.

t=4 Comme pour la création de l'adresse lien-local, l'adresse unicast globale est obtenue en concaténant le préfixe avec l'identifiant de l'interface. Le préfixe provient du message d'annonce de routeurs et plus précisément de l'option ´information sur le préfixe[a]. Bien qu'il faille vérifier l'unicité de toutes les adresses unicast, dans le cas d'une adresse unicast obtenue par autoconfiguration sans état cela n'est pas obligatoire. En effet, l'unicité de l'identifiant de l'interface a déjà été contrÙlé dans l'étape de création de l'adresse lien-local. L'identifiant étant le même, il n'y a plus aucune ambiguïté sur son unicité. L'adresse unicast globale constituée est aussi unique que celle lien-local. La renumérotation des machines d'un lien s'effectue au moyen des routeurs qui passent les adresses utilisées dans un état déprécié et annoncent en même temps le nouveau préfixe. Les machines pourront recréer une adresse préférée.

t=5 La machine fait un DAD sur sa nouvelle adresse pour vérifier son unicité

t=6 Si aucune réponse au DAD n'est reçue, l'adresse globale est valide et le routeur ayant annoncé le préfixe est retenu comme routeur par défaut.

Optimistic DAD RFC 4429

- DAD is a long process:
  - Send NS
  - Timeout
  - May be repeated
- For Link-Local and Global addresses
- Mobile nodes are penalized
  - Discover Network
  - Authentication
  - DAD, RS/RA, DAD
- oDAD allows a host to use the address before DAD
- If no answer to DAD then the address becomes a valid one

Comments I

La duplication d'adresses est un processus relativement long puisqu'un équipement qui souhaite garantir l'unicité de son adresses doit être un message NS et attendre une absence de réponse. De plus, comme le réseau peut perdre les messages NS, un équipement peut tenter plusieurs fois de résoudre sa propre adresse avant de la garantir unique. Finalement, le processus se répète pour l'adresse lien-local et l'adresse globale. Il faut donc plusieurs secondes avant qu'un équipement puisse envoyer des paquets sur le réseau. En situation de mobilité, ce délais qui s'ajoute à ceux de la découverte des réseaux disponibles, à l'authentification peut conduire à des ruptures de connectivité (par exemple pour la voix sur IP).

Le RFC 4429 rend plus tolérant la détection d'adresse dupliquée en autorisant un site à utiliser son adresse bien qu'elle n'ait pas été encore garantie unique. Ce comportement est appelé DAD optimiste (optimistic DAD). L'état tentative de l'adresse (voir Cycle de vie d'une adresse est remplacé par l'état optimiste pendant lequel l'unicité de l'adresse n'est pas garanti mais qui permet son utilisation. En parallèle, un DAD classique est lancé. les messages NS sont émis avec le bit O (Override) à 0 pour que les caches ND ne soit pas mis à jour au cas où cette adresse existerait déjà sur le réseau.

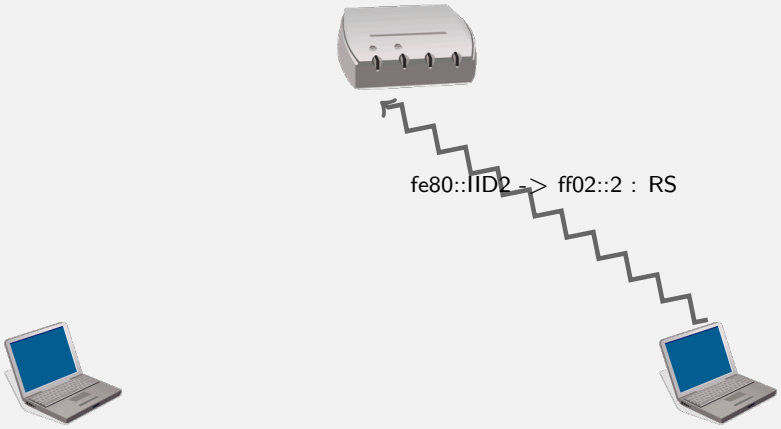
Associated Protocols & Mechanisms

Non-Broadcast Multiple Access (NBMA) Networks

NBMA Networks

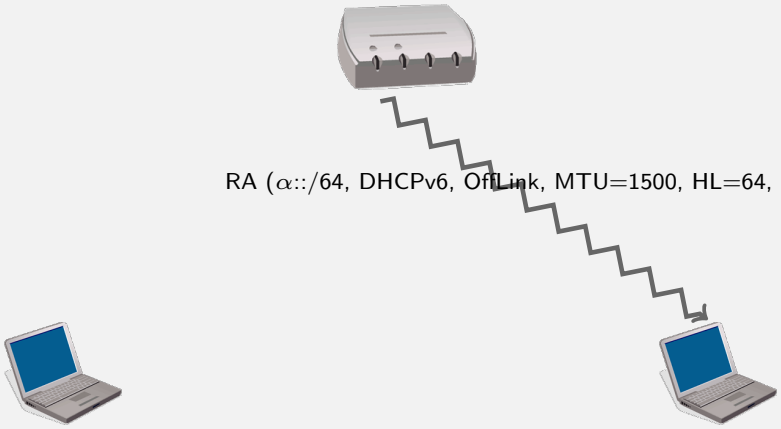
- NDP can handle efficiently NBMA networks
  - Every host can be joined separately, but no broadcast
  - Telephony network, ATM...
- Off-link bit is RA by the router to inform of a NBMA network
  - 3G, Sensor Networks (broadcast expensive)
- All packets are sent to to the router, which will forward to destination
  - No NS
  - ICMP Redirect can be used.

Off Link example Optional

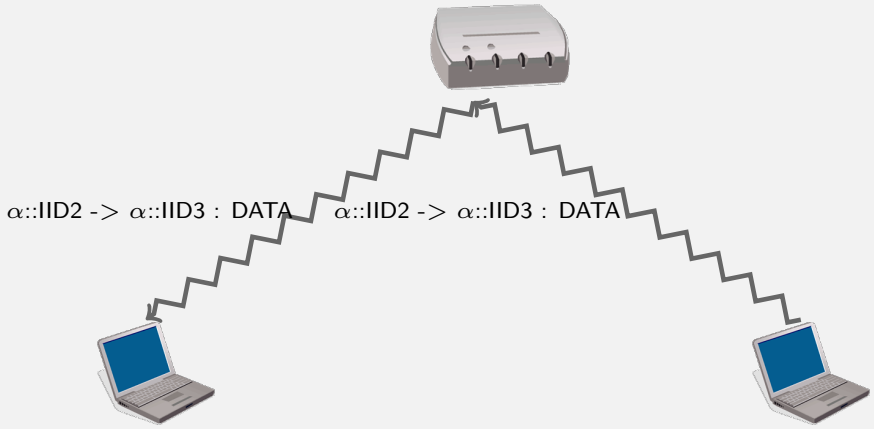


Off Link example Optional

Associated Protocols & Mechanisms ▶ Non-Broadcast Multiple Access (NBMA) Networks

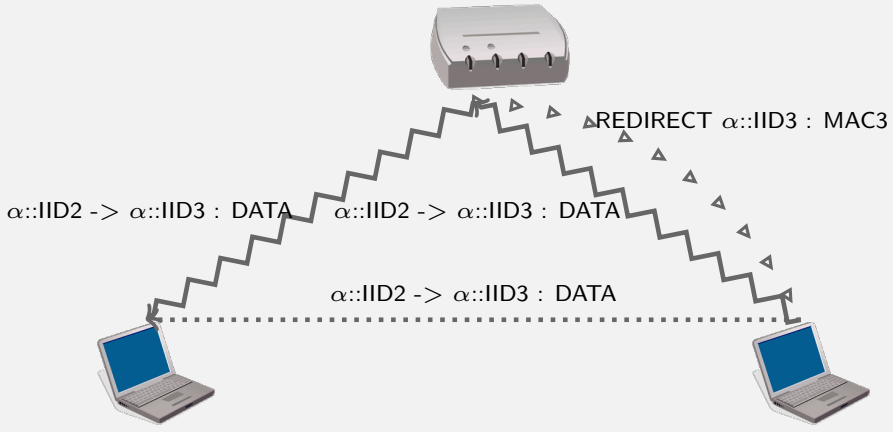


Off Link example Optional



Off Link example Optional

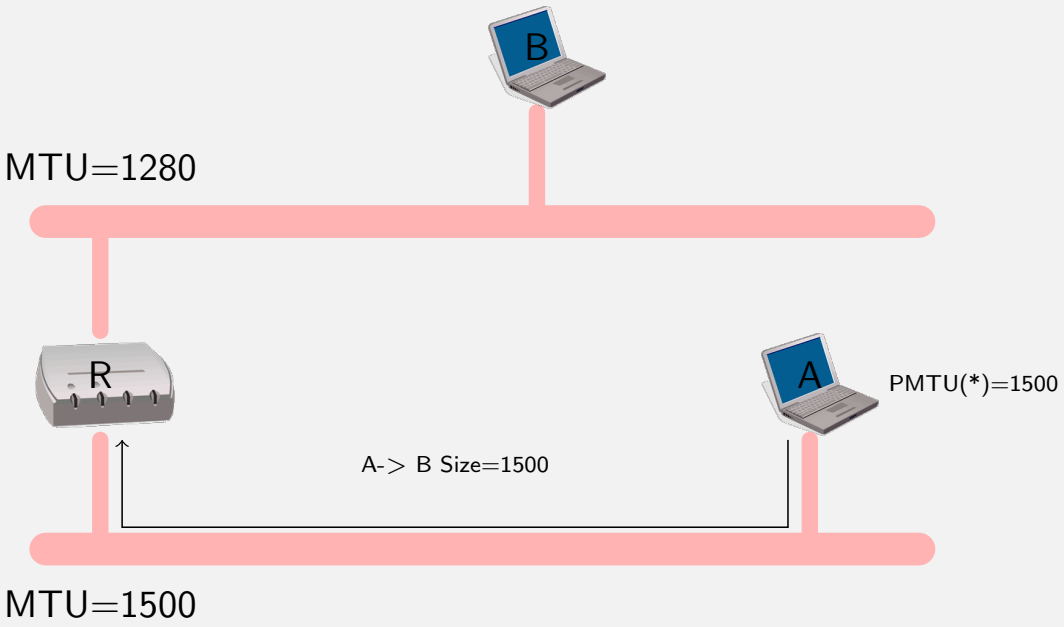
Associated Protocols & Mechanisms ▶ Non-Broadcast Multiple Access (NBMA) Networks



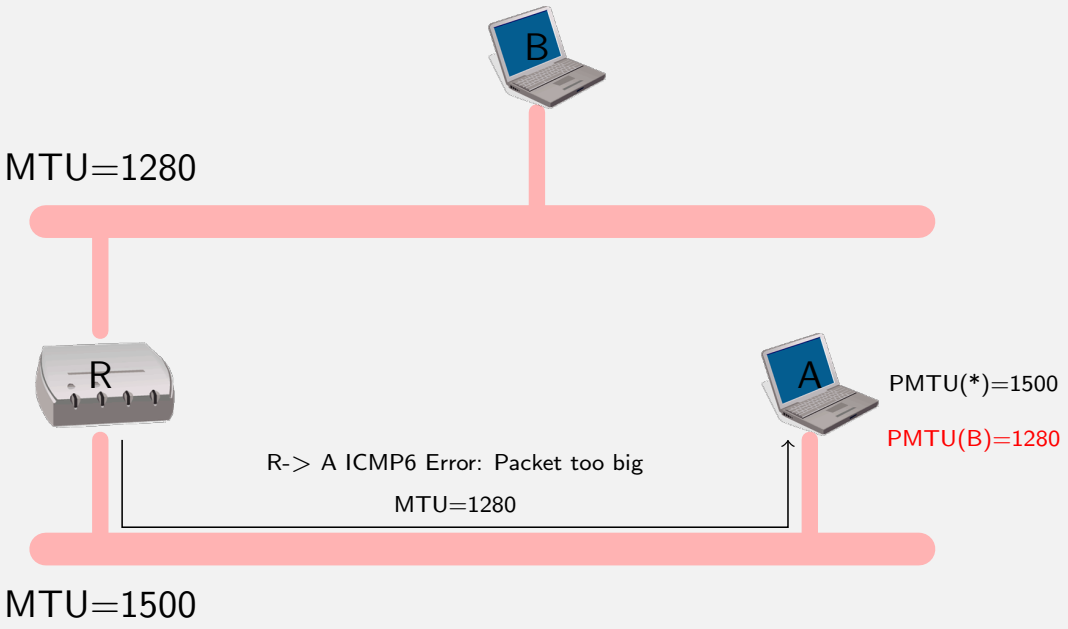
Associated Protocols & Mechanisms

Path MTU discovery

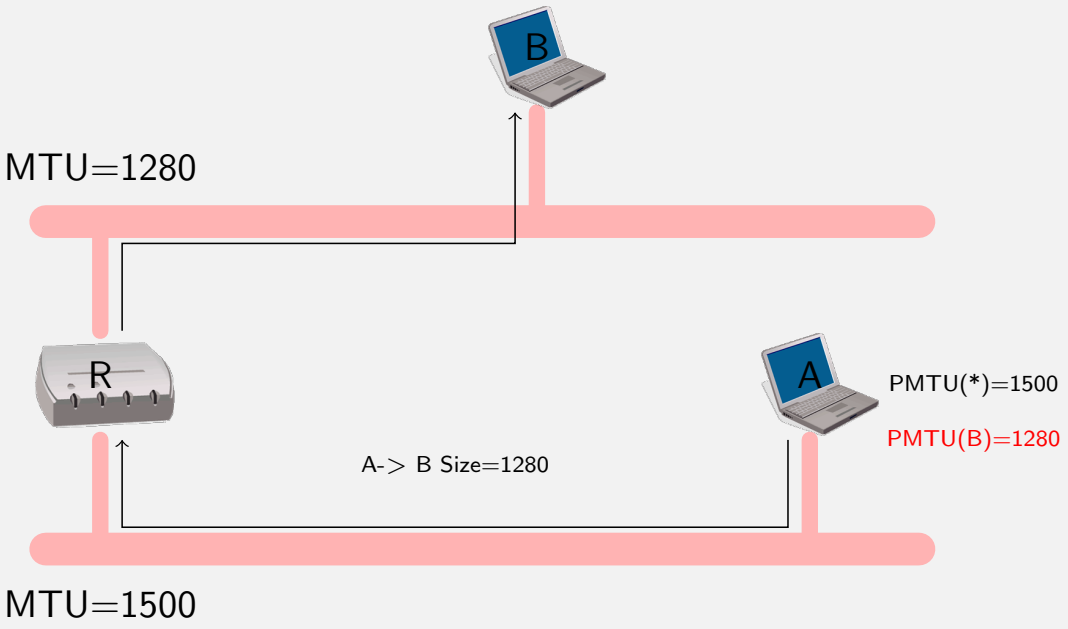
Path MTU discovery for IPv6 (RFC 1981)



Path MTU discovery for IPv6 (RFC 1981)



Path MTU discovery for IPv6 (RFC 1981)



Comments I

Pour des considérations d'efficacité, il est généralement préférable que les informations échangées entre équipements soient contenues dans des datagrammes de taille maximale. Cette taille dépend du chemin suivi par les datagrammes et est égale à la plus grande taille autorisée par l'ensemble des liens traversés. Elle est de ce fait appelée PMTU, ou Path Maximum Transmission Unit (unité de transfert de taille maximale sur le chemin).
Initialement, l'équipement émetteur fait l'hypothèse que le PMTU d'un certain chemin est égal au MTU du lien auquel il est directement attaché. S'il s'avère que les paquets transmis sur ce chemin excèdent la taille maximale autorisée par un lien intermédiaire, alors le routeur associé détruit ces paquets et retourne un message d'erreur ICMPv6 de type ´paquet trop grand[a], en y indiquant le MTU accepté. Fort de ces informations, l'équipement émetteur réduit le PMTU supposé pour ce chemin.
Plusieurs itérations peuvent être nécessaires avant d'obtenir un PMTU permettant à tout paquet d'arriver à l'équipement destinataire sans jamais excéder le MTU de chaque lien traversé. Le protocole IPv6 garantit que le MTU de tout lien ne peut descendre en dessous de 1 280 octets, valeur qui constitue ainsi une borne inférieure pour le PMTU. Ce protocole reposant sur la perte de paquets, il est laissé le soin aux couches supérieures de gérer la fiabilité de la communication en retransmettant si nécessaire (paquet 6 de l'exemple). Figure : Découverte du MTU seconde phase: réception d'un message ICMPv6
Si la détermination du PMTU se fait essentiellement lors des premiers échanges entre les équipements concernés, elle peut également être revue en cours de transfert si, suite à un changement de route, un lien plus contraignant est traversé.
L'émetteur vérifie aussi que le PMTU n'a pas augmenté en envoyant de temps en temps un paquet plus grand. Si celui-ci traverse le réseau sans problème, la valeur du PMTU est augmentée.
Signalons enfin que l'algorithme de découverte du PMTU fonctionne indifféremment avec des échanges point-à-point ou multipoints. Dans ce dernier cas, le PMTU sera le PMTU minimal permis par l'ensemble des chemins vers chaque site destinataire du groupe de diffusion.
L'exploitation de l'information de PMTU se fait de plusieurs façons suivant l'endroit o˘ les données à transmettre sont segmentées :

Comments II

si un protocole de type TCP est utilisé, celui-ci assurera la segmentation de façon transparente pour les applications, en fonction des informations de PMTU que pourra lui communiquer la couche IPv6. si un protocole de type UDP est utilisé, alors cette segmentation devra être assurée par une couche supérieure, éventuellement l'application. Il faut donc que celle-ci

- (1) puisse être informée du PMTU autorisé, même dans le cas o˘ celui-ci change par la suite, et
- (2) puisse segmenter ses données en conséquence. Parce que ces deux conditions ne sont pas toujours réunies, IPv6 a conservé un mécanisme de fragmentation (voir fragmentation).

Un deuxième aspect concerne l'identification des chemins afin de pouvoir y associer les informations de PMTU. Plusieurs possibilités, laissées à l'implémenteur, sont possibles. Un chemin peut être identifié par l'adresse destination, ou par l'identificateur de flux si celui-ci est utilisé, ou par la route suivie dans le cas o˘ elle est imposée (voir routage).

Enfin, s'il est fortement recommandé que chaque équipement supporte le mécanisme de recherche du PMTU, ce n'est pas obligatoire. Ainsi, un équipement qui n'en dispose pas (par exemple une ROM de boot) devra restreindre la taille de tout paquet transmis au MTU minimal que doit supporter tout lien, soit 1280 octets.

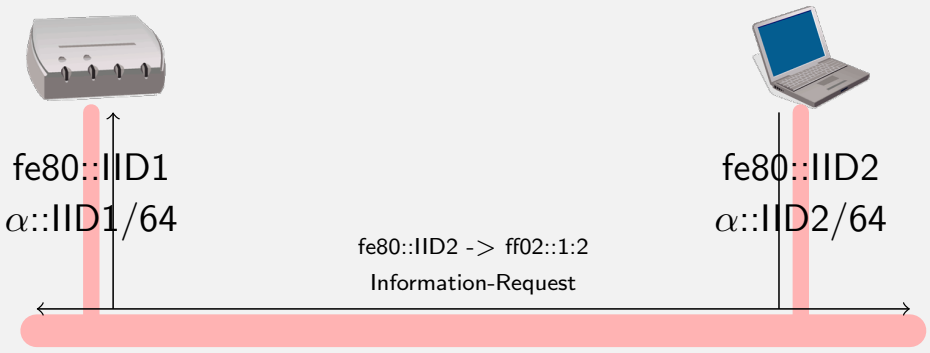
Associated Protocols & Mechanisms

Examples

Router Configuration Example

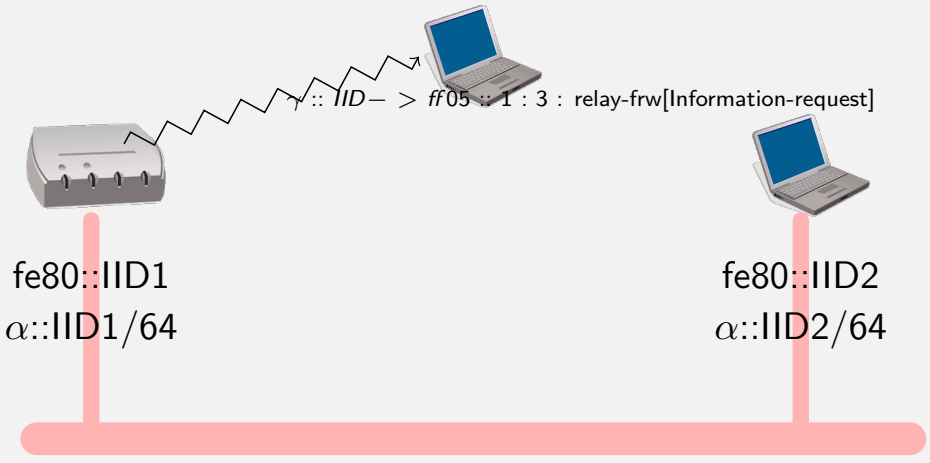

```
interface Vlan5
 description reseau C5
 ip address 192.108.119.190 255.255.255.128
...
 ipv6 address 2001:660:7301:1::/64 eui-64
 ipv6 enable
 ipv6 nd ra-interval 10
 ipv6 nd prefix-advertisement 2001:660:7301:1::/64 2592000\
 604800 onlink autoconfig
```

Stateless DHCPv6 (RFC 3736): With static parameters



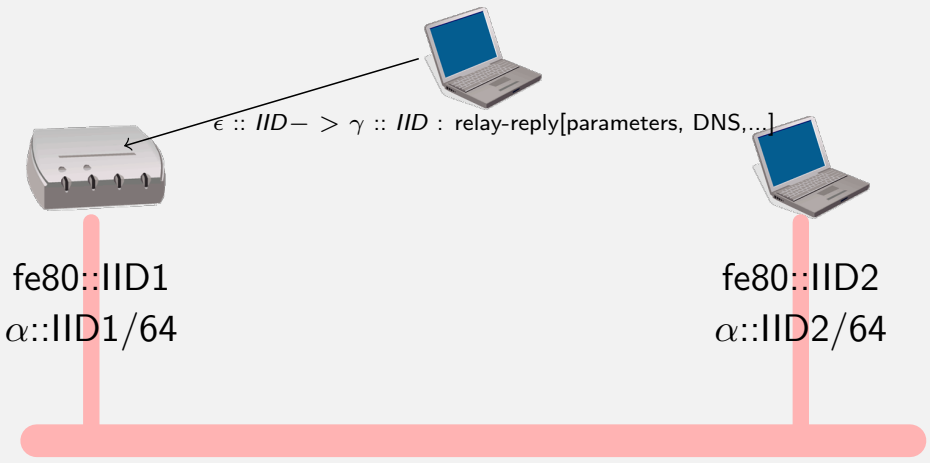
Host needs only static parameters (DNS, NTP,...). It sends an Information-Request message to All_DHCP_Agents multicast group. The scope of this address is link-local.

Stateless DHCPv6 (RFC 3736): With static parameters



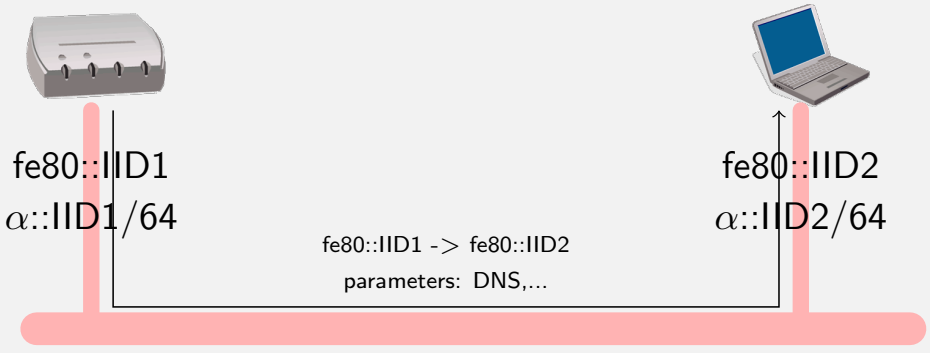
A relay (generally the router) encapsulates the request into a *Forward message and sends it either to the All_DHCP_Servers site-local multicast group or to a list of pre-defined unicast addresses.*

Stateless DHCPv6 (RFC 3736): With static parameters



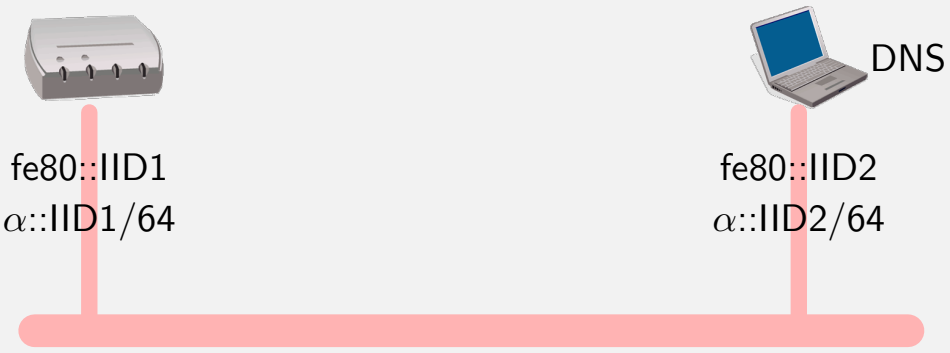
The server responds to the relay

Stateless DHCPv6 (RFC 3736): With static parameters



The router extracts information from the message to create answer and sends information to the host

Stateless DHCPv6 (RFC 3736): With static parameters



Host is now configured to resolve domain names through the DNS

Associated Protocols & Mechanisms

Neighbor Discovery Security

Security issues with Neighbor Discovery

From an attacker point of view, IPv6 attacks are:

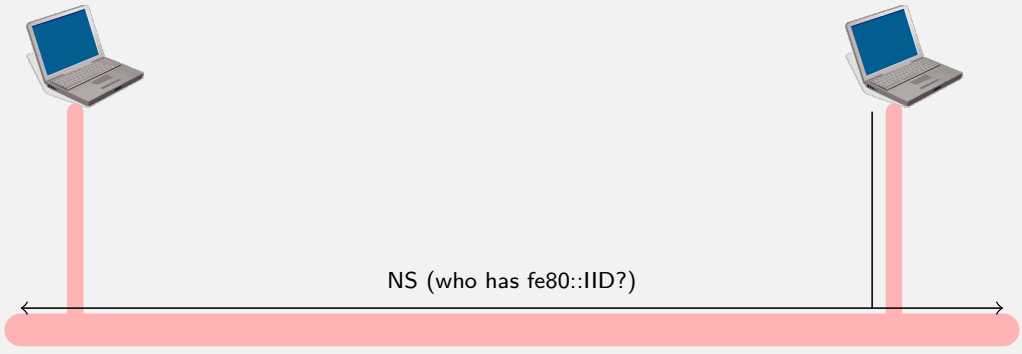
- **Difficult** from remote network:
  - Scanning IPv6 network is hard (2^{64} addresses)
    - May use random IID instead of MAC-based IID (if needed)
  - No broadcast address
  - Remote attacks would mainly target hosts exposed through the DNS
- **Easy** from local network:
  - Neighbor Discovery is basically not secured (see SEND later)
  - Attacks inspired by ARP flaws + new attacks
  - Implementations not (yet) heavily tested

Attacker toolkits already available !

See `http://www.thc.org/thc-ipv6/`

Examples of attacks using ND

Neighbor Discovery Snooping

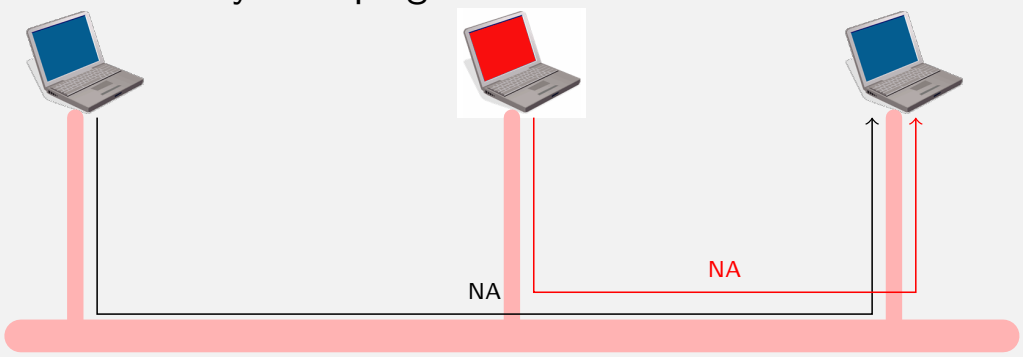


Host uses Neighbor Discovery notably in these two cases:

- To get the link-layer information (typically the MAC address) of another host (ARP-like)
- To verify address uniqueness (DAD)

Examples of attacks using ND

Neighbor Discovery Snooping

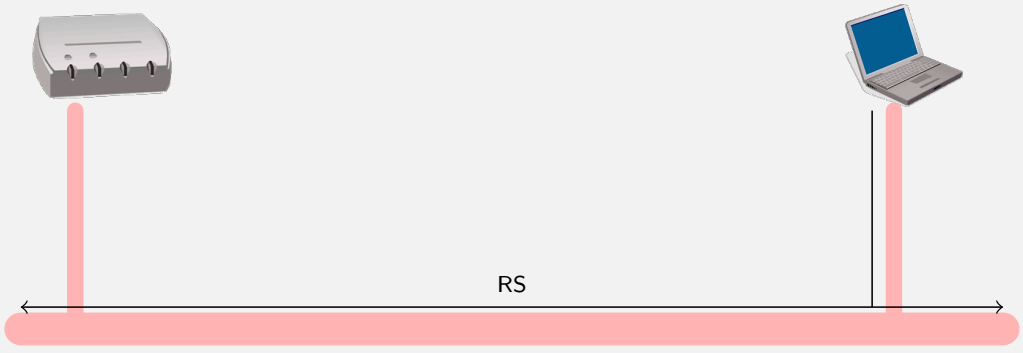


An attacker on the LAN can perform an attack by responding to ND messages

- ARP-like: Claim to be a given host on the LAN => Man in the Middle
- DAD: Claim to have any address asked for on the LAN => Deny of Service

Examples of attacks using ND

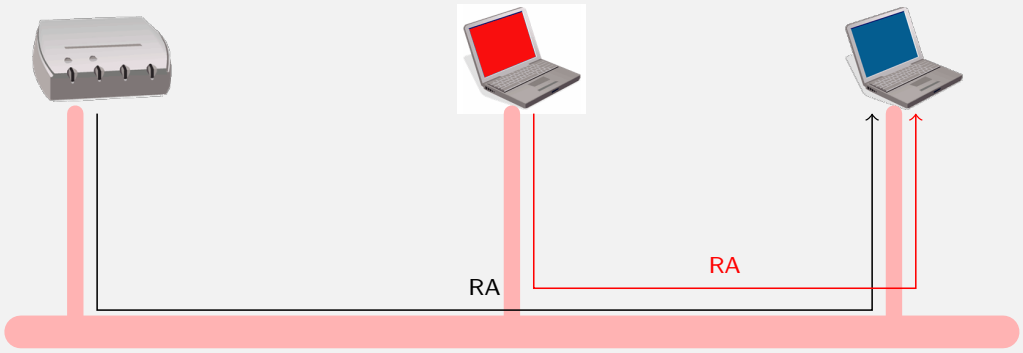
Rogue router



Host uses the Router Solicitation to get the address of the exit router and the prefix used on the LAN.

Examples of attacks using ND

Rogue router



An attacker on the LAN can perform an attack by responding to RS messages

- Claim to be the exit router => Man in the Middle
- Claim to route another prefix on the LAN => Deny of Service

Solutions to mitigate or prevent attacks?

Prevention of attacks:

- SEND (Secure Neighbor Discovery)
  - IETF proposed solution: RFC 3971 (note: too complex to deploy for an average site!)
  - Use signed ND messages, with a trust relationship
- Level-2 Filtering
  - Filter ND on switch port (ex. only one port allowed to send RA)
  - A few switch still implements it ... (Cisco ?)

Detection of attacks: ndpmon

- Similar to ARP-watch
- Detect Snooping and Denial of Services
- `http://ndpmon.sf.net`

Example: Interface during an IETF meeting


```
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   inet6 fe80::223:6cff:fe97:679c%en3 prefixlen 64 scopeid 0x6
   inet6 2002:8281:1c8c:d:223:6cff:fe97:679c prefixlen 64 autoconf
   inet6 2002:c15f:2011:d:223:6cff:fe97:679c prefixlen 64 autoconf
   inet6 fec0::d:223:6cff:fe97:679c prefixlen 64 autoconf
   inet6 2001:df8::24:223:6cff:fe97:679c prefixlen 64 autoconf
   inet 130.129.28.215 netmask 0xfffff800 broadcast 130.129.31.255
   inet6 2002:8281:1ccb:9:223:6cff:fe97:679c prefixlen 64 autoconf
   inet6 fec0::9:223:6cff:fe97:679c prefixlen 64 autoconf
   ether 00:23:6c:97:67:9c
   media: autoselect status: active
   supported media: autoselect
```

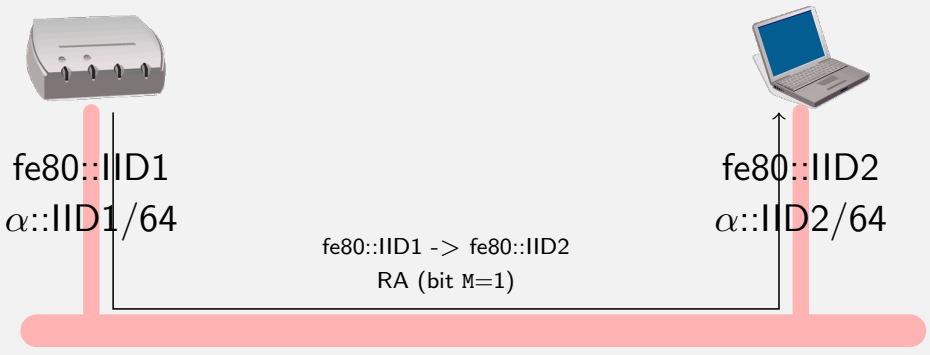

How to solve wrong RA

- SeND: Secure Neighbor Discovery
  - Use of cryptography to protect and authenticate announcements
  - Protect against bad guys
  - Complex and not verify flexible
- SAVI : Source Address Validation
  - : Work in Progress: see `http://tools.ietf.org/html/draft-ietf-savi-framework-01`
  - Implement in switches functions to control announcements
  - Flexible, but not a strong protection
  - Under experimentation
- Otherwise filter announcements with a firewall

Associated Protocols & Mechanisms

DHCPv6

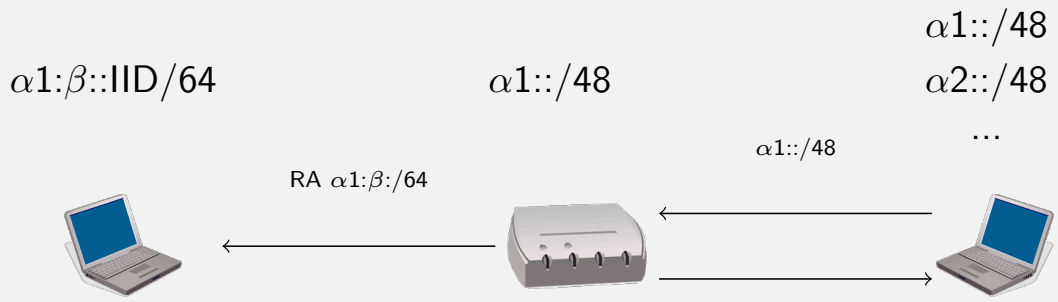
DHCPv6 : Stateful Auto-Configuration



Router responds to RS with a RA message with bit `M` set to 1. Host should request its IPv6 address from a DHCPv6 server.

DHCPv6 : Prefix Delegation

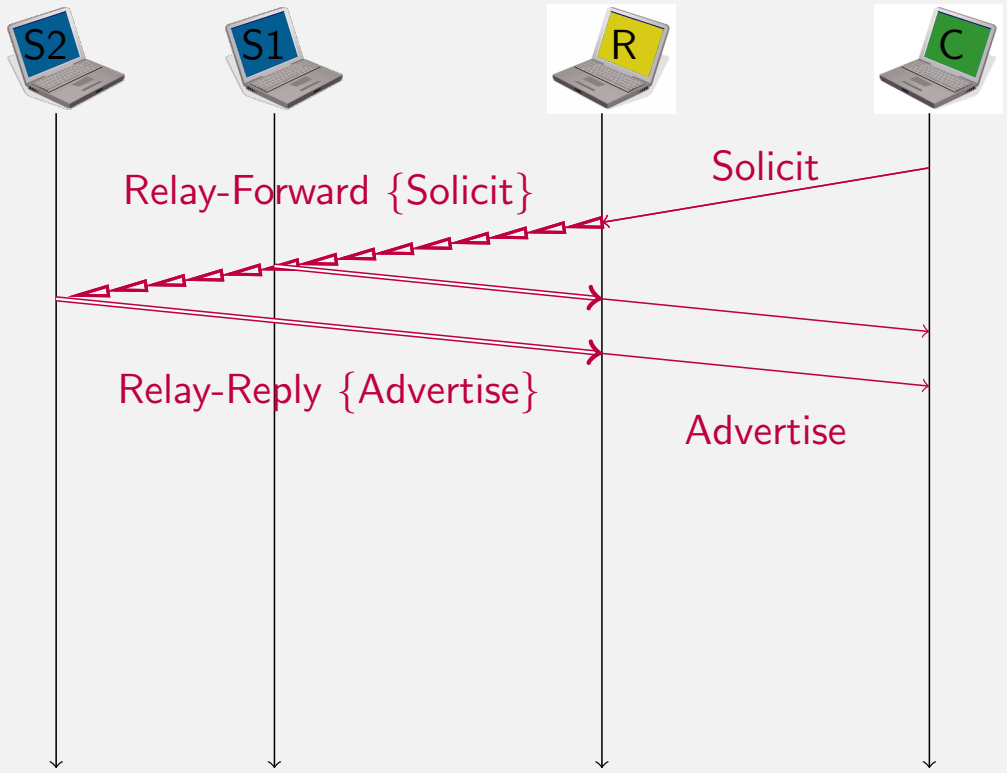
- Dynamic configuration for routers
- ISP solution to delegate prefixes over the network



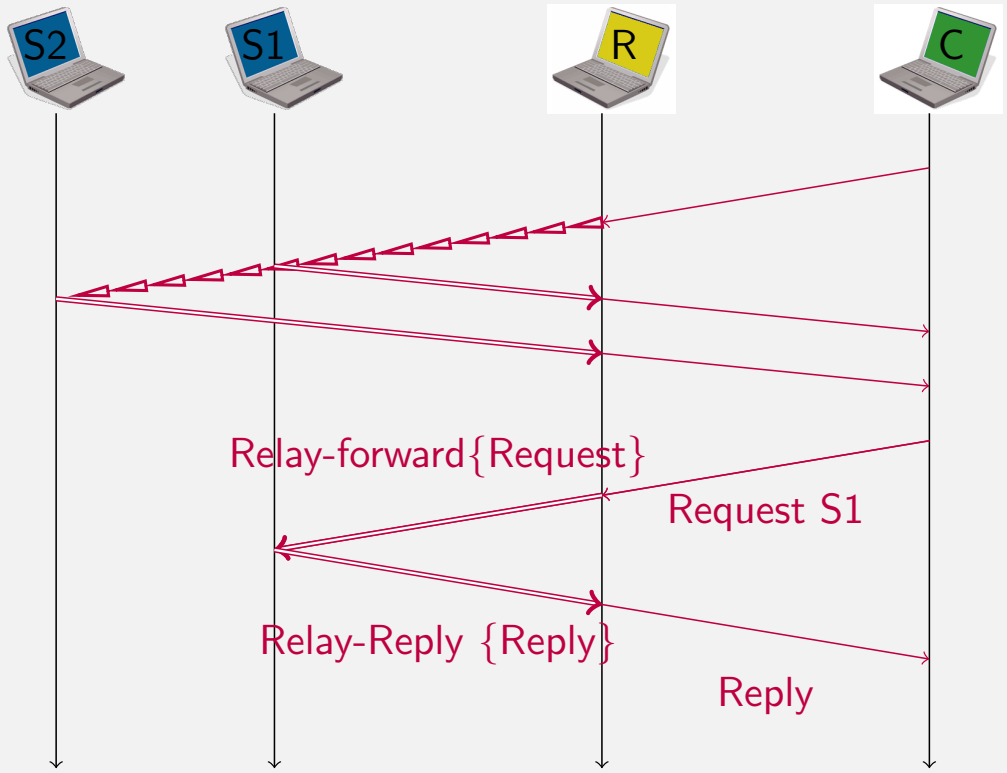
DHCPv6 Full Features

- For address or prefix allocation information form only one DHCPv6 must be taken into account. Four message exchange :
  - Solicit : send by clients to locate servers
  - Advertise : send by servers to indicate services available
  - Request : send by client to a specific server (could be through relays)
  - Reply : send by server with parameters requested
- Addresses or Prefixes are allocated for certain period of time
  - Renew : Send by the client tells the server to extend lifetime
  - Rebind : If no answer from renew, the client use rebind to extend lifetime of addresses and update other configuration parameters
  - Reconfigure : Server informs availability of new or update information. Clients can send renew or Information-request
  - Release : Send by the client tells the server the client does not need any longer addresses or prefixes.
  - Decline : to inform server that allocated addresses are already in use on the link

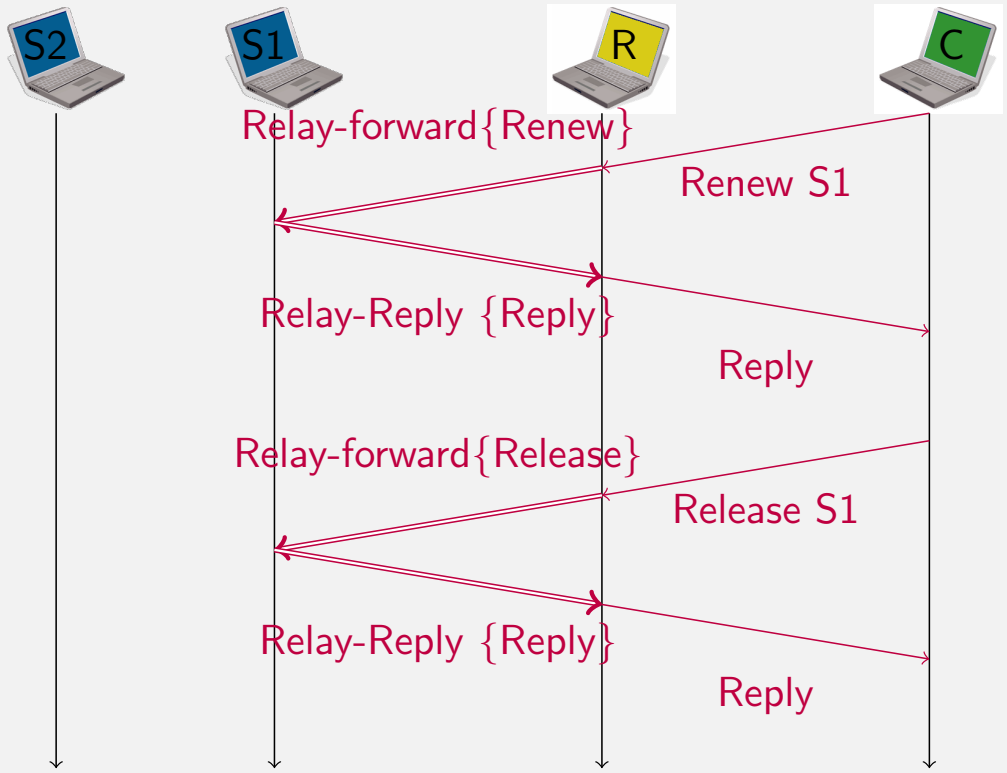
DHCPv6 Scenarii



DHCPv6 Scenarii



DHCPv6 Scenarii



DHCPv6 Identifiers

- DHCPv6 defines several stable identifiers
- After a reboot, the host can get the same information.
- DUID (DHCPv6 Unique IDentifier) :
  - Identify the client
  - Variable length:
    - Link-layer address plus time
    - Vendor-assigned unique ID based on Enterprise Number
    - Link-layer address
- For instance:

```
>od -x /var/db/dhcp6c_duid
0000000 000e 0100 0100 5d0a 5233 0400 9e76 0467
```

DHCPv6 Identifier : IA and IA_PD

- IA and IA_PD are used to link Request and Reply
  - IA is used for Address Allocation and is linked to an Interface
  - IA_PD is used for Prefix Delegation and can be shared among interfaces
- They must be stable (e.g. defined in the configuration file)

Associated Protocols & Mechanisms

Stateless vs Stateful

Auto-configuration: Stateless vs. Stateful

Stateless

Pro:

- Reduce manual configuration
- No server, no state (the router provides all information)

Cons:

- Non-obvious addresses
- No control on addresses on the LAN

Stateful (DHCPv6)

Pro:

- Control of addresses on the LAN
- Control of address format

Cons:

- Requires an extra server
- Still needs RA mechanism
- Clients to be deployed

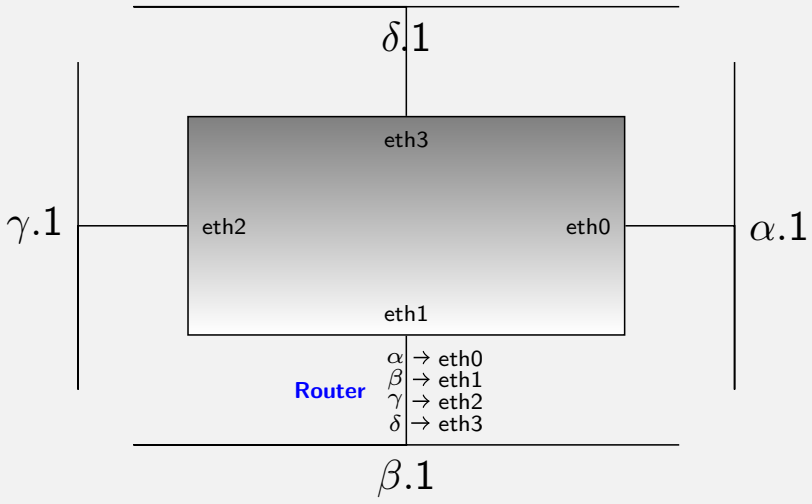
- Stateless: Typically, for Plug-and-Play networks (Home Network)
- Stateful: Typically, for administrated networks (enterprise, institution)

Forwarding Tables

F.I.B

Router configuration

- Router's interface must be manually configured
  - assign an address + a prefix length
  - routers install automatically in the FIB the prefix assigned to the interface



Example : Cisco router

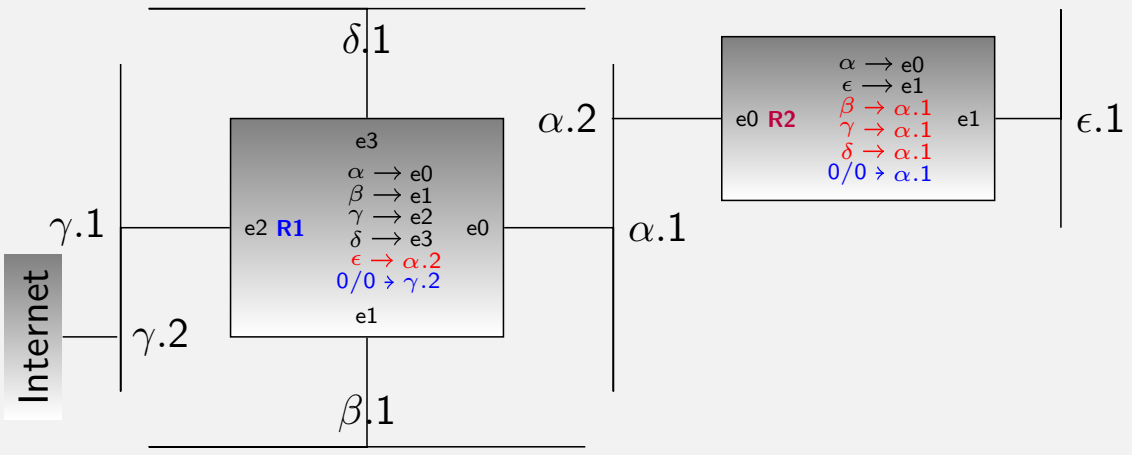

```
RouterB25#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route


Gateway of last resort is 193.50.69.73 to network 0.0.0.0


B 193.52.74.0/24 [200/0] via 193.50.69.74, 1d05h
192.44.77.0/24 is variably subnetted, 2 subnets, 2 masks
B 192.44.77.0/24 [200/0] via 193.50.69.74, 1d05h
C 192.44.77.96/28 is directly connected, GigabitEthernet0/1.16
10.0.0.0/24 is subnetted, 6 subnets
S 10.1.1.0 [1/0] via 192.108.119.196
S 10.35.1.0 [1/0] via 192.108.119.205
S 10.35.4.0 [1/0] via 192.108.119.162
S 10.35.30.0 [1/0] via 192.108.119.160
C 10.51.0.0 is directly connected, GigabitEthernet0/1.51
C 10.49.0.0 is directly connected, GigabitEthernet0/1.49
192.108.119.0/24 is variably subnetted, 6 subnets, 4 masks
C 192.108.119.128/25 is directly connected, GigabitEthernet0/1.5
....
```

Example : Unix


```
% ifconfig en1 192.168.2.100/24
% netstat -rnf inet
Routing tables

Internet:
Destination        Gateway            Flags     Refs      Use  Netif Expire
default            192.168.2.1        UGSc        30       16    en1
10.37.129/24       link#8             UCS          1        0    en2
10.37.129.2        127.0.0.1          UHS          0      160    lo0
10.37.129.255      link#8             UHLWb        2      131    en2
10.211.55/24       link#9             UCS          2        0    en3
10.211.55.2        127.0.0.1          UHS          0       39    lo0
10.211.55.255      link#9             UHLWb        1       76    en3
127                127.0.0.1          UCS          0        0    lo0
127.0.0.1          127.0.0.1          UH           9   252039    lo0
169.254            link#5             UCS          0        0    en1
192.168.2          link#5             UC           1        0    en1
192.168.2.1        0:13:10:83:d5:3c   UHLW         3       54    en1   1194
192.168.2.100      127.0.0.1          UHS          0        3    lo0
```


Static route

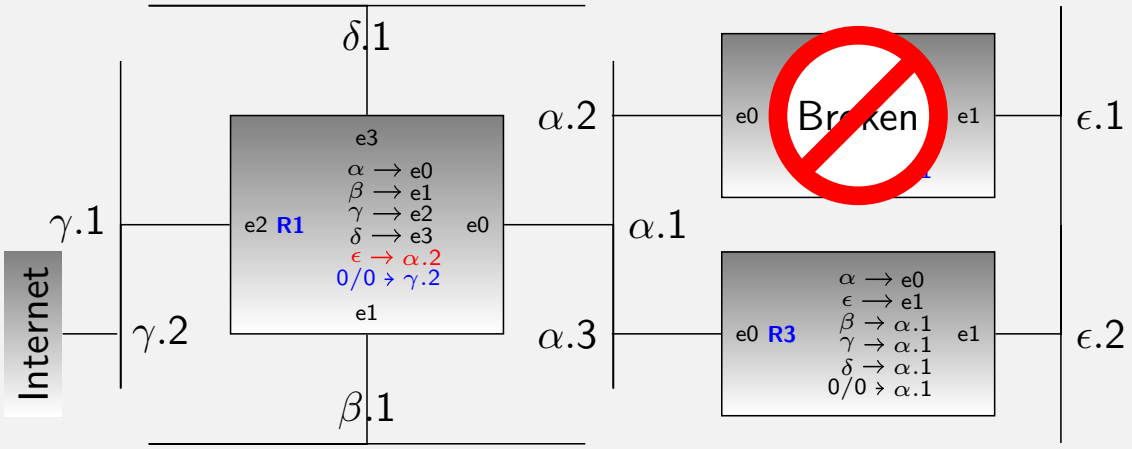
- Manually configured routes



- simple to configure, but subject to errors (loops)
- cannot find another path if router fails

Static route

- Manually configured routes



- simple to configure, but subject to errors (loops)
- cannot find another path if router fails

Example : commands

- BSD:
  - `route add 10.1.2.0/24 10.0.0.1`
  - `route add default 10.0.0.1`
- Linux:
  - `route add 10.1.2.0/24 gw 10.0.0.1`
  - `route add default gw 10.0.0.1`
- Cisco:
  - `ip route 10.1.2.0 255.255.255.0 10.0.0.1`
  - `ip route 0.0.0.0 0.0.0.0 10.0.0.1`

Example : Cisco router


```
RouterB25#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route


Gateway of last resort is 193.50.69.73 to network 0.0.0.0


B 193.52.74.0/24 [200/0] via 193.50.69.74, 1d05h
192.44.77.0/24 is variably subnetted, 2 subnets, 2 masks
B 192.44.77.0/24 [200/0] via 193.50.69.74, 1d05h
C 192.44.77.96/28 is directly connected, GigabitEthernet0/1.16
10.0.0.0/24 is subnetted, 6 subnets
S 10.1.1.0 [1/0] via 192.108.119.196
S 10.35.1.0 [1/0] via 192.108.119.205
S 10.35.4.0 [1/0] via 192.108.119.162
S 10.35.30.0 [1/0] via 192.108.119.160
C 10.51.0.0 is directly connected, GigabitEthernet0/1.51
C 10.49.0.0 is directly connected, GigabitEthernet0/1.49
192.108.119.0/24 is variably subnetted, 6 subnets, 4 masks
C 192.108.119.128/25 is directly connected, GigabitEthernet0/1.5
....
```


Forwarding Tables

Routing Protocols

Routing Protocol

Definition

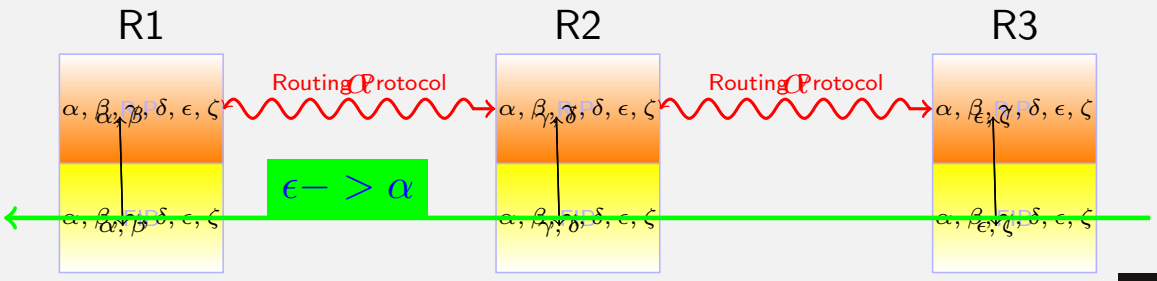
A routing protocol is an application sharing knowledge among routers to construct FIB

- The application may have its own database different from the FIB
- Two families of routing protocol are defined:
  - Interior Gateway Protocol
    - Simple configuration (even if protocol can be complex)
    - Discover other routers, and exchange information
    - Ex: RIPv2 (Distant Vector), OSPF or IS-IS (Link State)
  - Exterior GAteway Protocol
    - A lot of controls, nothing is automatically learnt
    - Complex configuration to hide or show prefixes
    - Used mainly between providers
    - Only one protocol used : Border Gateway Protocol

TELECOM Bretagne

router's behavior

- All router have a manual configuration of their interfaces
- Prefixes learnt from this configuration are spread among other routers
- Routers select announcement and add these prefixes in their FIB
- Routing announcement and traffic (forwarding) go the opposite direction

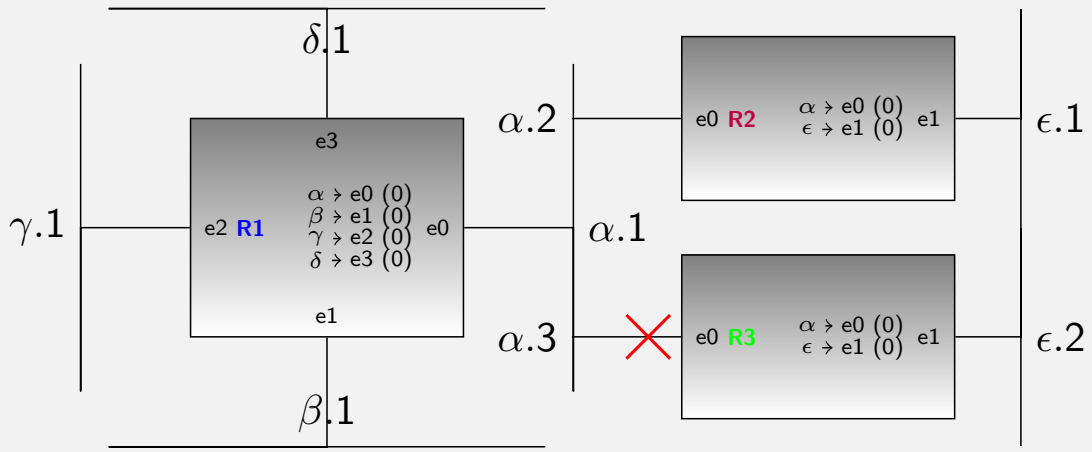


TELECOM Bretagne

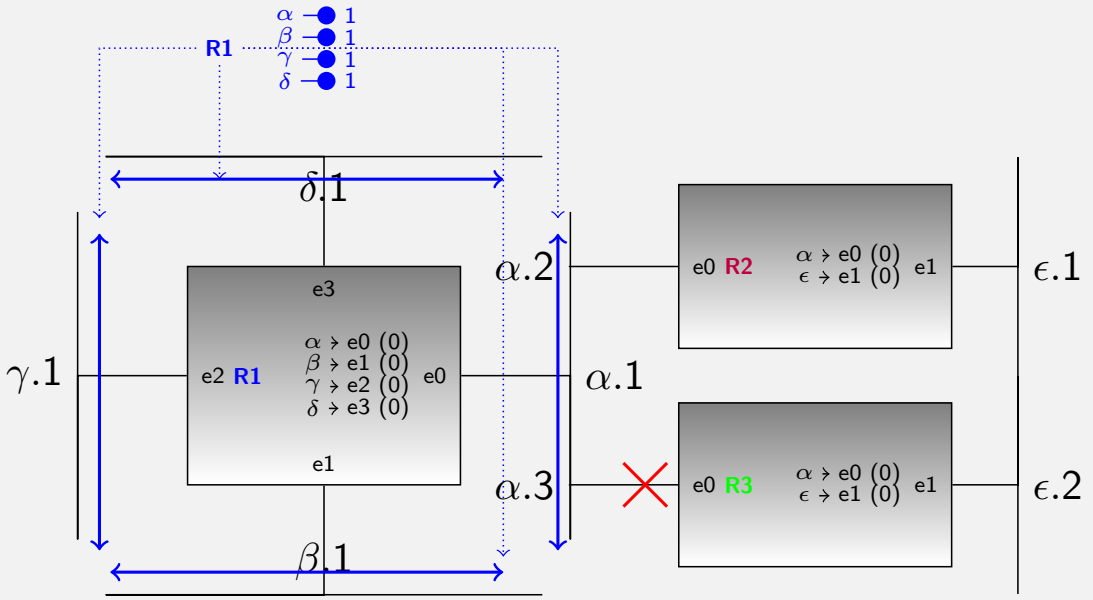
Distant Vector Algorithm

- Very simple algorithm
- Routing Protocol database and FIB are common
  - A cost (usually the number of router to cross) is just added to each prefix
- Router periodically broadcast their FIB on each link they are connected
- If a router find a new prefix in broadcasted information, ...
- or if an broadcasted prefix as a lower cost than the one already stored
  - This entry is added/changed in the FIB
  - Next Hop is set to the IP address of the broadcasting router
  - Cost is increased by one
- Otherwise the information is ignored
- Current implementation:
  - IPv4: RIPv2 (RFC 1723)
  - IPv6: RIPng (RFC 2080)

Distant Vector Example (Add Route)

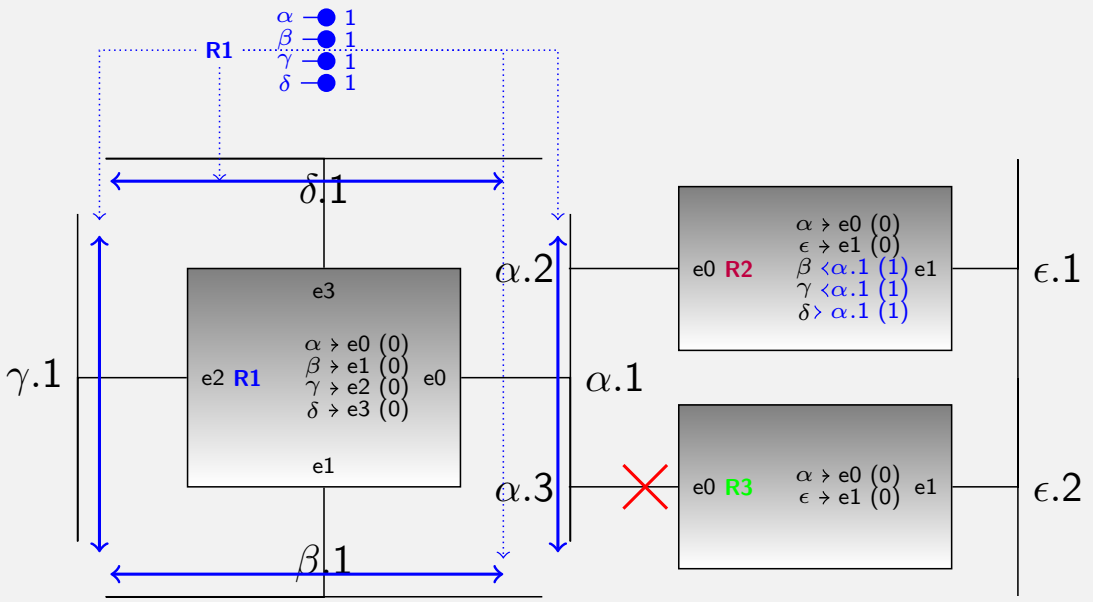


Distant Vector Example (Add Route)



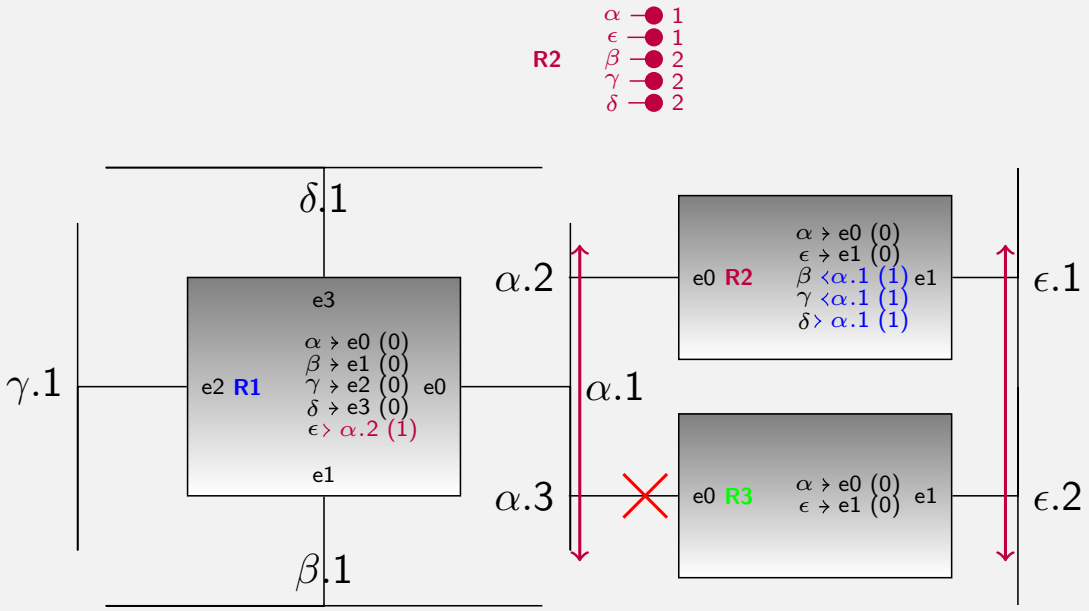
TELECOM Bretagne

Distant Vector Example (Add Route)



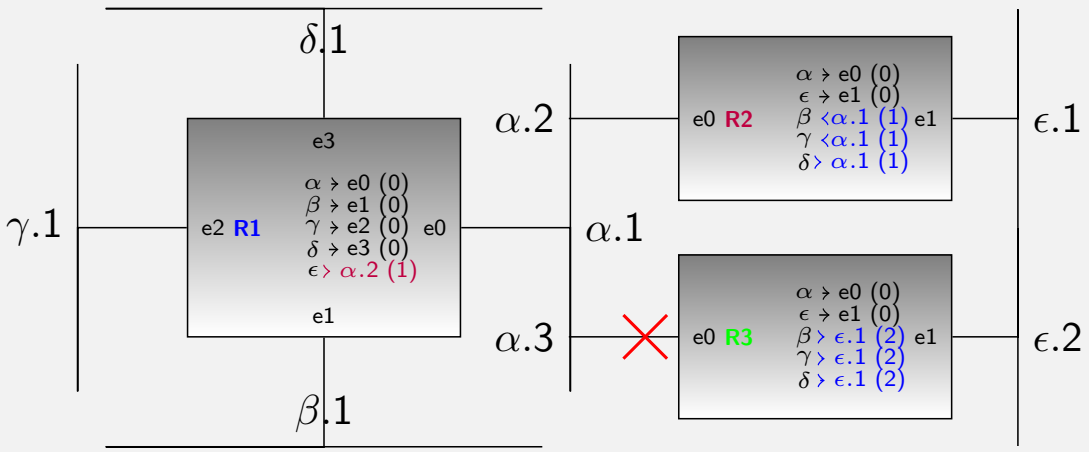
TELECOM Bretagne

Distant Vector Example (Add Route)



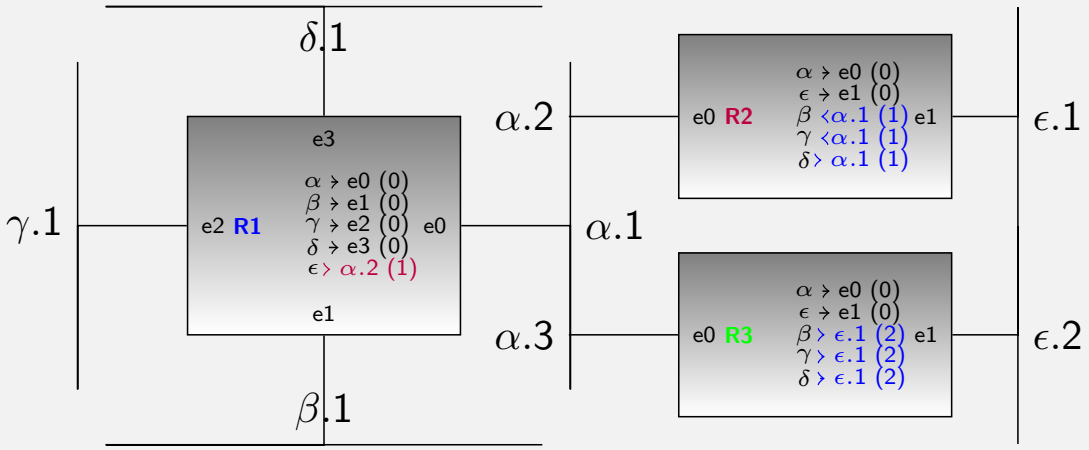
TELECOM Bretagne

Distant Vector Example (Add Route)

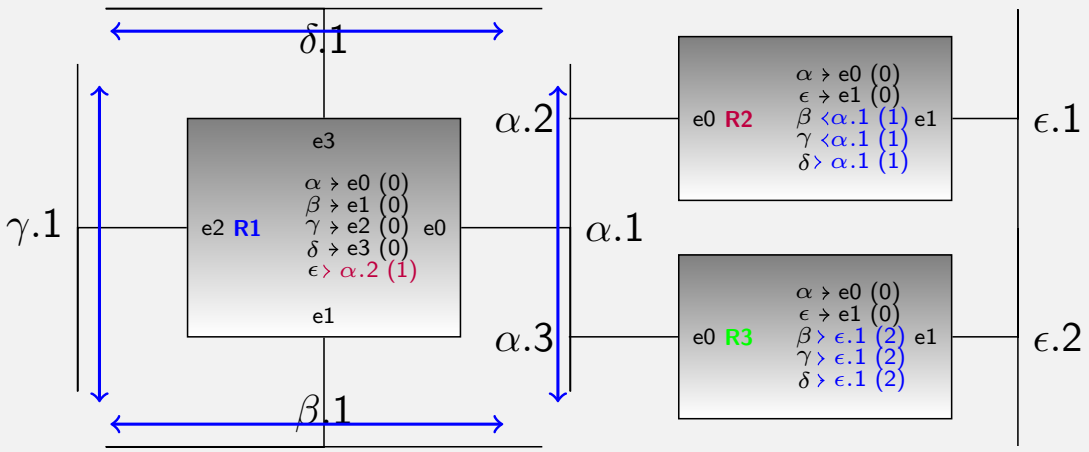


TELECOM Bretagne

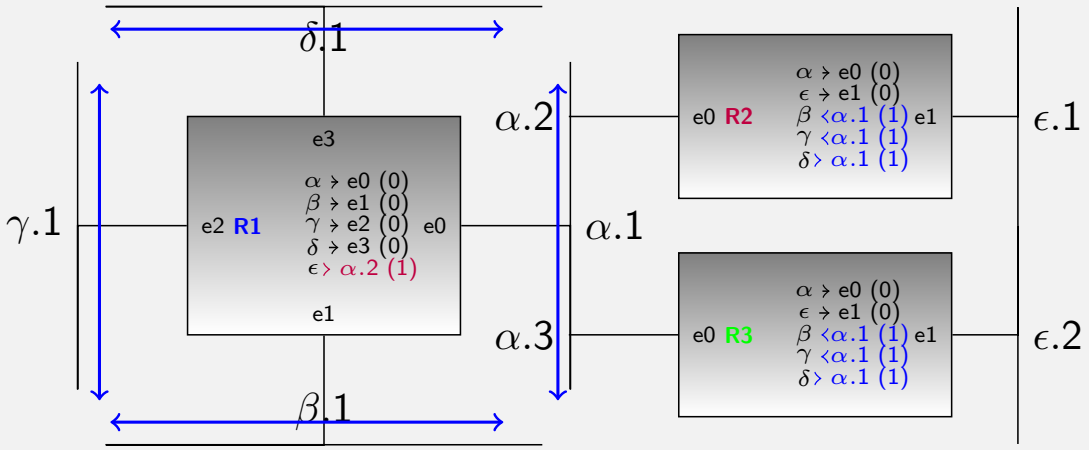
Distant Vector Example (Add Route)



Distant Vector Example (Add Route)



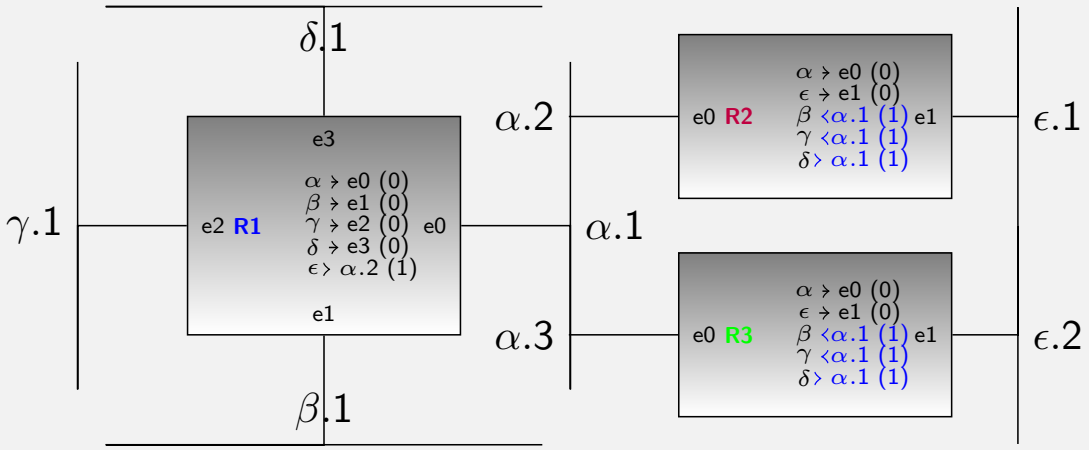
Distant Vector Example (Add Route)



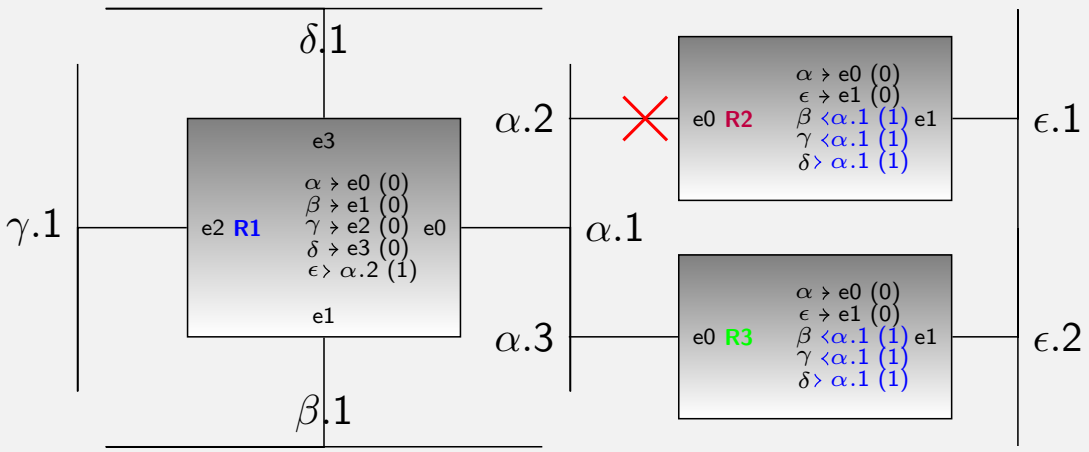
Distance Vector convergence

- Distant Vector (DV) implies periodic refreshing
  - To discover best path
  - To discover dead routers
- In RIP Routing Table are flooded every 30 seconds
- No specific message to remove an entry
  - If an entry is not seen during X flooding, this entry is removed
    - In RIP $X = 3$ (90 s)
  - Remove is similar to set cost to ∞
- DV cannot be used if Routing Table contains a lot of entries
  - For instance core network routers may contain 220 000 entries
- DV can only be used in very small networks
- DV have also bad convergence performances...

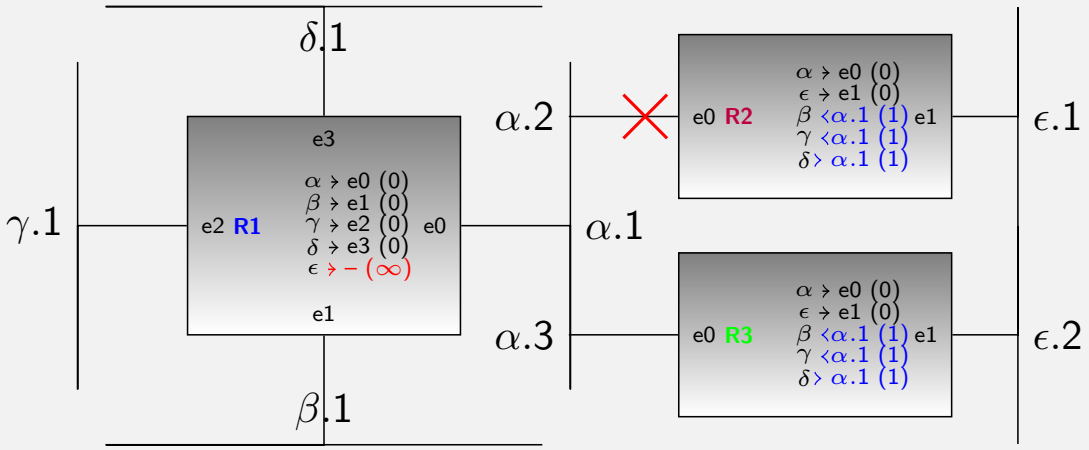
Distant Vector Example (Change Route)



Distant Vector Example (Change Route)

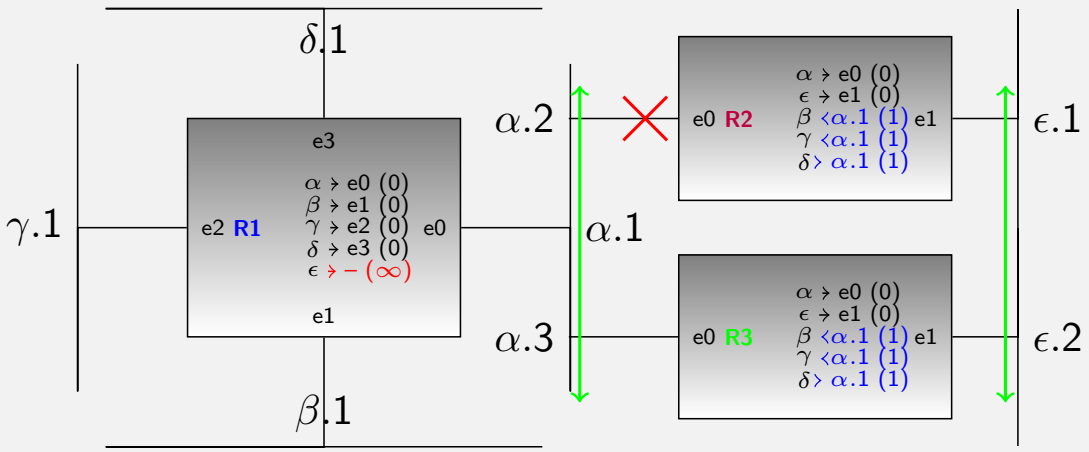


Distant Vector Example (Change Route)

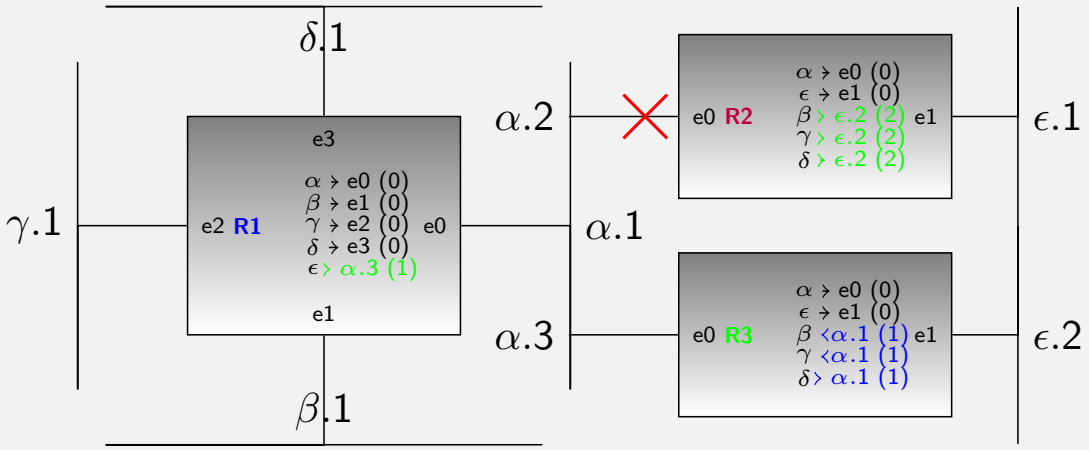


Distant Vector Example (Change Route)

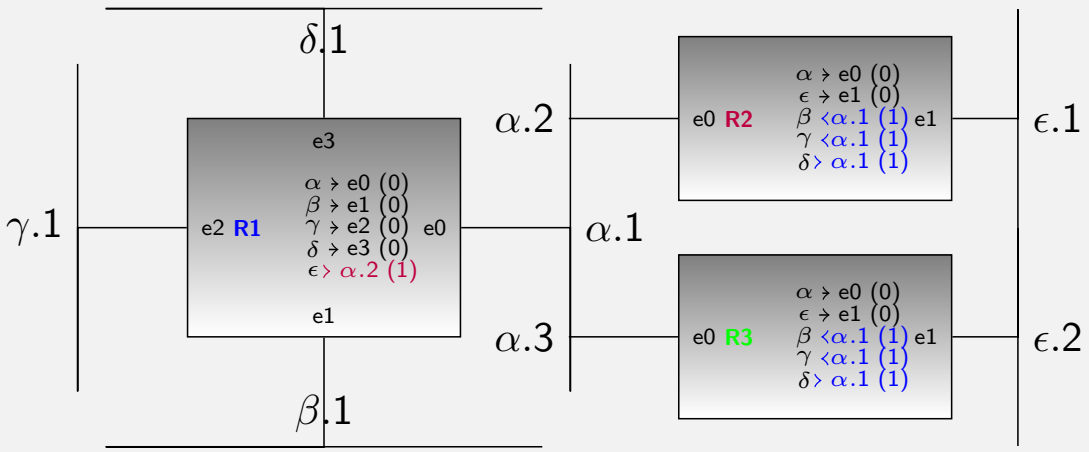
Distant Vector



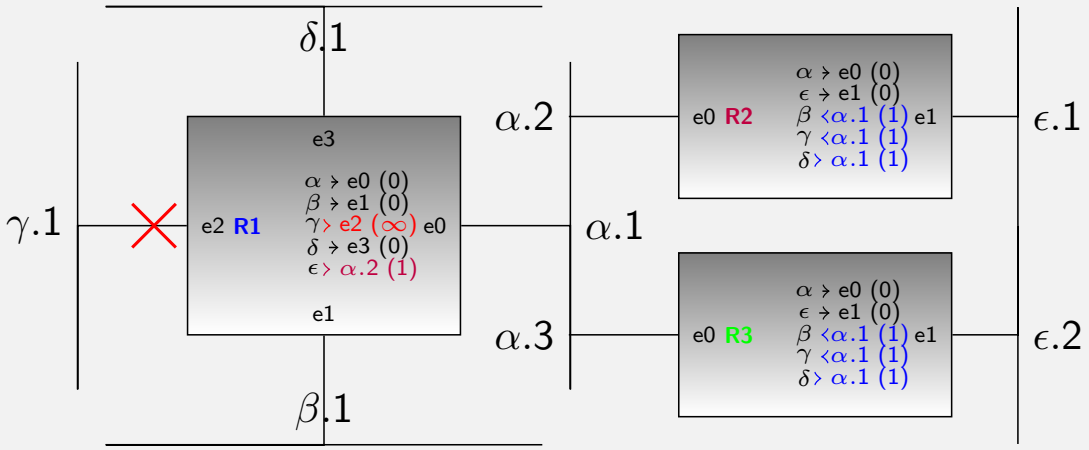
Distant Vector Example (Change Route)



Distant Vector Example (Bad Convergence)

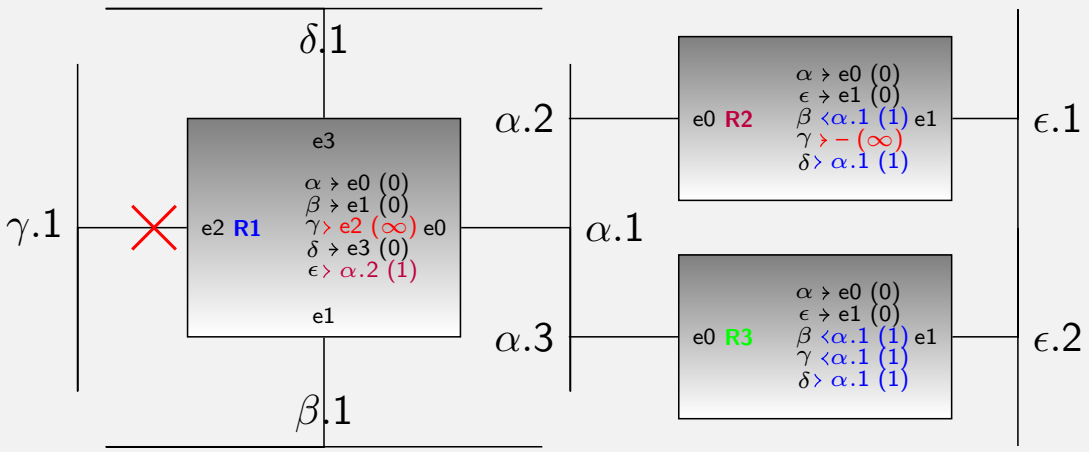


Distant Vector Example (Bad Convergence)

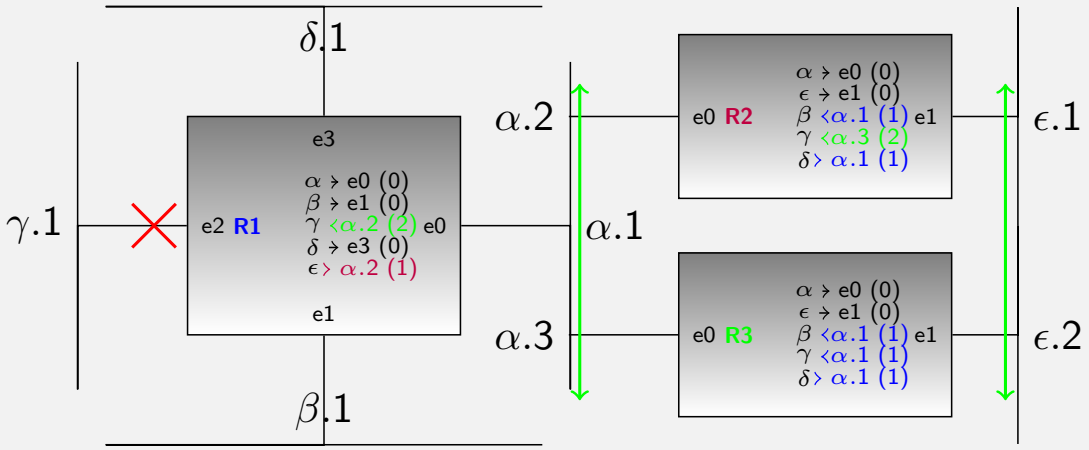


Distant Vector Example (Bad Convergence)

Distant Vector

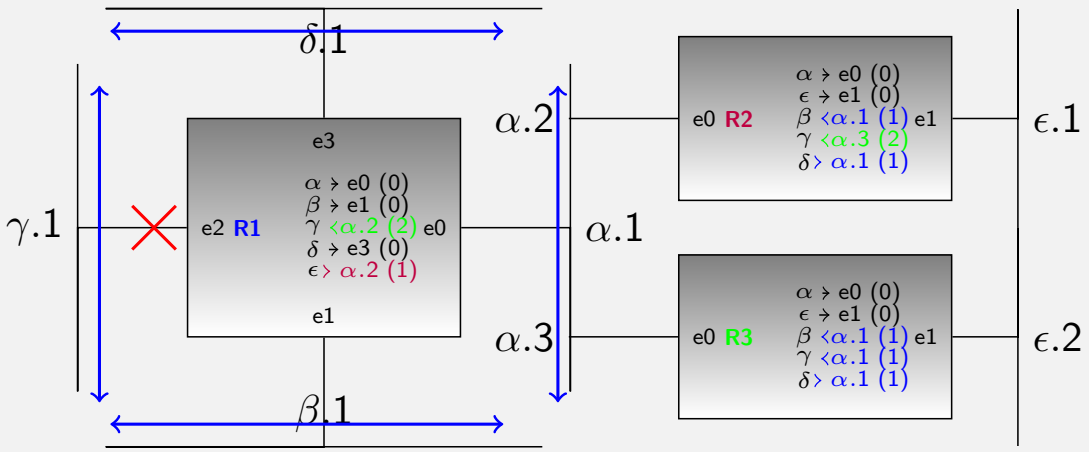


Distant Vector Example (Bad Convergence)

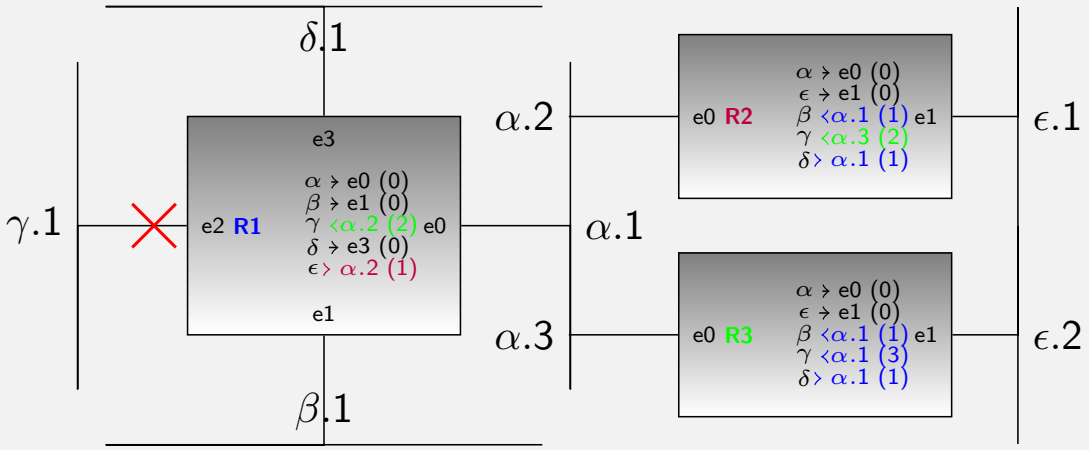


Distant Vector Example (Bad Convergence)

Distant Vector

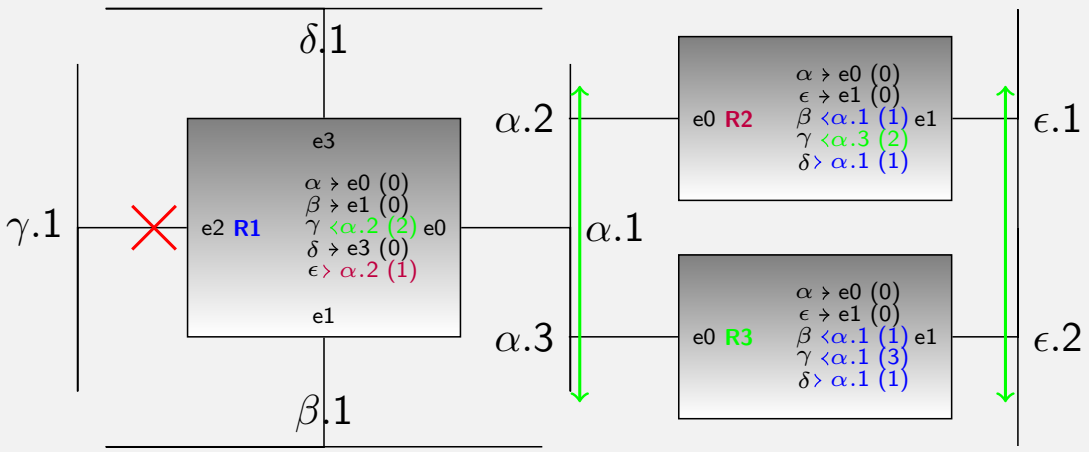


Distant Vector Example (Bad Convergence)

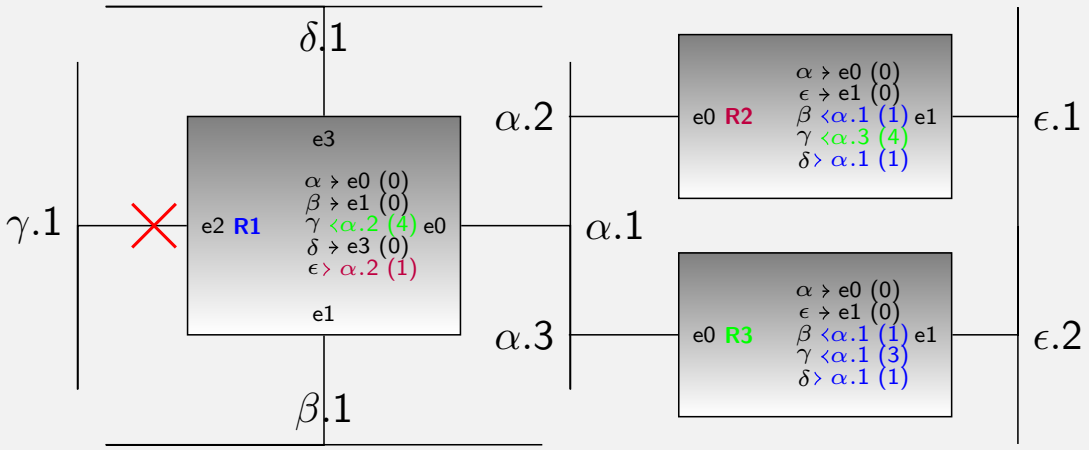


Distant Vector Example (Bad Convergence)

Distant Vector



Distant Vector Example (Bad Convergence)

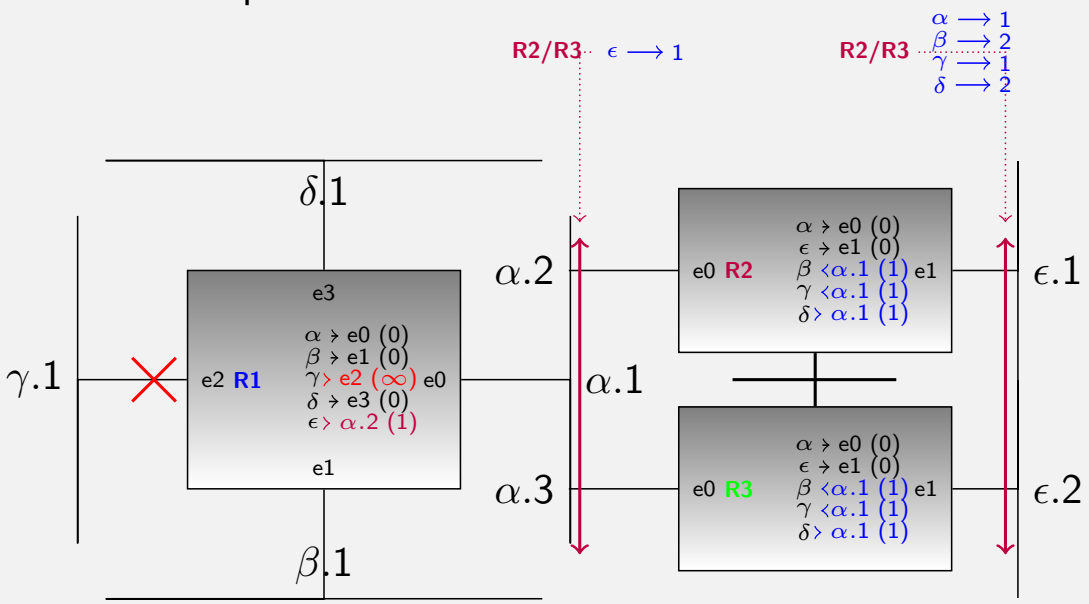


Counter Measures

- 3 ways to reduce this impact :
  - $16 = \infty$:
    - Not only 15 routers in the network but :
    - 15 routers between two links
  - Poisoning reverse:
    - When receiving a route marked ∞, mark it immediately ∞
    - Propagate rapidly route withdraw to avoid wrong routing messages
  - Split Horizon:
    - Do not announce through an interface, routes that use this interface.

Distant Vector Example (Split Horizon)

Split Horizon does not work here!



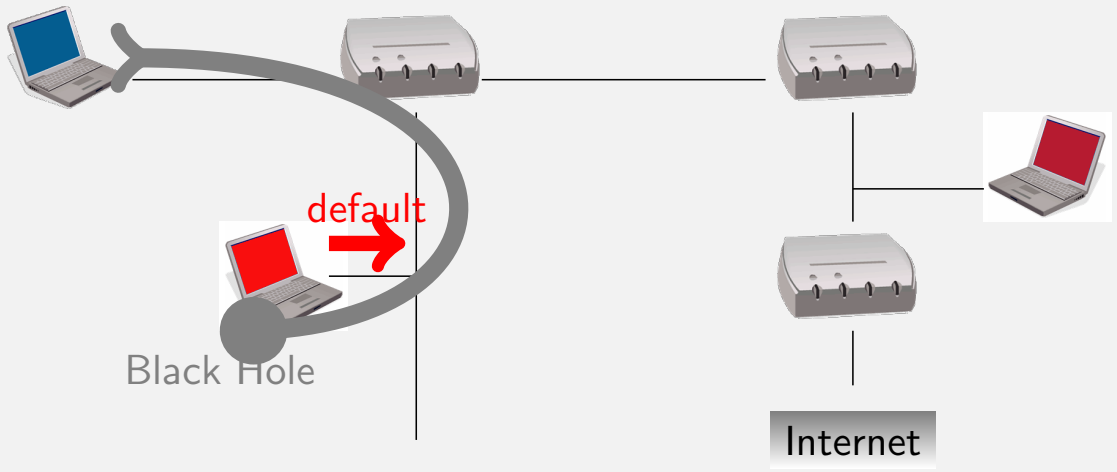
Distant Vector : Summary

- Distant Vector is very simple to understand, manage, implement.
- Performances are weak:
  - Shortsighted network vision: based on summary made by other routers
    - Like *concierges exchanging gossips*
  - Can lead to routing loops or long delays to reconnect parts of the network.
  - Periodically all routing table must be sent on the network:
    - to detect dead routers
    - to detect better paths
  - Generates network and processing load
- Distant Vector must be limited to small network auto-configuration
- RIPv2 implement this algorithm

Securing Routing Protocols

Distant Vector

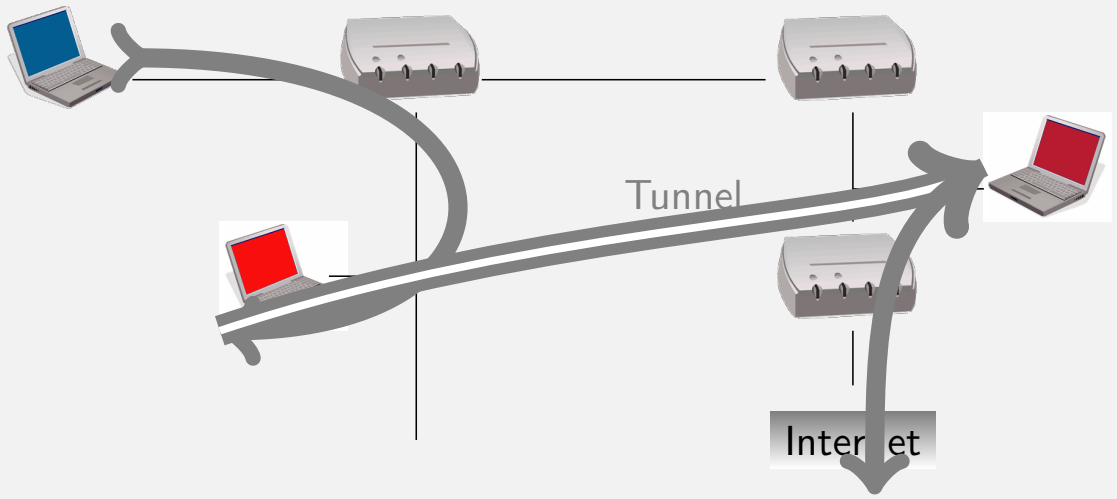
- Routing messages can be viewed as configuration messages:
  - Forging wrong routing messages can highjack some traffics
  - announcing default route may block communications



Securing Routing Protocols

Distant Vector

- Routing messages can be viewed as configuration messages:
  - Forging wrong routing messages can highjack some traffics
  - announcing default route may block communications



- include a secret password on each message
  - protect from configuration mistakes
  - weak security, since packets can be eavesdropped and secret learnt
- use cryptographic
  - RIPng uses built-in IPsec extentions
  - RIPv2 uses its own mechanisms
    - Originally based on MD5 algorithm (RFC 2082)
    - Obsoleted by RFC 4822, recommending SHA1 algorithm.

Link States

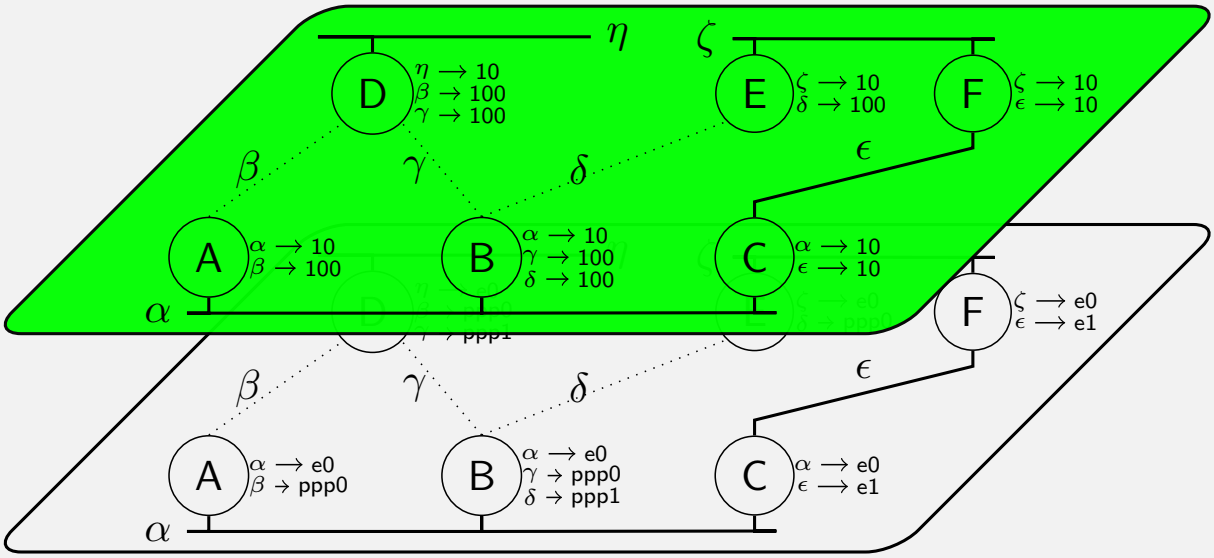
Algorithms

Link State Algorithm

- Distant Vector leads to a shortsighted view of network topology
  - each router processes routing messages
  - split horizon help for the first next hop
  - periodically flush all routing information
  - can be compared to caretaker exchanging gossip!
- Link State Protocols are divided in several states
  - learn network topology
  - exchange/flood local configuration
  - compute shortest path to a destination prefix
  - fill the FIB with Next Hop
  - less traffic, incremental updates
  - More a database synchronization algorithm than a routing protocol
- Two implementations
  - OSPF : v2 for IPv4 and v3 for IPv6
  - IS-IS : IP agnostic

Example: Database initialization

A cost is associated to each interface (for example related to the speed)



Example: Database state after synchronization

- After a flooding period (explained latter) all routers have exchanged their information
- Each router has all entries in its database

A $\alpha \longrightarrow 10$, $\beta \rightarrow 100$

B $\alpha \longrightarrow 10$, $\gamma \rightarrow 100$, $\delta \rightarrow 100$

C $\alpha \longrightarrow 10$, $\epsilon \longrightarrow 10$

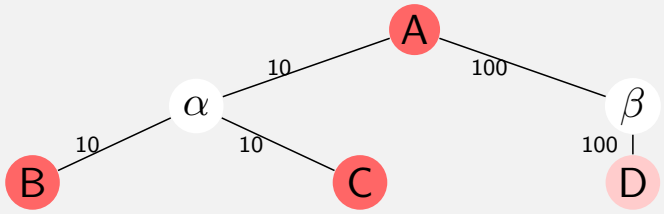
D $\eta \longrightarrow 10$, $\beta \rightarrow 100$, $\gamma \rightarrow 100$

E $\zeta \longrightarrow 10$, $\delta \rightarrow 100$

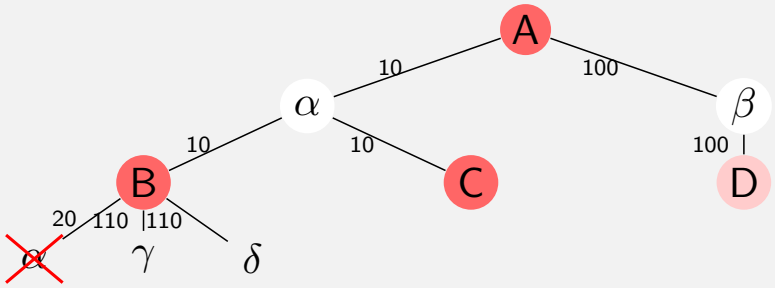
F $\zeta \longrightarrow 10$, $\epsilon \longrightarrow 10$

- Entries are called Link State
- These entries gives a full knowledge of the network topology
- can be viewed as a graph with nodes (router) and vertex (prefixes)
- find the shortest paths from the root (router doing computation) to prefixes
  - based on Dijkstra's algorithm
  - explore the graph always using the shortest path
  - complexity is $O(N^2)$

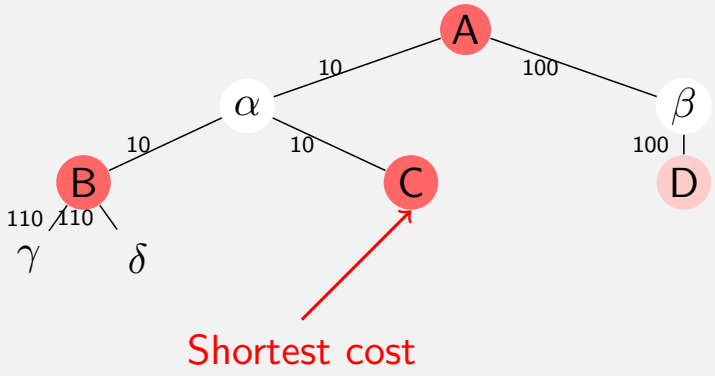
Example: Shortest Path First Algorithm



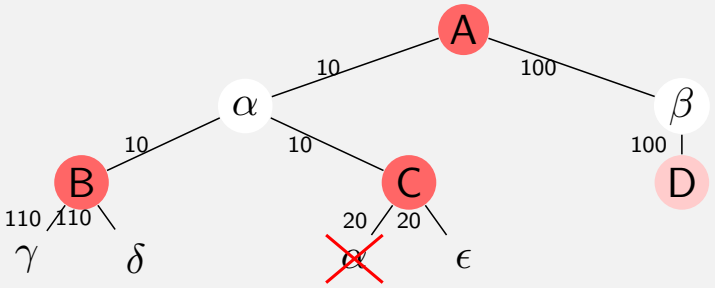
Example: Shortest Path First Algorithm



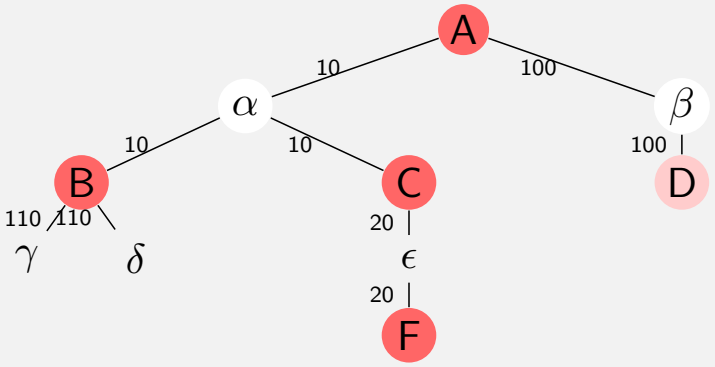
Example: Shortest Path First Algorithm



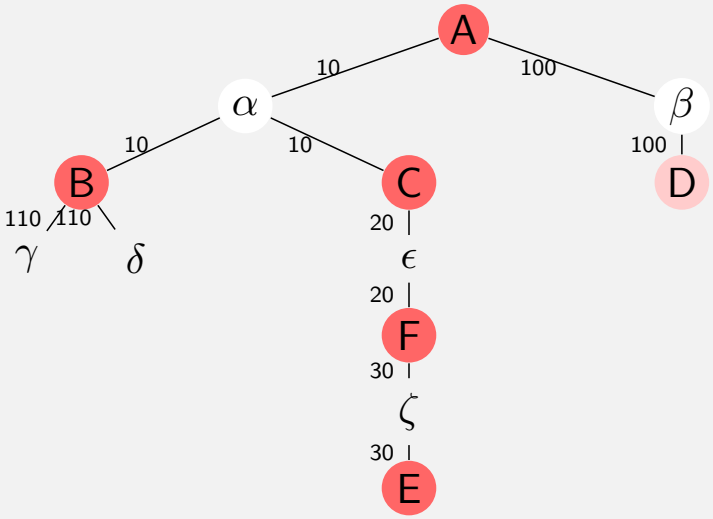
Example: Shortest Path First Algorithm



Example: Shortest Path First Algorithm

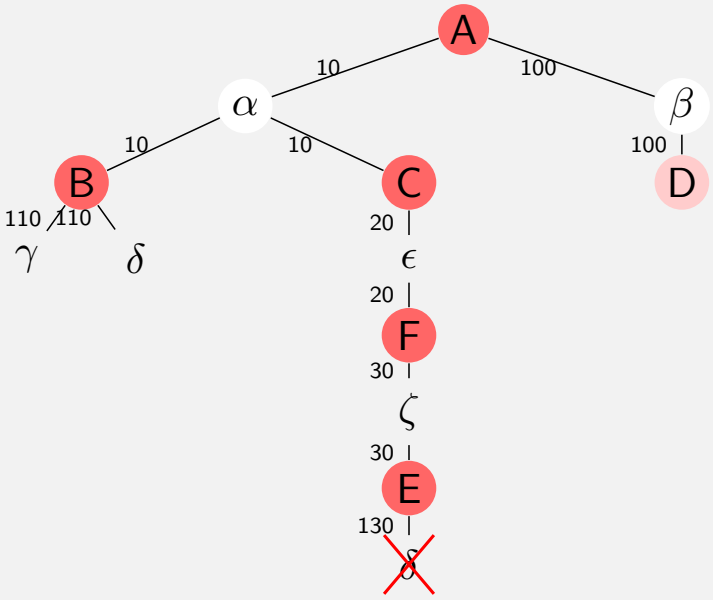


Example: Shortest Path First Algorithm



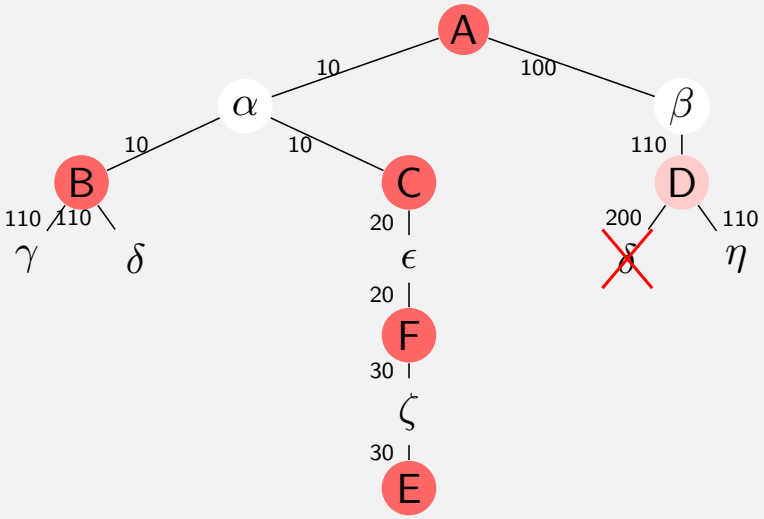
TELECOM Bretagne

Example: Shortest Path First Algorithm



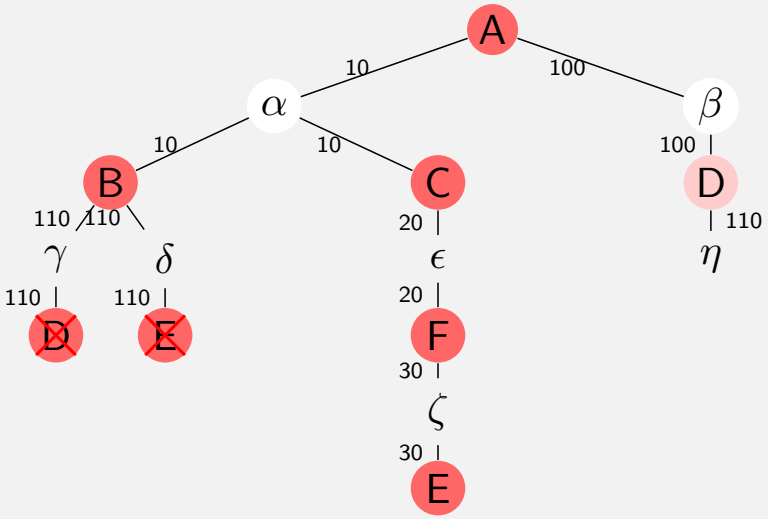
TELECOM Bretagne

Example: Shortest Path First Algorithm



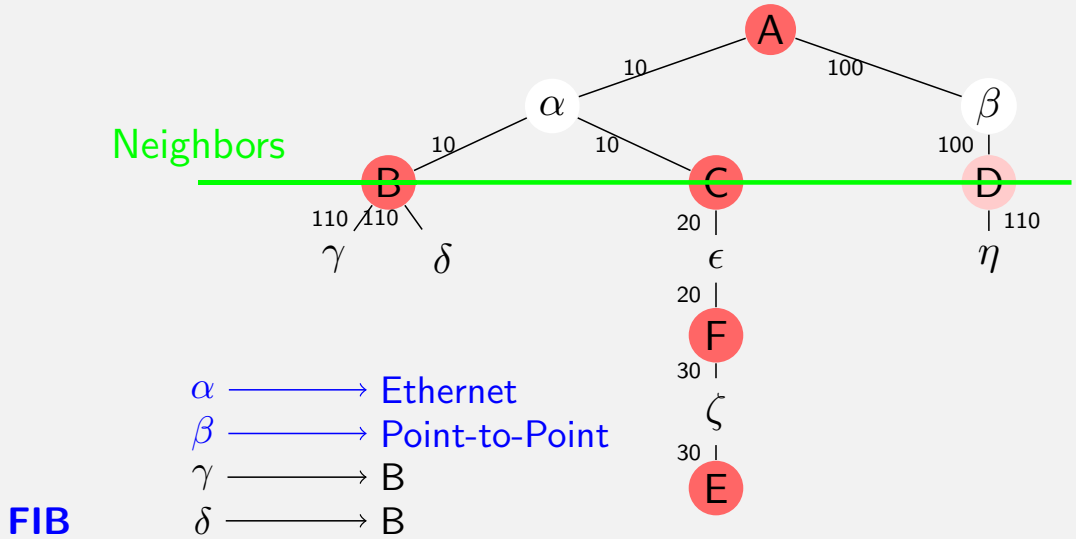
TELECOM Bretagne

Example: Shortest Path First Algorithm



TELECOM Bretagne

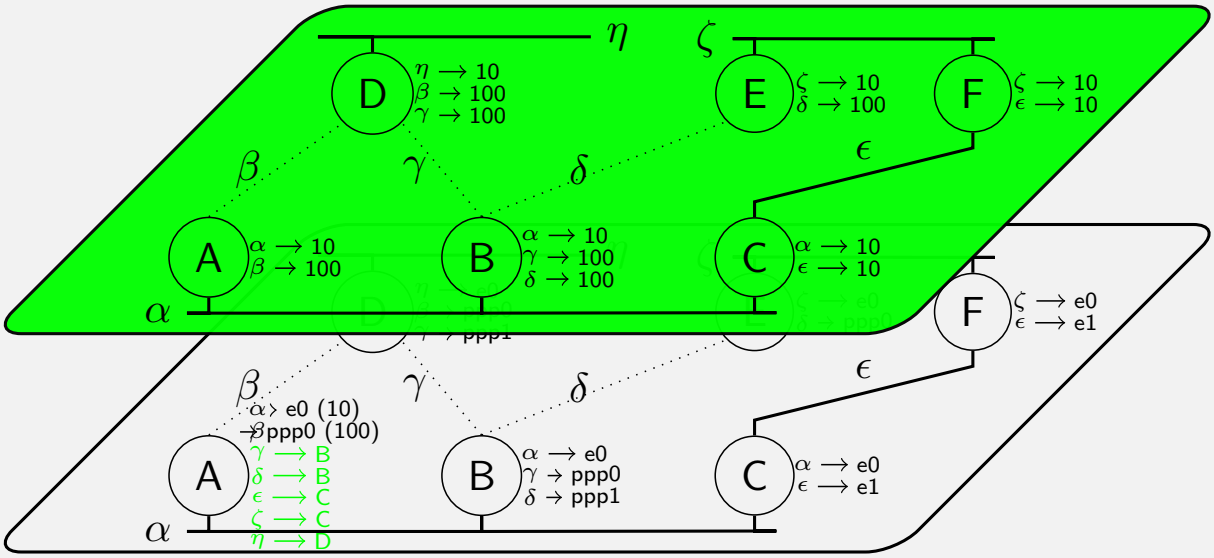
Example: Shortest Path First Algorithm



FIB

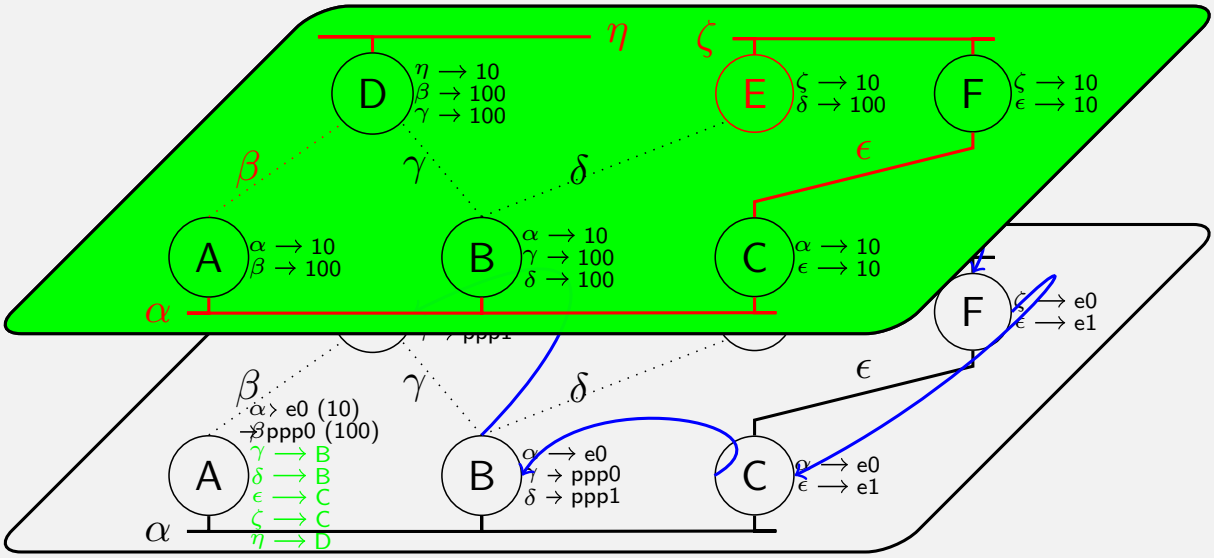
α	⟶	Ethernet
β	⟶	Point-to-Point
γ	⟶	B
δ	⟶	B
ϵ	⟶	C
ζ	⟶	C
η	⟶	D

Example: FIB entries



Warning: SPT is not forwarding path

Path is different, but cost is the same



TELECOM Bretagne

Link States

Areas

reducing SPF computation

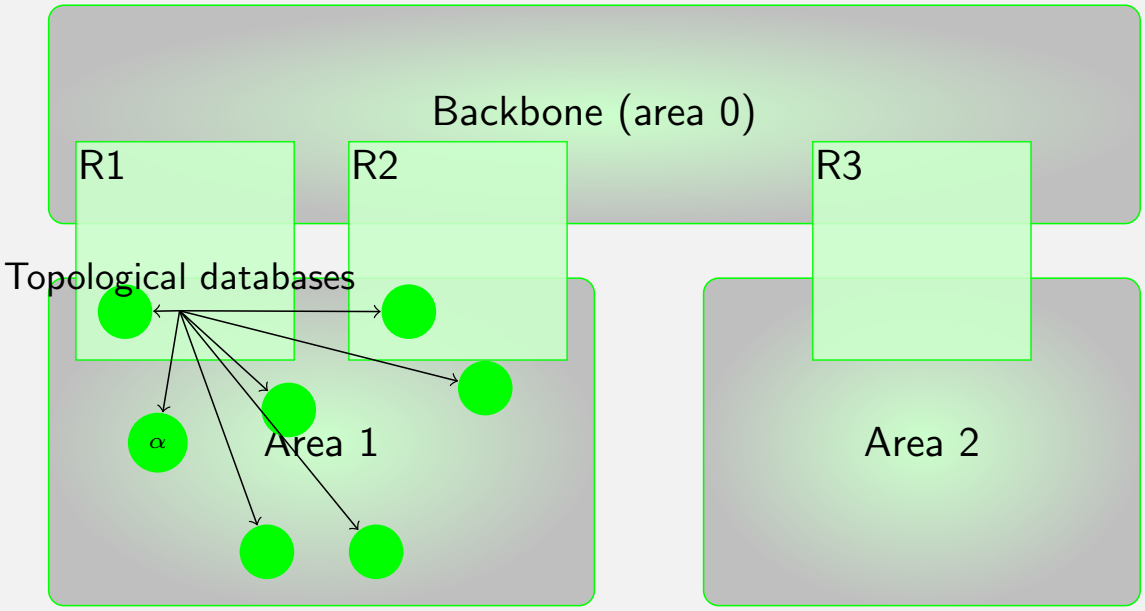
- SPF algorithm is complex ($O(N^2)$)
- every-time a router detect a topology change:
  - change is flooded to all routers
  - All router must rerun SPF algorithm

Definition

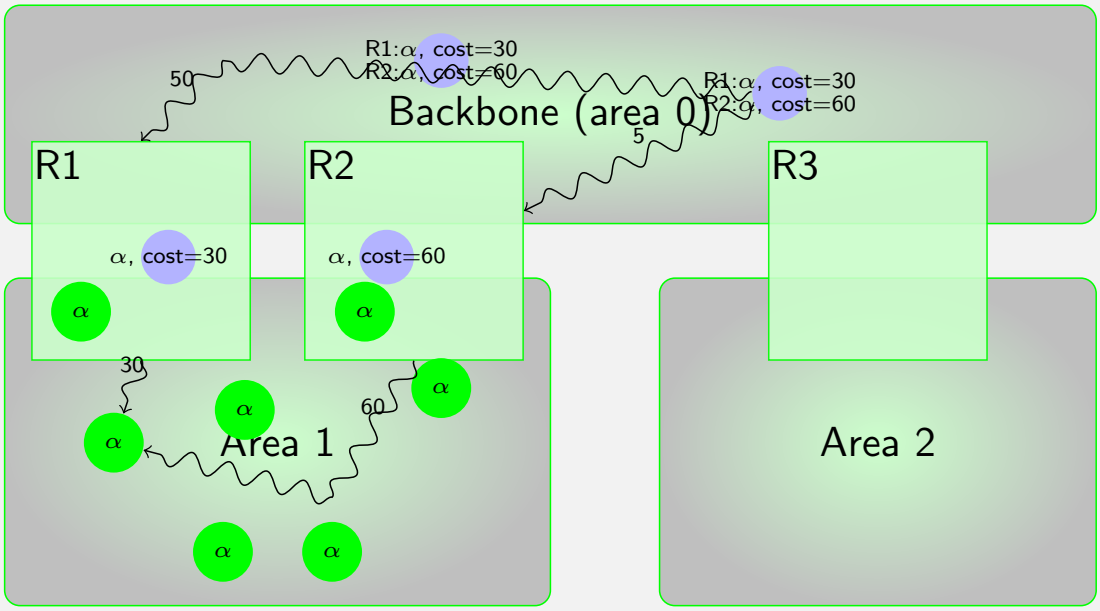
A network can be divided into several areas. All OSPF network contains

- A mandatory connexe backbone (Area 0)
- Optional areas, directly connected to the backbone through ABR (Area Border Router)
- ABR belongs to backbone and one (several) area(s)
- ABR sends summary (prefix+cost) of each area to the other ones.

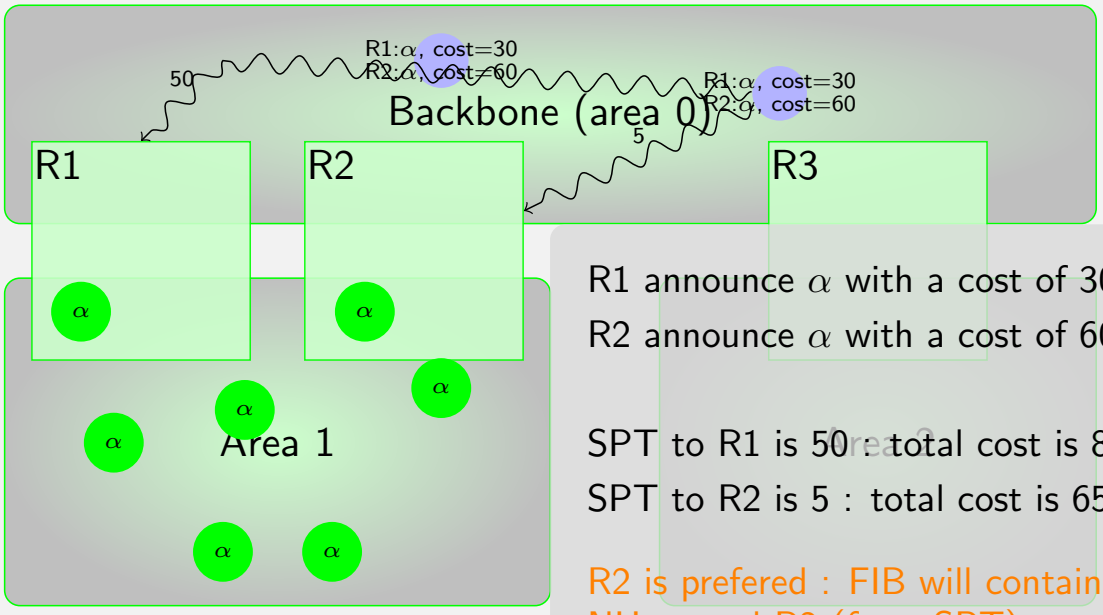
Division into Areas



Division into Areas



Division into Areas

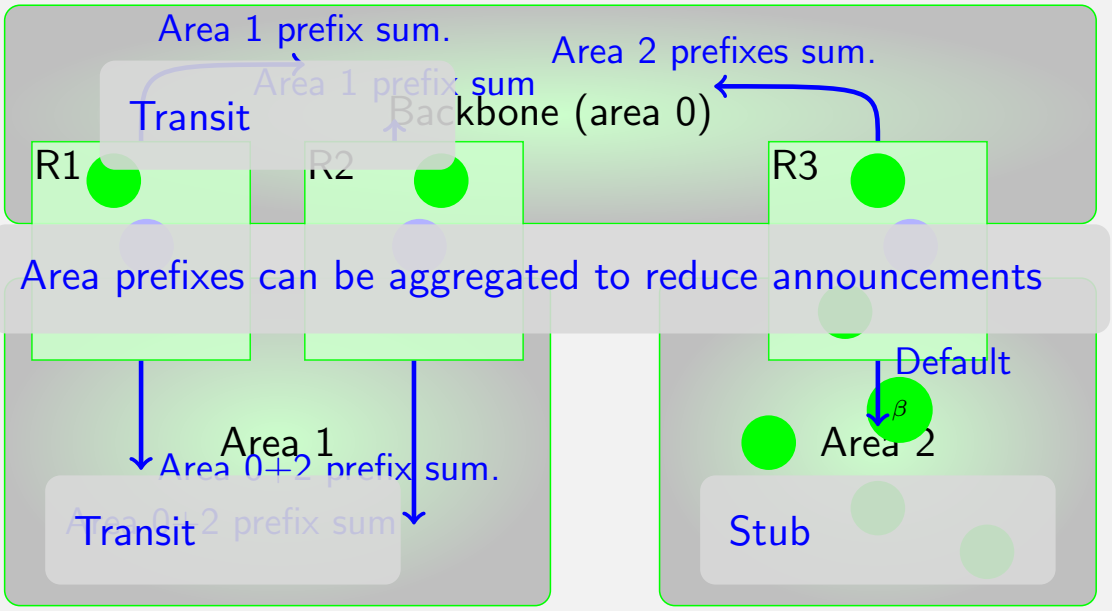


R1 announce α with a cost of 30
R2 announce α with a cost of 60

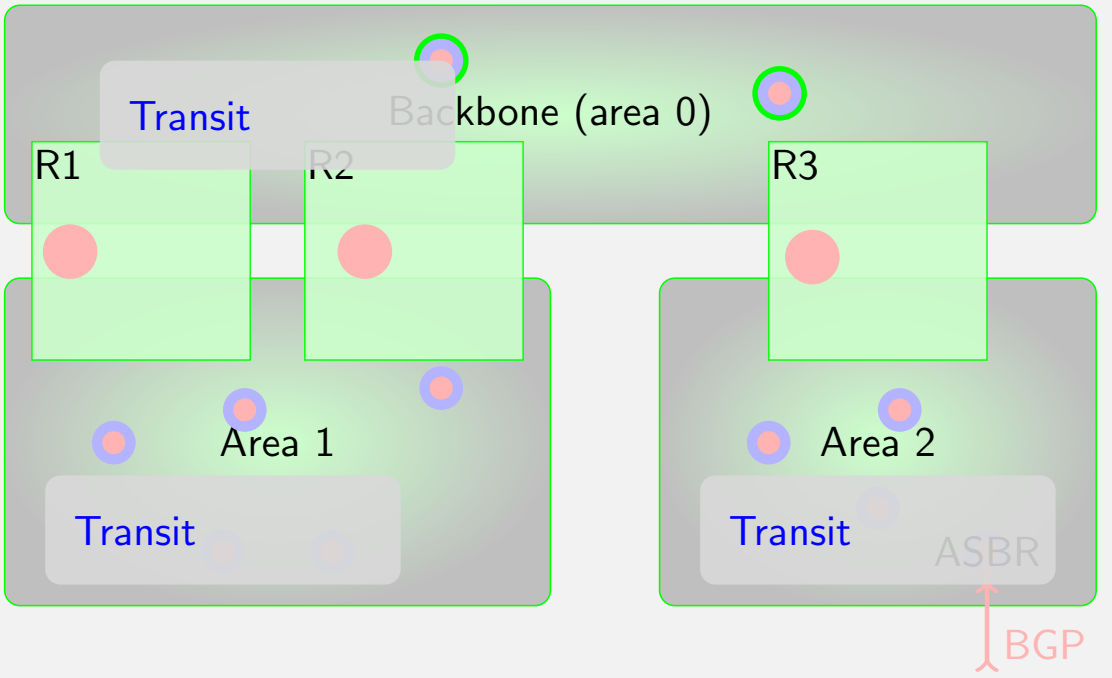
SPT to R1 is 50 : total cost is 80
SPT to R2 is 5 : total cost is 65

R2 is prefered : FIB will contain for α NH toward R2 (from SPT)

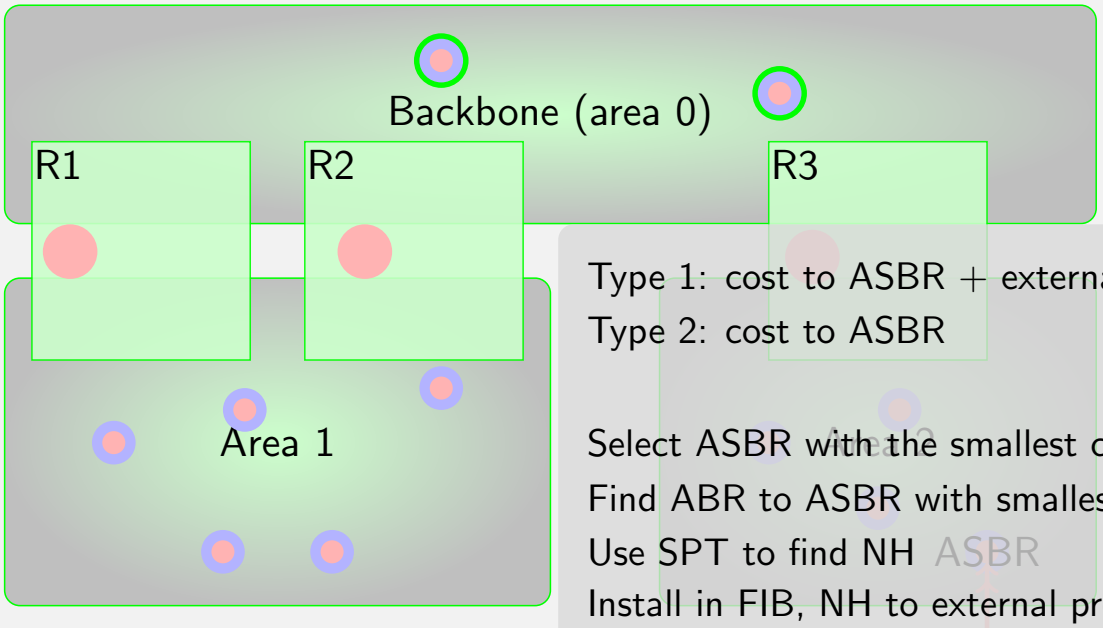
Division into Areas



Division into Areas



Division into Areas



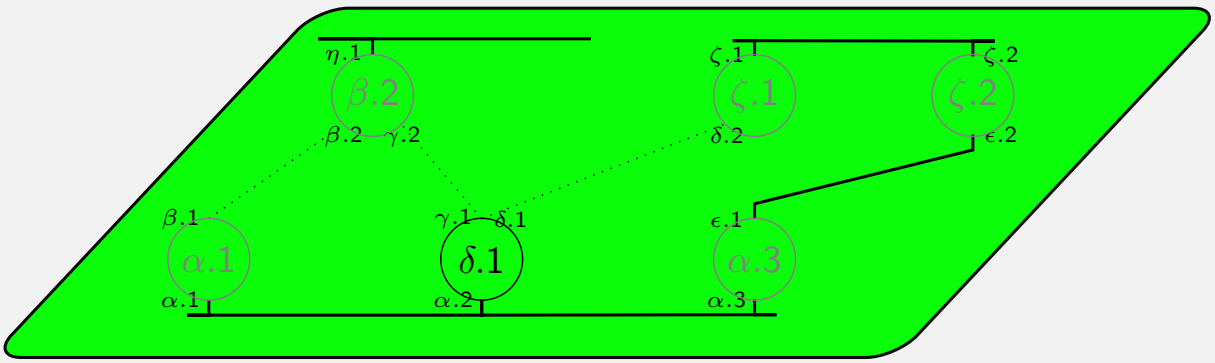
Type 1: cost to ASBR + external cost
Type 2: cost to ASBR

Select ASBR with the smallest cost
Find ABR to ASBR with smallest cost
Use SPT to find NH ASBR
Install in FIB, NH to external prefix

Database synchronization

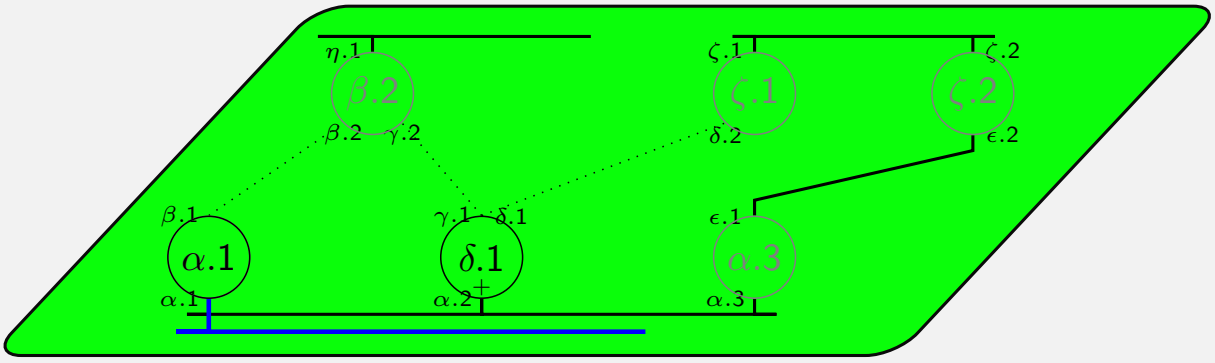
- OSPF maintain several kind of record (topological, summary, external prefixes, ASBR,...);
- All routers must share the same information
- Incremental update to reduce the bandwidth
  - compare to RIP, dumping database every 30 seconds
- exchange must be reliable in OSPF
- done is several steps and several protocols:
  - HELLO protocol: discover peers, elect a Designated Router
  - Database description: Describe information contained in each router's database
  - Database exchange: request and transfer information missing or more recent

Example: Flooding



Router ID: Generally one of the IP address
IP address unique => RID unique => order relation

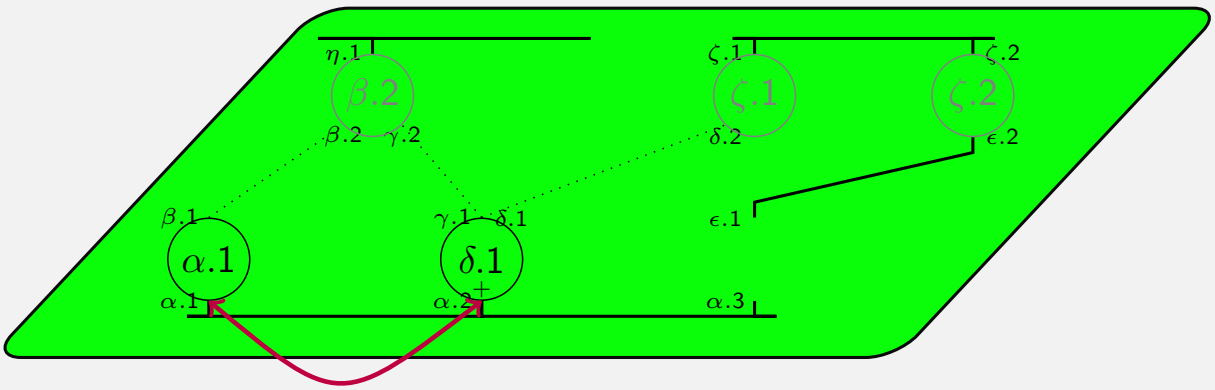
Example: Flooding



HELLO : DR = $\delta.1$ Nbrs $\{\delta.1\}$

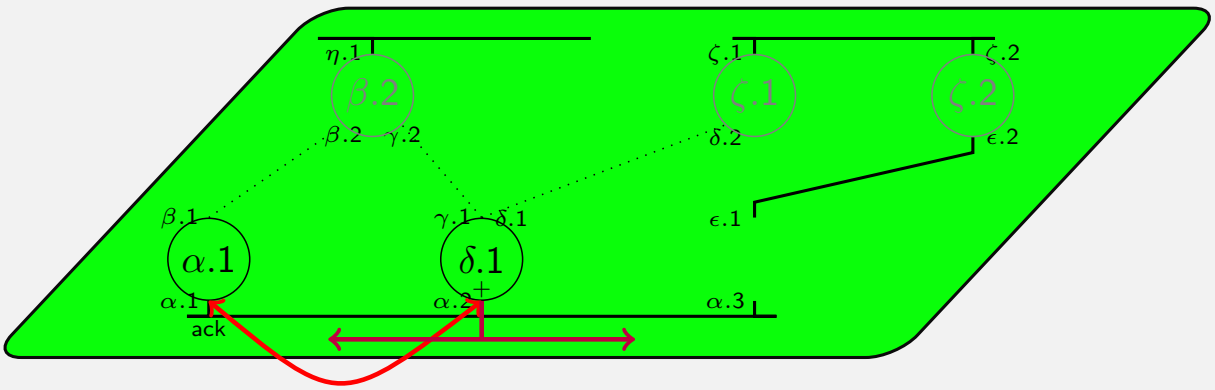
$\delta.1$ receives a Hello from $\overbrace{\alpha.1}$ with $\delta.1$ as neighbor
link between $\delta.1$ and $\alpha.1$ is bi directional (no risk for black hole)

Example: Flooding



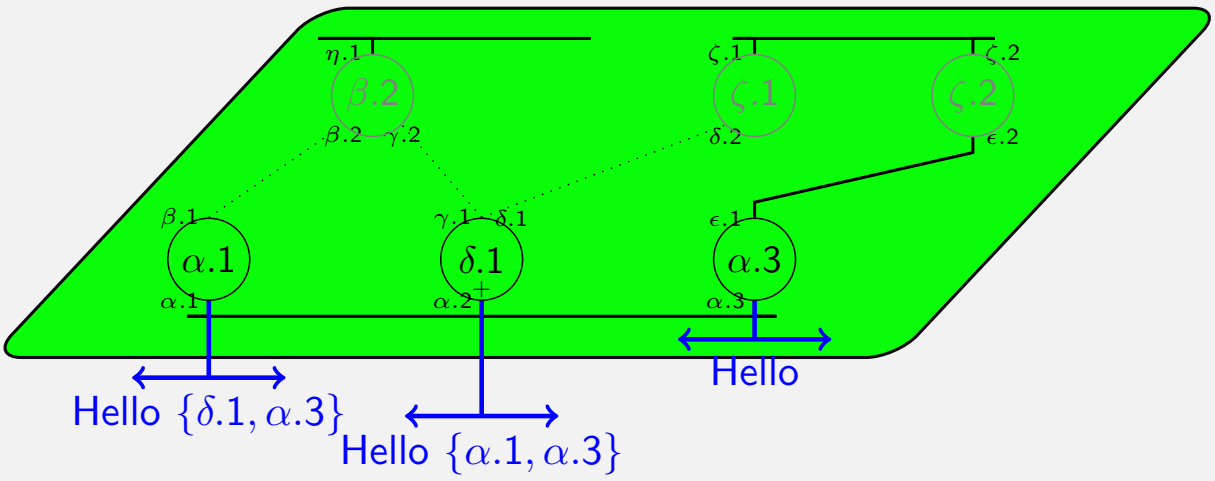
Database Description

Example: Flooding

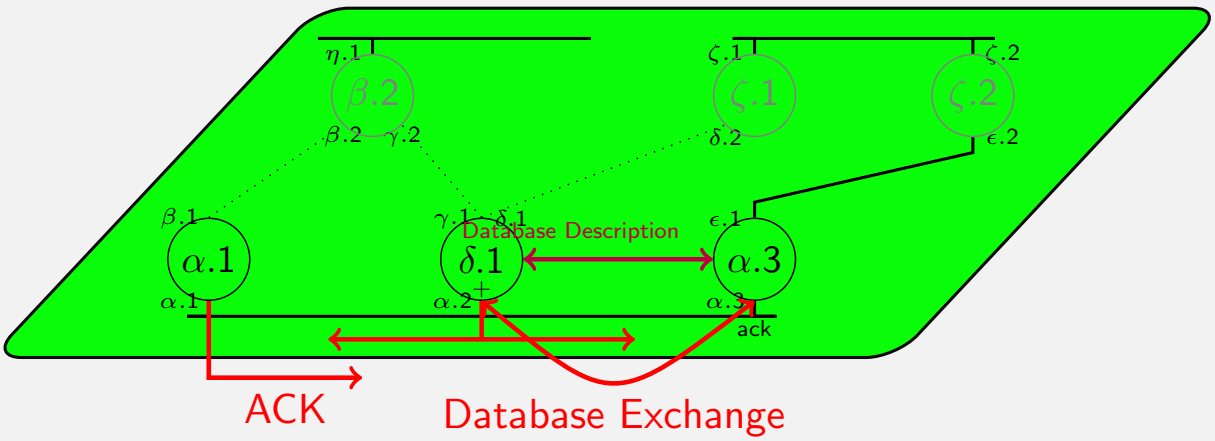


Database Exchange

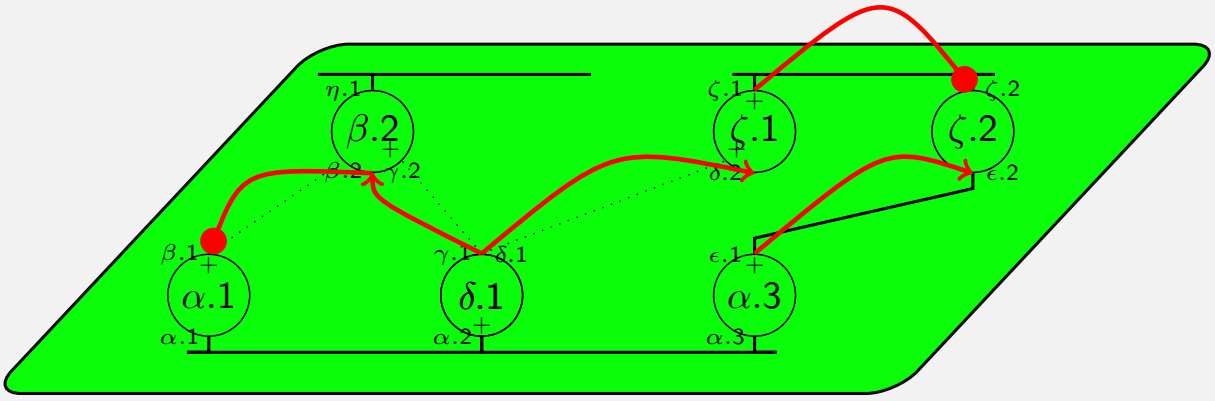
Example: Flooding



Example: Flooding



Example: Flooding



Not retransmitted since already stored in database

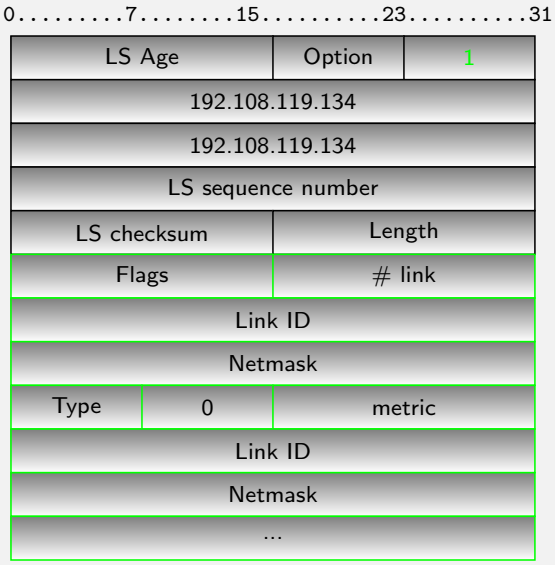
Link State Announcement Format

0.....7.....15.....23.....31

LS Age	Option	LS Type
LS Id		
Advertising Router		
LS sequence number		
LS checksum	Length	

- Age: after 40 min a new LS must be generated even if there is no change
- LS Type: 1 router, 2 NBMA link,3 and 4 : summary, 5: external routes,...
- LS Id: type 1 and 4: router ID, type 2, 3, 5: network
- Adv. router: type 1 = LS id others originating router
- Sequence number: form 0x80000001 to 0x7fffffff

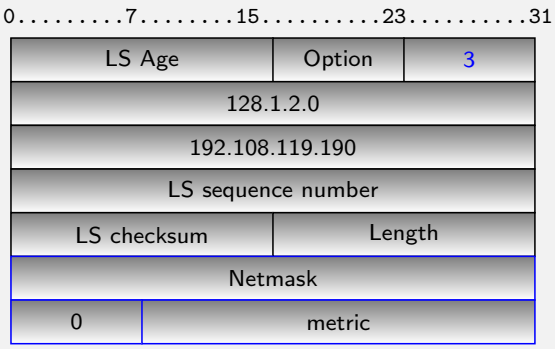
Link State Announcement Format



Type= 1 point-to-point, 2: transit, 3: stub, 4: virtual

- Age: after 40 min a new LS must be generated even if there is no change
- LS Type: 1 router, 2 NBMA link,3 and 4 : summary, 5: external routes,...
- LS Id: type 1 and 4: router ID, type 2, 3, 5: network
- Adv. router: type 1 = LS id others originating router
- Sequence number: form 0x80000001 to 0x7fffffff

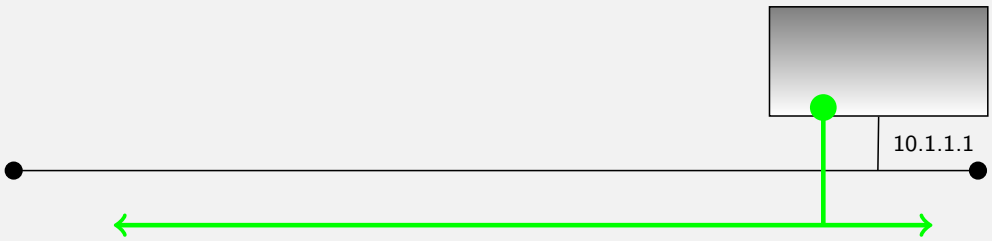
Link State Announcement Format



Link ID + Netmask gives prefix

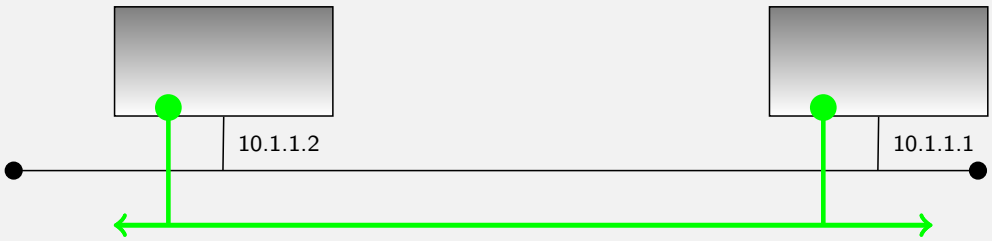
- Age: after 40 min a new LS must be generated even if there is no change
- LS Type: 1 router, 2 NBMA link,3 and 4 : summary, 5: external routes,...
- LS Id: type 1 and 4: router ID, type 2, 3, 5: network
- Adv. router: type 1 = LS id others originating router
- Sequence number: form 0x80000001 to 0x7fffffff

Example




```
10.1.1.1 > 224.0.0.5:  OSPFv2-hello 44:
           area 0.0.0.1 E mask 255.255.255.0 int 10 pri 5 dead 40 dr 10.1.1.1

           nbrs ;
```


Example




```
10.1.1.1 > 224.0.0.5:  OSPFv2-hello 48:

           area 0.0.0.1 E mask 255.255.255.0 int 10 pri 5 dead 40 dr 10.1.1.1 nbrs 10.1.1.2
```

- 10.1.1.1 is elected as designated router
- 10.1.1.2 appears in 10.1.1.1 neighbors: connection is bidirectional
- databases synchronization can start

Example

```
10.1.1.2 > 10.1.1.1:  OSPFv2-dd 32:  area 0.0.0.1 E I/M/MS S B

10.1.1.1 > 10.1.1.2:  OSPFv2-dd 32:  area 0.0.0.1 E I/M/MS S 1973
```

- 10.1.1.2 sequence number B - 10.1.1.1 sequence number 1973
- 10.1.1.2 is master (smallest sequence number)
- I: Initial packet, M: More packets, MS: Master

Example


```
{ S 80000002 age 5 rtr 10.1.1.2 }
```

```
{ E S 80000002 age 3:09 rtr 10.1.1.1 }
{ E S 80000001 age 2:49 sum 10.1.2.0 abr 10.1.1.1 }
{ E S 80000003 age 2:44 sum 10.1.100.0 abr 10.1.1.1
{ E S 80000001 age 2:59 abr 10.1.1.1 rtr 10.1.1.1 }
```

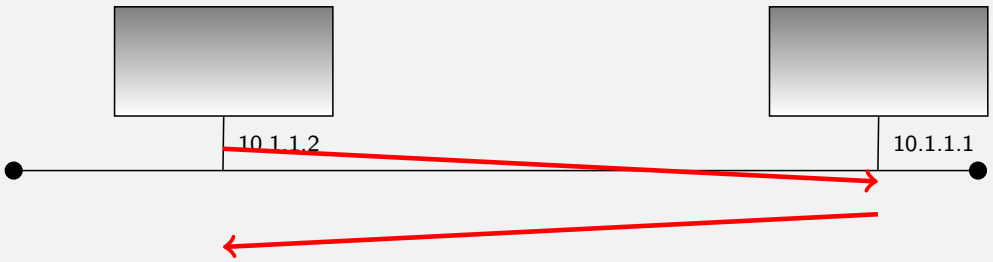
```
10.1.1.1 > 10.1.1.2:  OSPFv2-dd 112:  area 0.0.0.1 E M S B
          { E S 80000002 age 3:09 rtr 10.1.1.1 } TYPE 1
          { E S 80000001 age 2:49 sum 10.1.2.0 abr 10.1.1.1 } TYPE 3
          { E S 80000003 age 2:44 sum 10.1.100.0 abr 10.1.1.1 } TYPE 3

          { E S 80000001 age 2:59 abr 10.1.1.1 rtr 10.1.1.1 } TYPE 4
```

- 10.1.1.1 describes its database : TYPE 1 (one link), TYPE 3 (2 networks outside area 1), TYPE4 (one area border router)

Example


```
{ S 80000002 age 5 rtr 10.1.1.2 }
```

```
{ E S 80000002 age 3:09 rtr 10.1.1.1 }
{ E S 80000001 age 2:49 sum 10.1.2.0 abr 10.1.1.1 }
{ E S 80000003 age 2:44 sum 10.1.100.0 abr 10.1.1.1
{ E S 80000001 age 2:59 abr 10.1.1.1 rtr 10.1.1.1 }
```




```
10.1.1.2 > 10.1.1.1:  OSPFv2-dd 52:  area 0.0.0.1 E MS S C            { S 80000002 age 5 rtr
10.1.1.2 } TYPE 1
```

- 10.1.1.2 ack. previous message by incrementing Seq Number and give its database description : attached to one link.
- no more information (bit M not set)

TELECOM Bretagne

Example


```
{ S 80000002 age 5 rtr 10.1.1.2 }
```

```
{ E S 80000002 age 3:09 rtr 10.1.1.1 }
{ E S 80000001 age 2:49 sum 10.1.2.0 abr 10.1.1.1 }
{ E S 80000003 age 2:44 sum 10.1.100.0 abr 10.1.1.1
{ E S 80000001 age 2:59 abr 10.1.1.1 rtr 10.1.1.1 }
```

```
10.1.1.2 > 10.1.1.1:  OSPFv2-ls_req 72:  area 0.0.0.1
           { rtr 10.1.1.1 } { sum 10.1.2.0 abr 10.1.1.1 } { sum 10.1.100.0 abr 10.1.1.1 }
           { abr 10.1.1.1 rtr 10.1.1.1 }
10.1.1.1 > 10.1.1.2:  OSPFv2-ls_req 36:  area 0.0.0.1

           { rtr 10.1.1.2 }
```

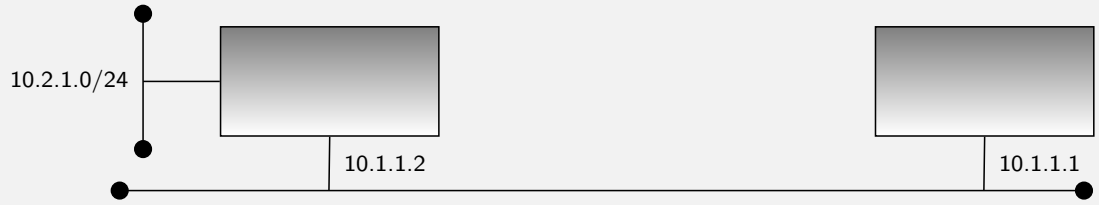
- Routers request missing or newest information

TELECOM Bretagne

Example

{ S 80000002 age 5 rtr 10.1.1.2 }

{ E S 80000002 age 3:09 rtr 10.1.1.1 }
{ E S 80000001 age 2:49 sum 10.1.2.0 abr 10.1.1.1 }
{ E S 80000003 age 2:44 sum 10.1.100.0 abr 10.1.1.1
{ E S 80000001 age 2:59 abr 10.1.1.1 rtr 10.1.1.1 }
{ S 80000002 age 5 rtr 10.1.1.2 }



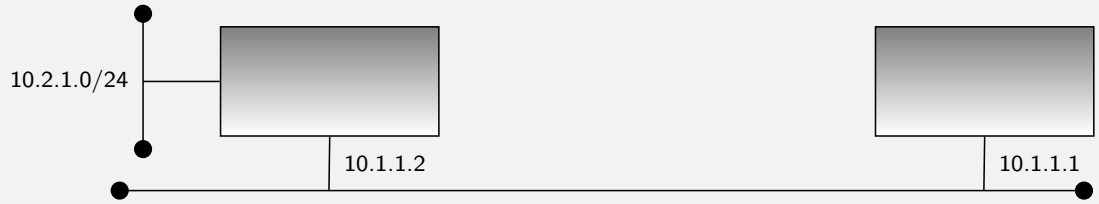

```
10.1.1.2 > 10.1.1.1:  OSPFv2-ls_upd 76:  area 0.0.0.1
           { S 80000002 age 6 rtr 10.1.1.2
                      { net 10.1.1.0 mask 255.255.255.0 tos 0 metric 1 }

                      { net 10.2.1.0 mask 255.255.255.0 tos 0 metric 1 } }
```

- Router 10.1.1.2 is connected to two links

Example

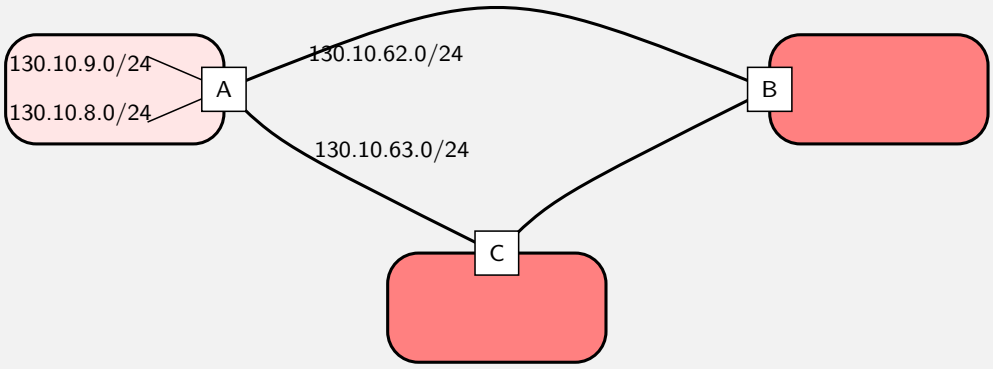
{ S 80000002 age 5 rtr 10.1.1.2 }
{ E S 80000002 age 3:09 rtr 10.1.1.1 }
{ E S 80000001 age 2:49 sum ... }
{ E S 80000003 age 2:44 sum ... }
{ E S 80000001 age 2:59 abr... }

{ E S 80000002 age 3:09 rtr 10.1.1.1 }
{ E S 80000001 age 2:49 sum 10.1.2.0 abr 10.1.1.1 }
{ E S 80000003 age 2:44 sum 10.1.100.0 abr 10.1.1.1
{ E S 80000001 age 2:59 abr 10.1.1.1 rtr 10.1.1.1 }
{ S 80000002 age 5 rtr 10.1.1.2 }



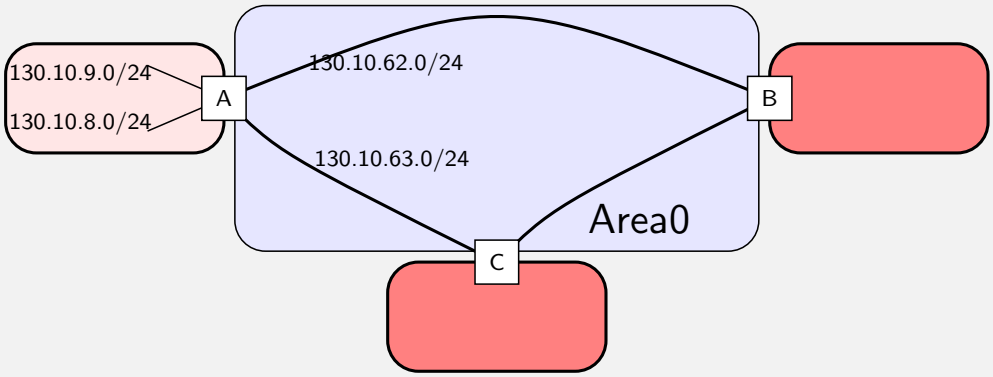

```
10.1.1.1 > 10.1.1.2:  OSPFv2-ls_upd 148:  area 0.0.0.1
           { E S 80000002 age 3:10 rtr 10.1.1.1 B
                      { net 10.1.1.0 mask 255.255.255.0 tos 0 metric 10 } }
           { E S 80000001 age 2:50 sum 10.1.2.0 abr 10.1.1.1 mask 255.255.255.0 tos 0 metric 20
}
           { E S 80000003 age 2:45 sum 10.1.100.0 abr 10.1.1.1 mask 255.255.255.0 tos 0 metric
10 }

           { E S 80000001 age 3:01 abr 10.1.1.1 rtr 10.1.1.1 tos 0 metric 16777215 }
```

Case study: RIP to OSPF




```
interface serial 0
        ip address 130.10.62.1 255.255.255.0
interface serial 1
        ip address 130.10.63.1 255.255.255.0
interface ethernet 0
        ip address 130.10.8.1 255.255.255.0
interface ethernet 1
        ip address 130.10.9.1 255.255.255.0
router rip
        network 130.10.0.0
```

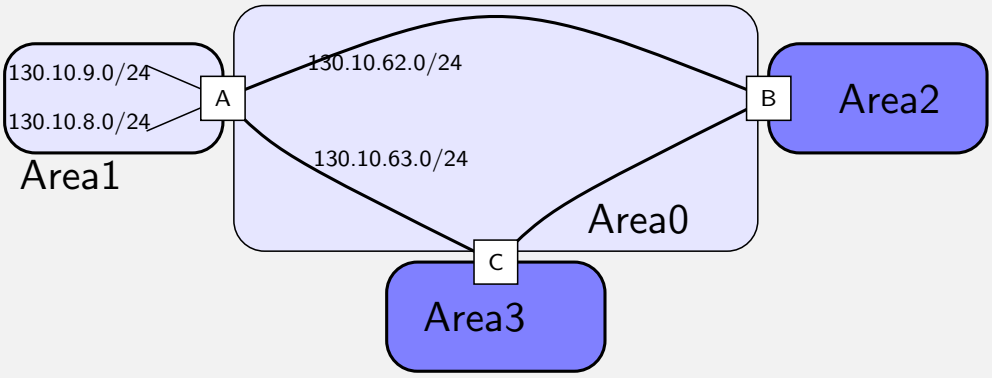

Case study: RIP to OSPF




```
! same interface definition
router rip
   default-metric 10
   network 130.10.0.0
   passive-interface serial 0
   passive-interface serial 1
   redistribute ospf 109 match internal external 1
external 2
```

```
router ospf 109
   network 130.10.62.0 0.0.0.255 area 0
   network 130.10.63.0 0.0.0.255 area 0
   redistribute rip subnets
   distribute-list 11 out rip
!
access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.2
```

Case study: RIP to OSPF




```
interface serial 0
   ip address 130.10.62.1 255.255.255.248
interface serial 1
   ip address 130.10.63.1 255.255.255.248
interface ethernet 0
   ip address 130.10.8.1 255.255.255.0
   ip irdp
```

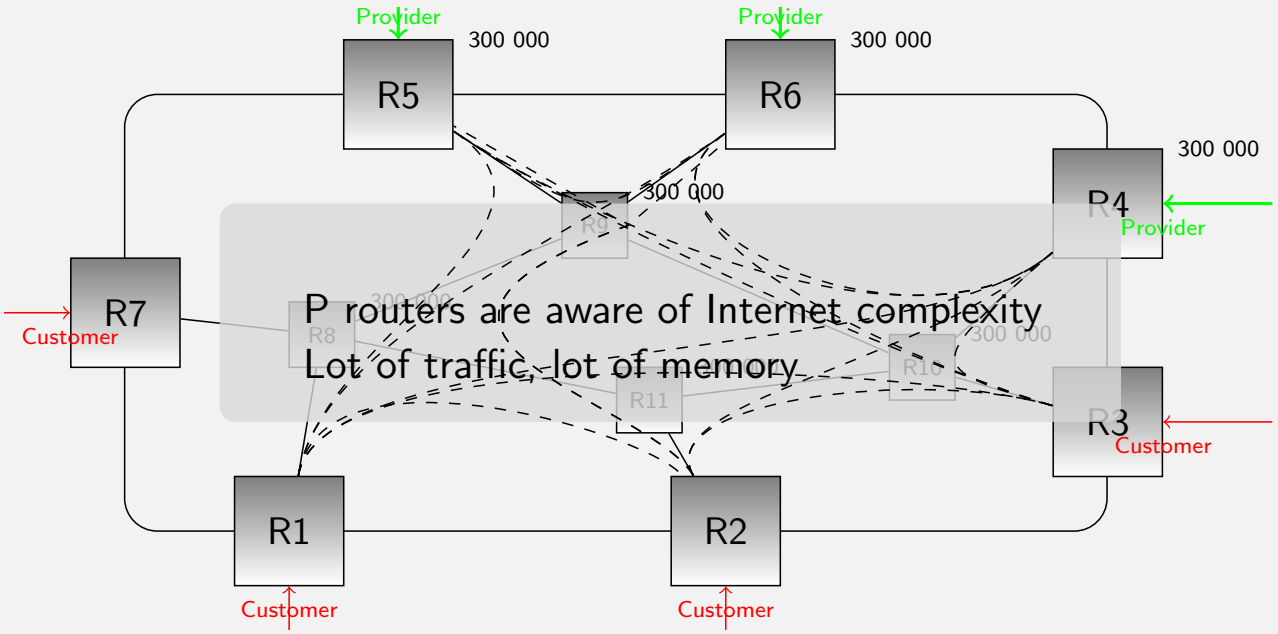
```
interface ethernet 1
   ip address 130.10.9.1 255.255.255.0
   ip irdp
router ospf 109
   network 130.10.62.0 0.0.0.255 area 0
   network 130.10.63.0 0.0.0.255 area 0
   network 130.10.8.0 0.0.7.255 area 1
   area 1 range 130.10.8.0 255.255.248.0
```


Multi Protocol Label Switching

IGP has some drawbacks:

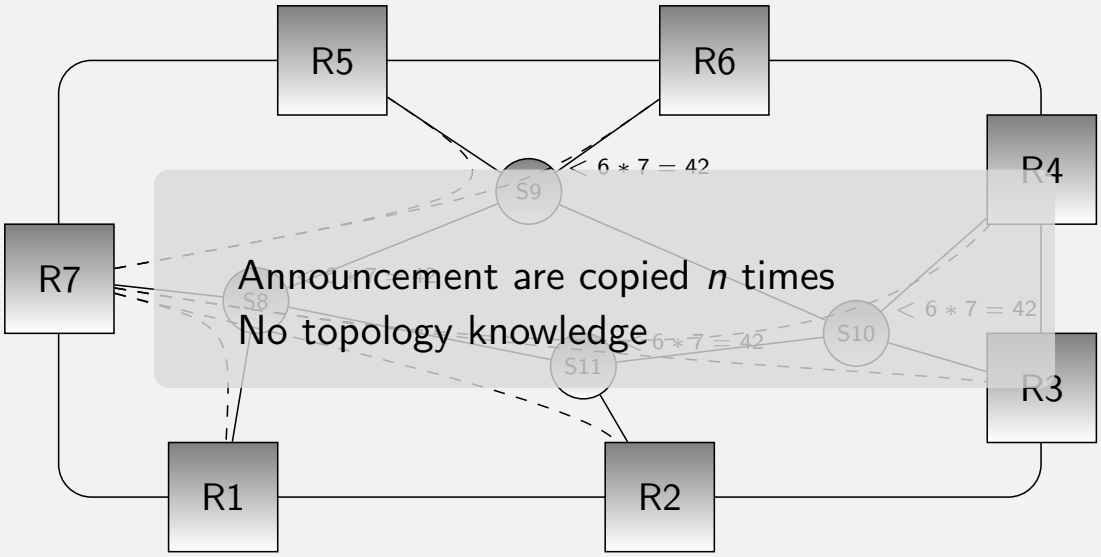
- Does not isolate AS form outside:
  - Lack of scalability
  - Same protocol inside and outside the AS
- Few control on routing decision
- No global vision on end-to end path inside the AS:
  - Next Hop dictatorship
- No traffic engineering : path selection, resource reservation, backup routes,...

Example: Layer 2 network (eg: ATM)



P routers are aware of Internet complexity
Lot of traffic, lot of memory

Example: Layer 2 network (eg: ATM)



Announcement are copied n times
No topology knowledge

L2 switching

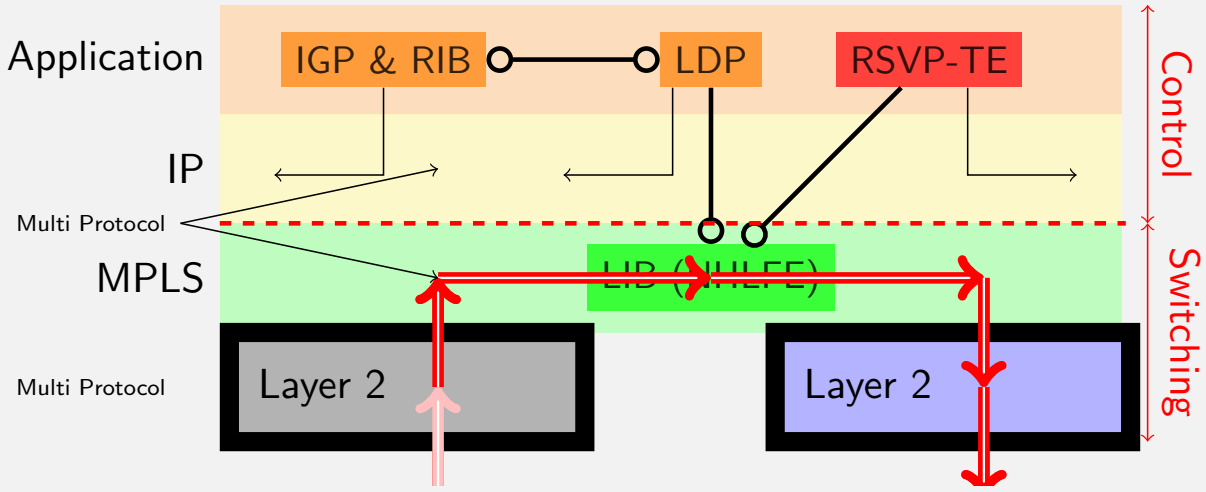
- Each PE router open a VP with other PE router
- VPs are viewed as direct links
- Switches do not seen routing tables
- Only PE has to memorize all DFZ (Default-Free Zone) routes
- VP can be set manually
  - better use or resources (load balancing)
- BGP and IGP exchanges between routers inside the domain
  - same information is transfered several times on the same physical link
- Routers are not aware of physical topology

MPLS: Multi Protocol Label Switching

- Goal:
  - Use routing protocols as a *signaling plane*
  - Switch the data plane
    - Use of virtual path called Label Switched Path
    - A switching table: Next Hop Label Forwarding Element
  - Offer more flexibility for traffc engineering
  - Routers are called Label Switch Routers
- LDP Label Distribution Protocol convert IGP routing into LSP

- RSVP-TE can be used to by-pass IGP routing and assign bandwidth

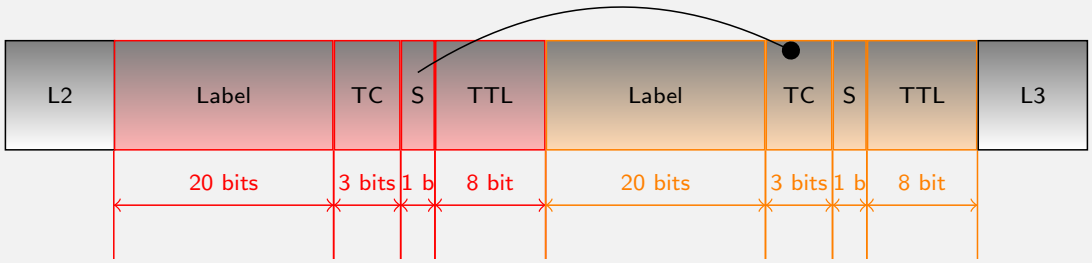
Simplified architecture of a LSR



MPLS: Multi Protocol...

- Layer 2
  - Any layer 2 with VPs like ATM, Frame Relay
  - Point-to-Point Networks
  - Ethernet
  - Extension for Optical Networks (GMPLS)
- Layer 3
  - Forwarding is under Layer 3
  - IPv4 (with public or private addresses), IPv6
  - Ethernet / Bridging
- Done by hardware: more efficient than an IP tunnel.

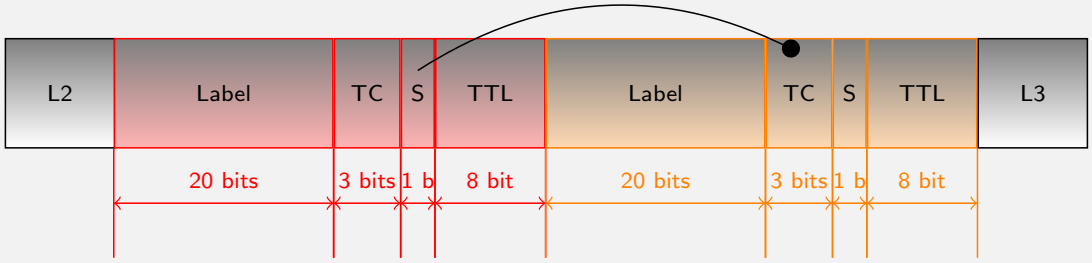
MPLS: ...Label...



- ATM and Frame relay: Label in copied in the VPI/VCI or DLCI
- Ethernet: Ethertype 8847 (Unicast), 8848 (Multicast)
- PPP: protocol : 281 (Unicast), 282 (Multicast)
- Label can be :
  - per platform: unique for each LSR
  - per interface: unique for one interface (may be reused elsewhere) :
    - Default behavior

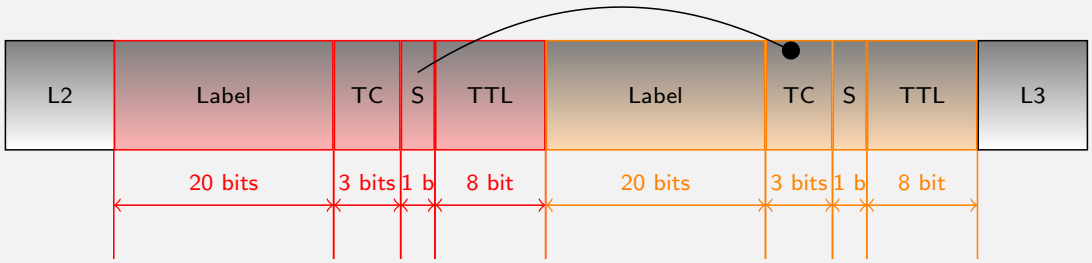
MPLS: ...Label...

Link States ▶ Areas



- Label can be :
  - per platform: unique for each LSR
    - Not necessary to remember interface
  - per interface: unique for one interface
    - must associate value and interface in the LIB
  - Special labels:
    - 0: IPv4 Explicit NULL = POP and forwarding
    - 1: Router Alert
    - 2: IPv6 Explicit NULL = POP and forwarding
    - 3: Implicit NULL = POP
    - 14: OAM Alert [RFC3429]

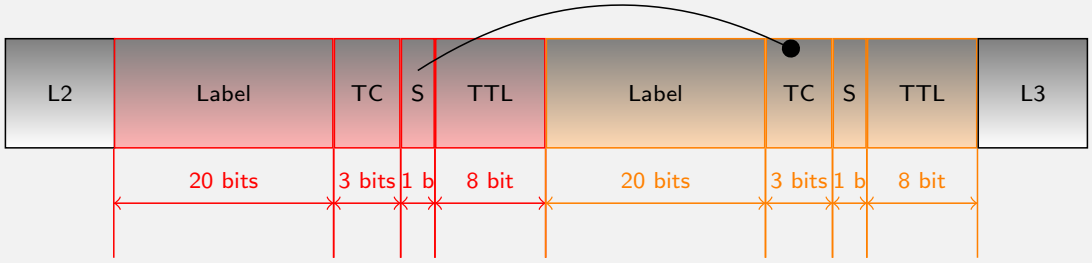
MPLS: ...Label...



- Payload type is lost
  - Label can be used to recover it
  - Depend of the context

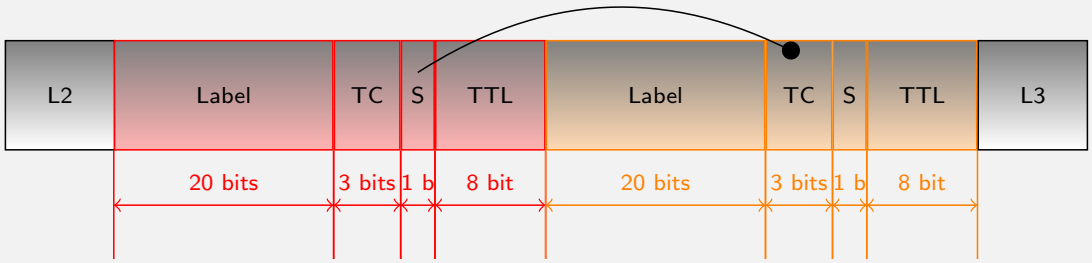
MPLS: ...Label...

Link States ▶ Areas



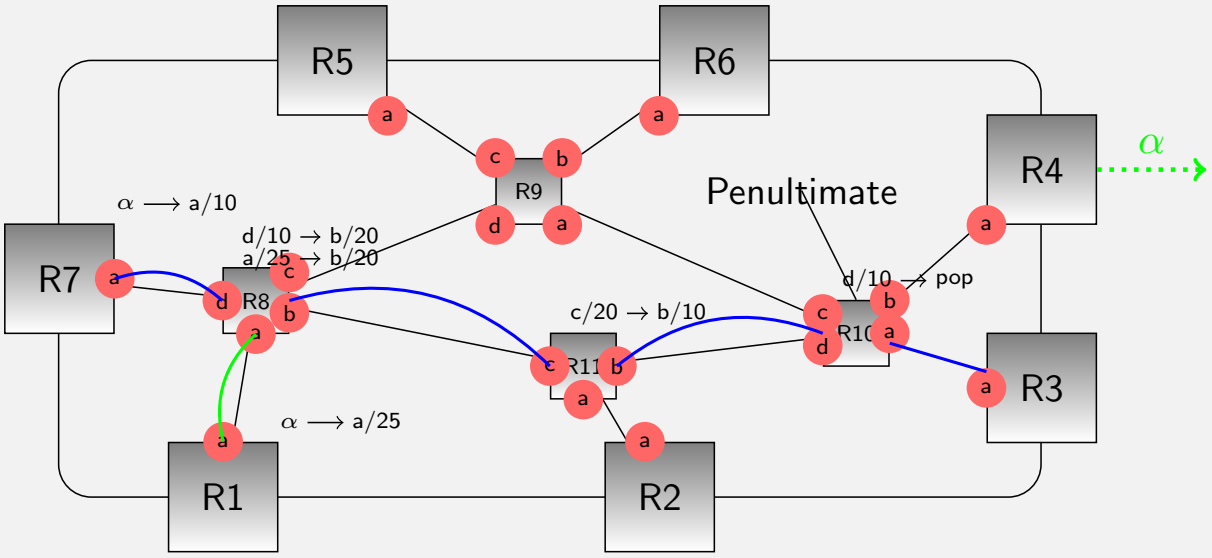
- TC : Traffic Class (RFC 5462)
  - initially experimental (RFC 3032), but RFC 3270 defines class of services
  - map DiffServ bits, IEEE 802.1Q classes, ...
  - For instance, three first bits of DiffServ Classes :
  - ECN can also be used (RFC 5129)

MPLS: ...Label...



- S=1 : Another label is stacked
- TTL : decreased by one by each LSR
  - Avoid loops based on routing protocol
  - Initial value copied from IP field (TTL or HL)
    - traceroute works.
    - ICMP extension for MPLS (RFC 4950)

MPLS: ...Switching VC merging



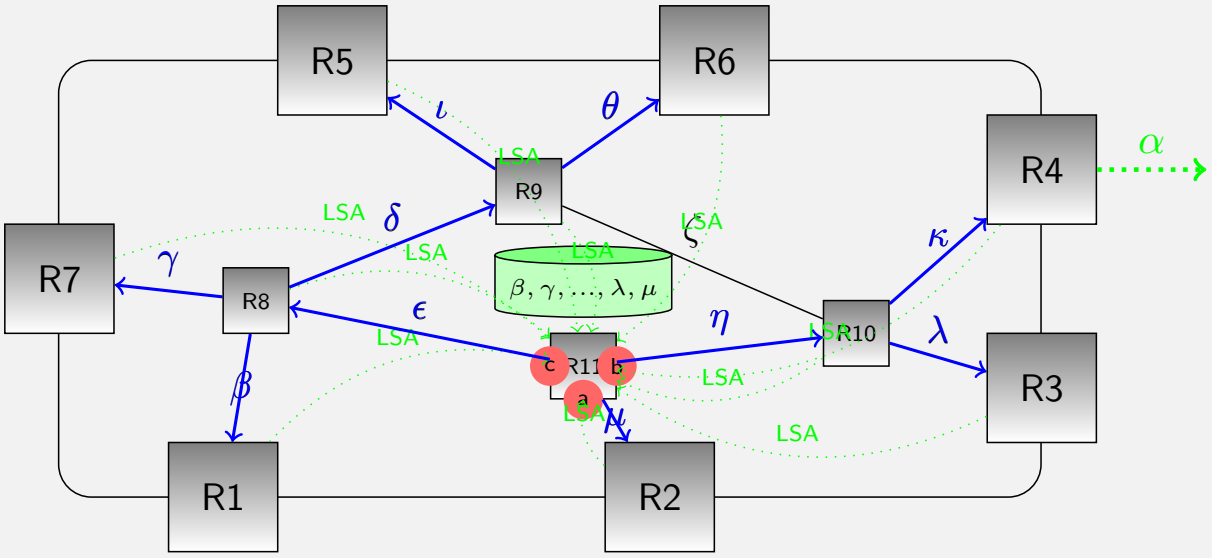
How to build FEC and LIB ?

- Manually
- Using Label Distribution Protocol
  - works with an IGP (prefix discovery and route selection)
  - two label distribution modes :
    - independent : each LSR works independently on FEC
    - ordered : downstream LSR has to establish the FEC first
  - two label retention modes :
    - conservative : keep in memory just useful labels
    - liberal : memorize label not useful for the LSP (faster reroute)
- RSVP-TE
  - independent of IGP: allow the provider to force paths and reserve resources
  - allow rerouting in case of link failure
- MP-BGP
  - Dissociate internal routing and external routing
  - VPN, IPv6 over IPv4, IPv4 over IPv6

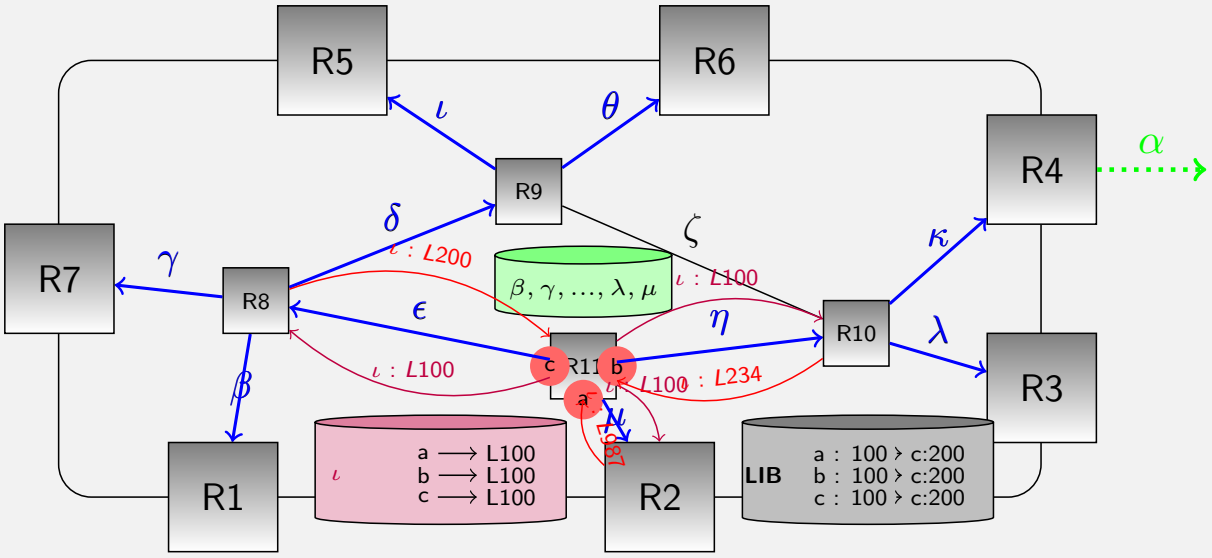
Principle: Overview

- An IGP is running on all LSR
- LSR know site prefixes
- LSR issue a label for each prefix and interface
- LSR use LDP to send the binding between label and prefix
- When receiving a label, if the sender is the next hop in the IGP's Shortest Path Tree, the label is pushed to the LIB.

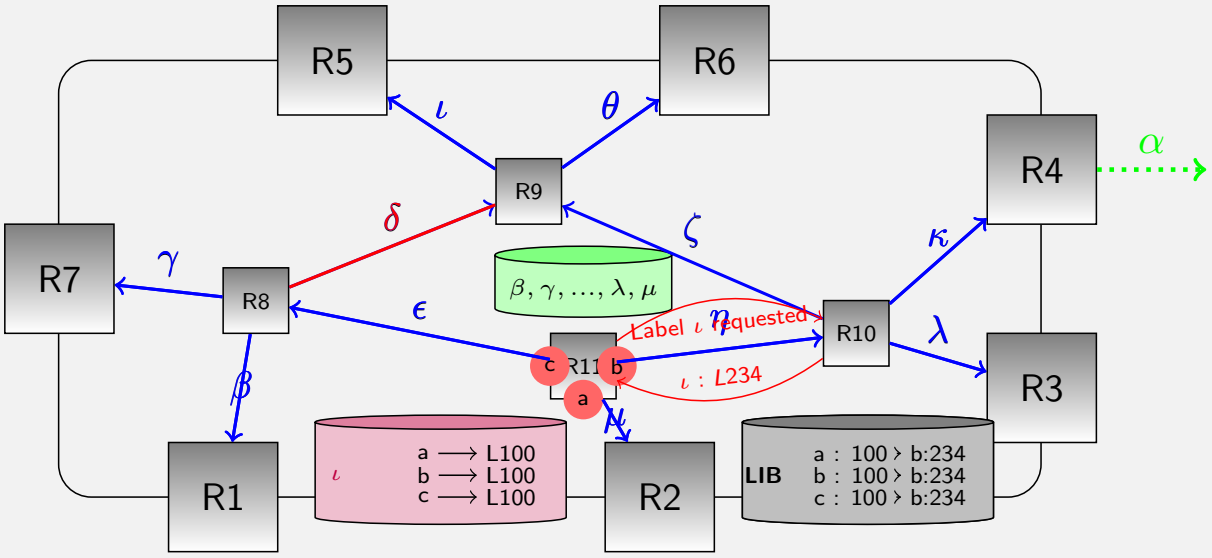
Label distribution example



Label distribution example



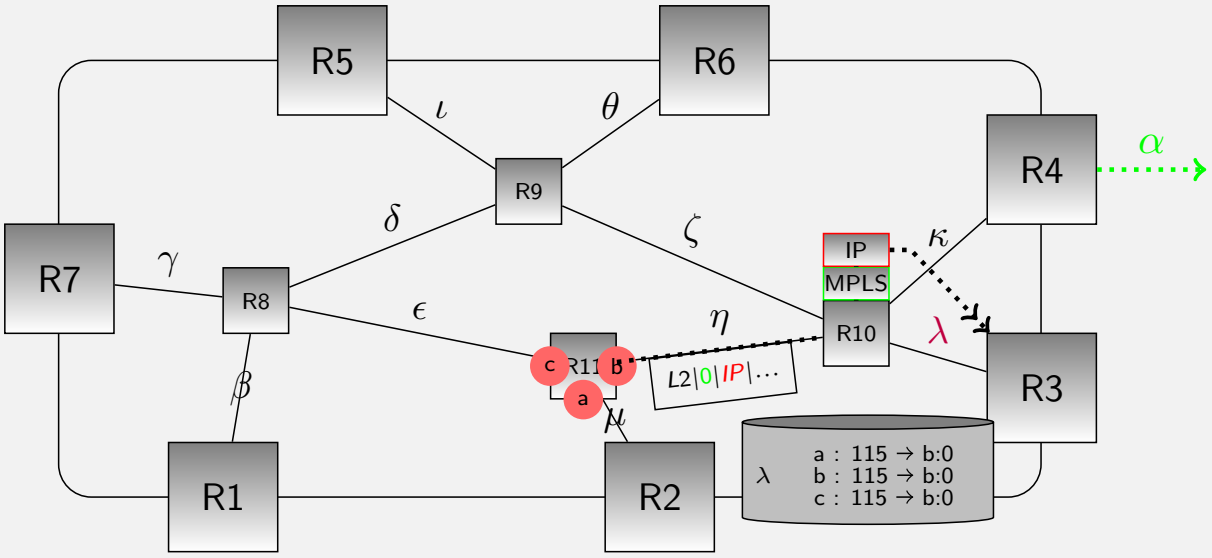
Label distribution example



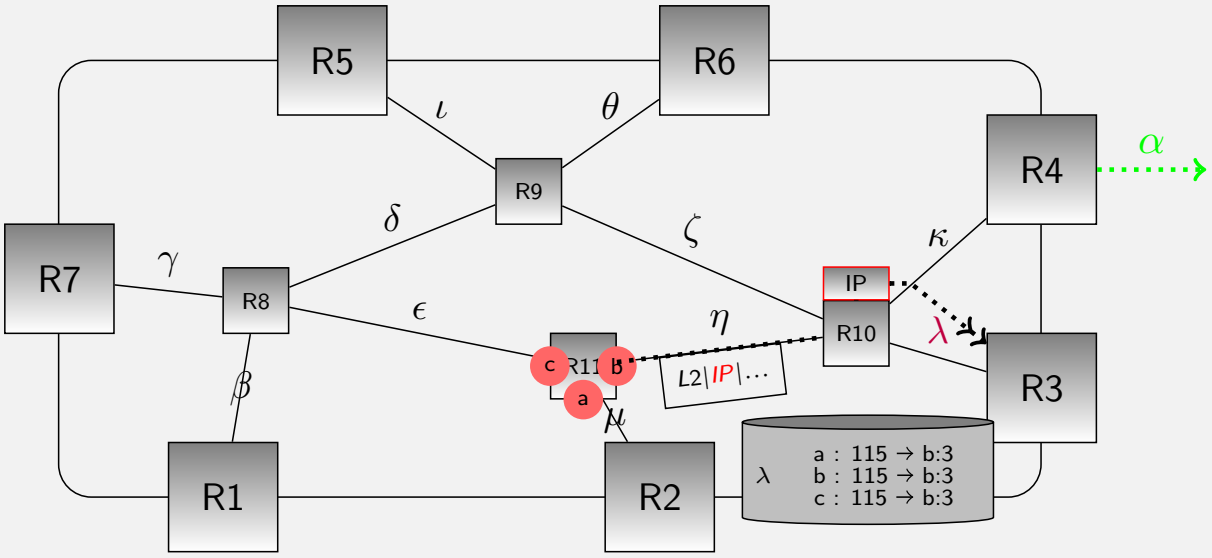
POP

- Pop can be :
  - implicit (value 3): the LSR remove the label and send it to the forwarding engine.
    - the LSR must know which forwarding engine from the context.
    - NEVER sent on the link.
    - On the last link, packet is sent normally avoiding MPLS pop on the last router.
  - explicit (value 0 for IPv4, 2 for IPv6):
    - when receiving this label, the LSR must send the MPLS payload to the corresponding forwarding engine.
    - Sent on the link
    - Initially last on the label stack, RFC 4182 removes this constraint

Label distribution example



Label distribution example



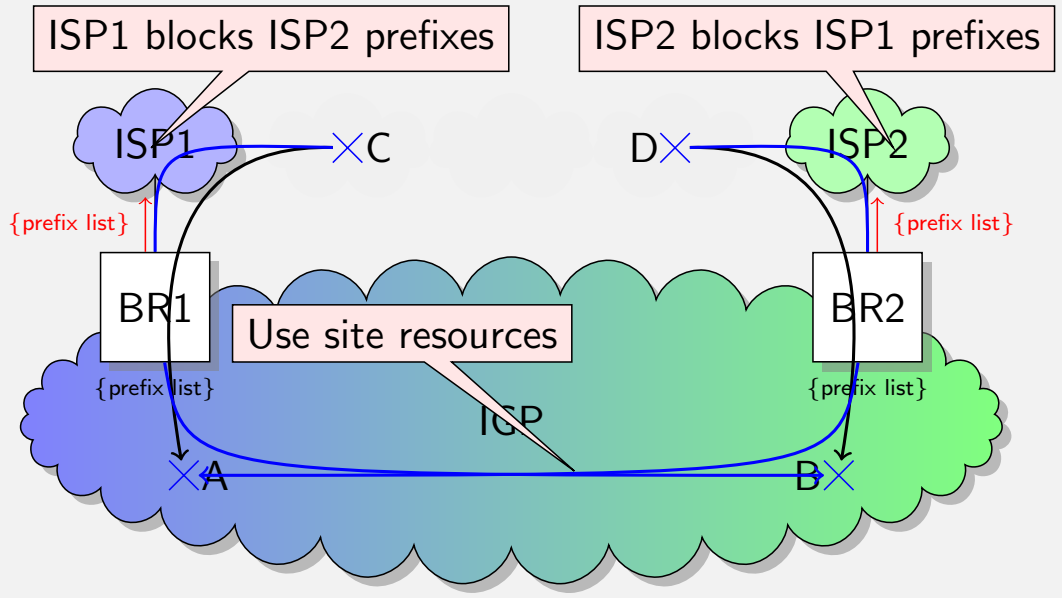
BGP

External Gateway Protocol

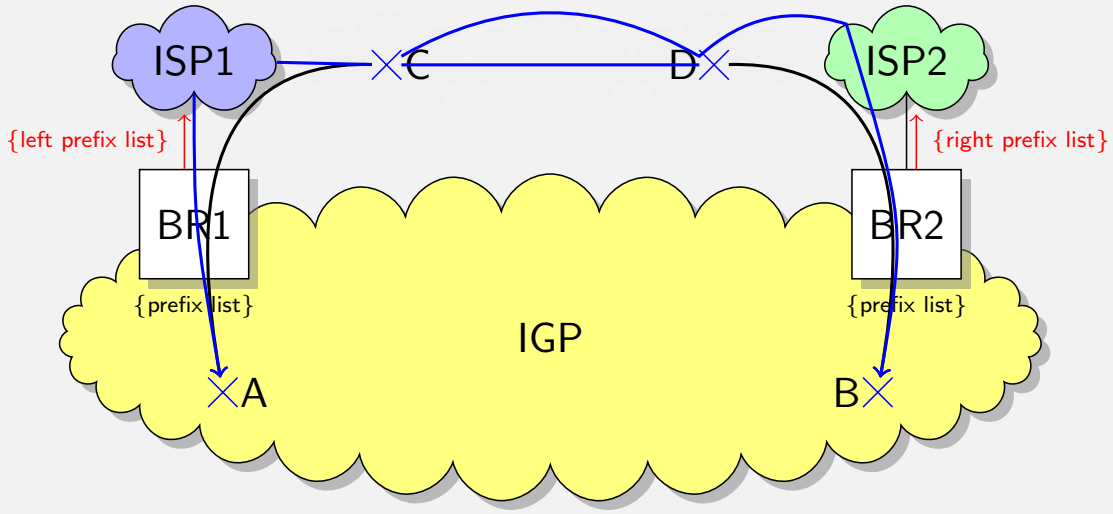
Definition

- IGP simplifies network management
  - Neighbor discovery
  - Shortest path criteria (technical)
- EGP defines the interface between providers
  - Hide internal topology to the other peer
  - Select traffic to send/receive (political)
  - Control information received or send
  - Full configuration
  - Internal topology is not important $\Longrightarrow$ graph of ISP

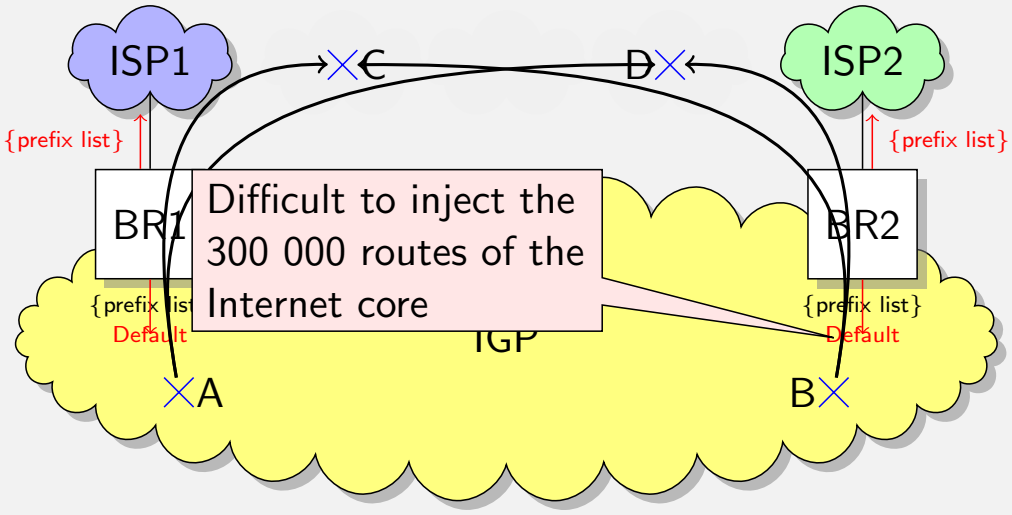
Routing policies



Routing policies

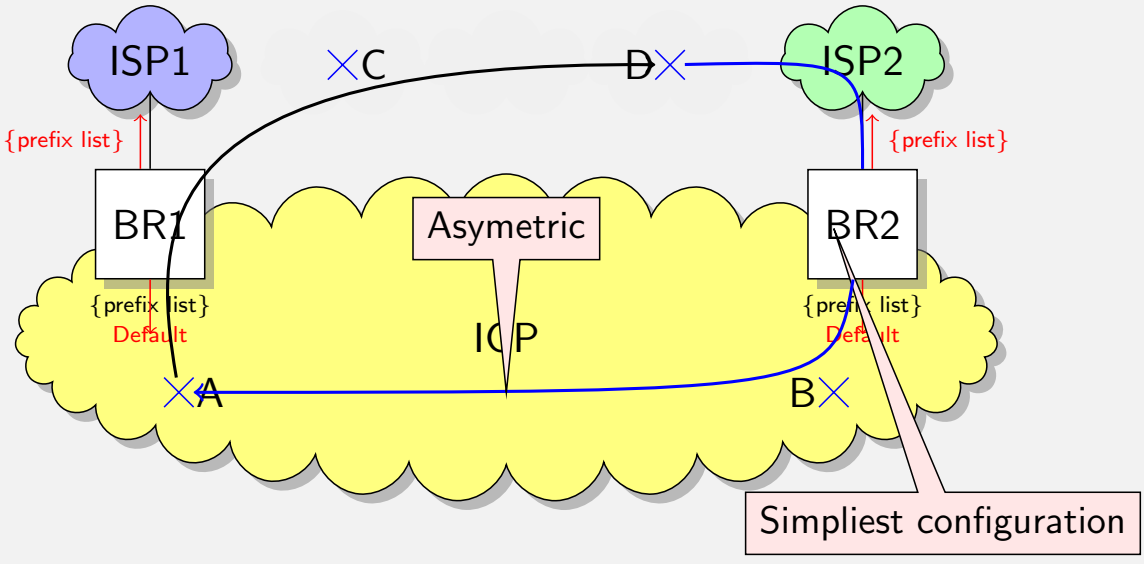


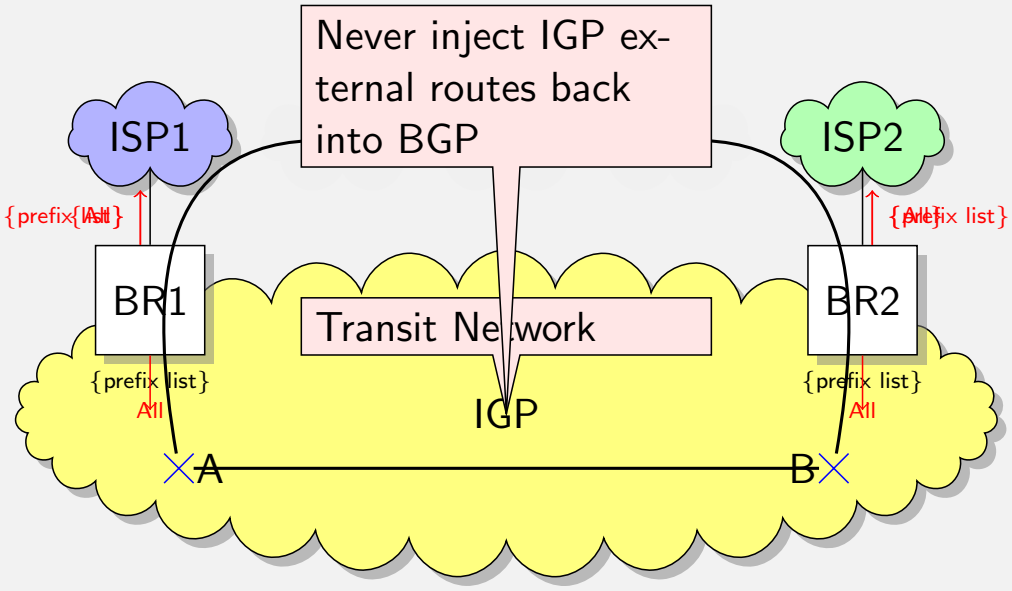
Routing policies



Routing policies

BGP ▶ External Gateway Protocol





BGP

AS Peering

Autonomous System

- Autonomous System (AS): Network managed by an authority
  - Internet is a graph of AS
- Prefixes cannot be used to identify an AS
- ASN: Autonoumous System Number is used
  - on 16 bits, but transition to 32 bits is intiated
  - assigned by RIR
  - values higher than 65 000 are private
- `whois` databases store information related to ASes
  - Can be used to make a link between a prefix and an ASN
  - Each RIR maintains AS database ⇒ `whois.ripe.net`
  - `whois.ra.net` regroups all registries information

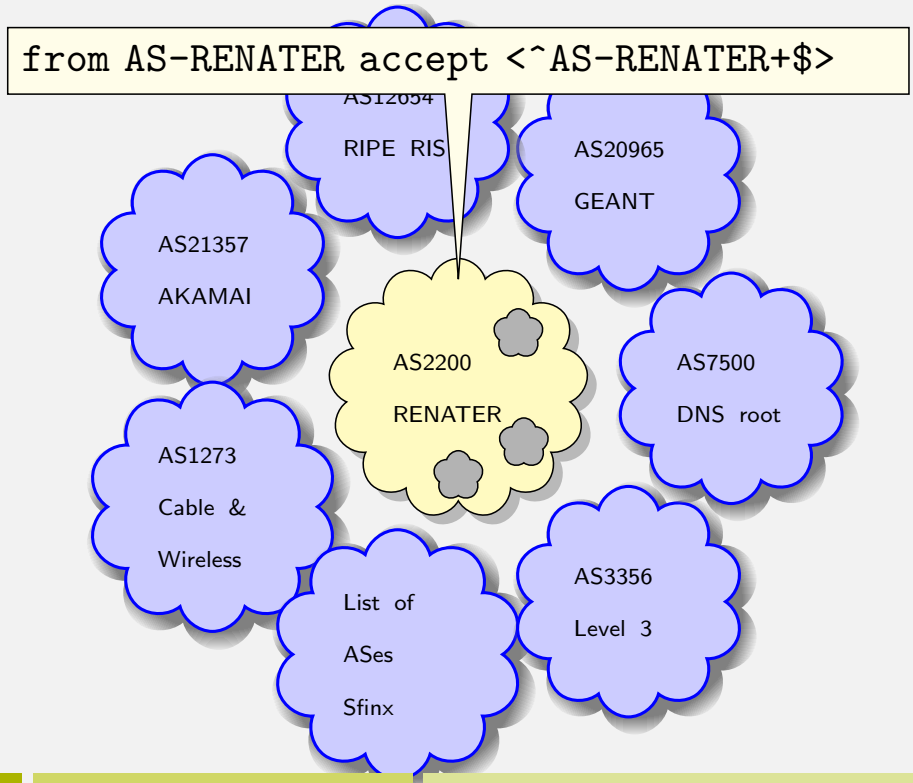
Example: whois -h whois.ripe.net AS2200


```
aut-num: AS2200
as-name: FR-RENATER
descr: Reseau National de telecommunications pour la Technologie
descr: l'Enseignement et la Recherche
descr: FR
import: from AS-RENATER accept <^AS-RENATER+$>
import: from AS-SFINX-PEERS action pref=190; accept <^AS-SFINX-PEERS .$>
import: from AS20965 action pref=300; accept ANY
import: from AS7500 action pref=190; accept AS7500
import: from AS1273 action pref=300; accept ANY
import: from AS3356 action pref=300; accept ANY
export: to AS-RENATER announce ANY
export: to AS-SFINX-PEERS announce AS-RENATER
export: to AS20965 announce AS-RENATER AS7500
export: to AS12654 announce AS-RENATER
export: to AS21357 announce AS-RENATER
export: to AS1273 announce AS-RENATER
export: to AS3356 announce AS-RENATER
admin-c: GR1378-RIPE
tech-c: GR1378-RIPE
mnt-by: RENATER-MNT
source: RIPE # Filtered
```

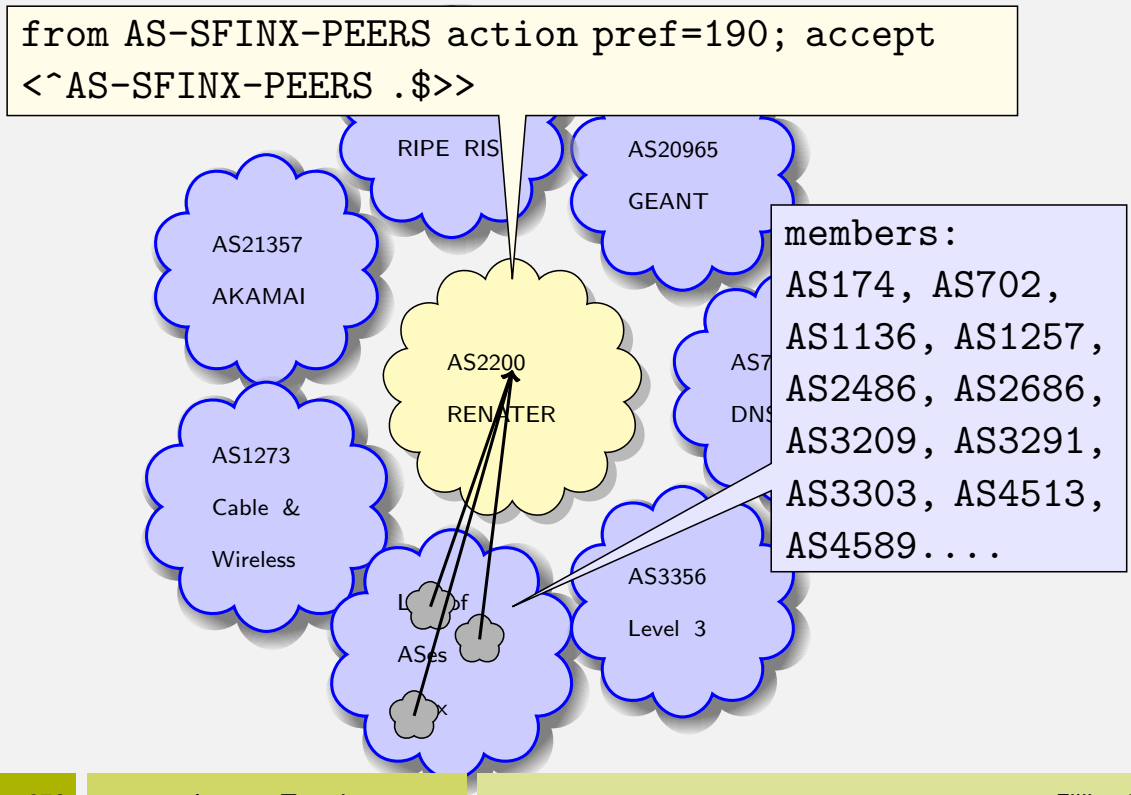
RPSL: Routing Policy Specification Language (RFC 4012)

- describes routing policies between provider in whois database

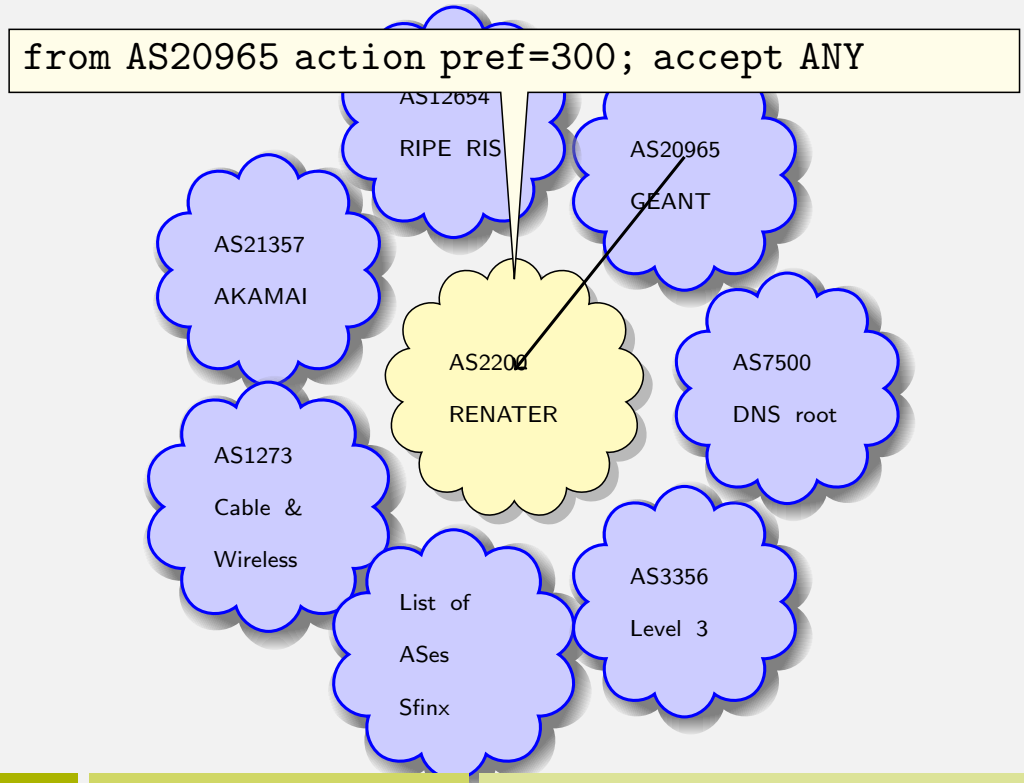
Network topology from AS2200



Network topology from AS2200

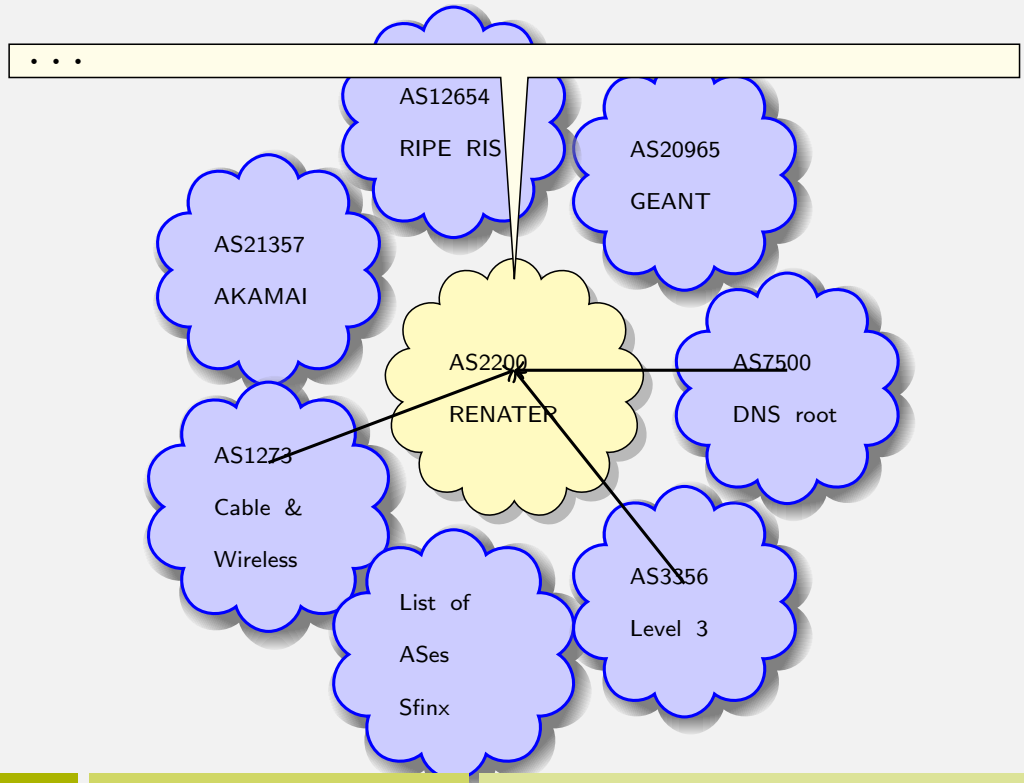


Network topology from AS2200



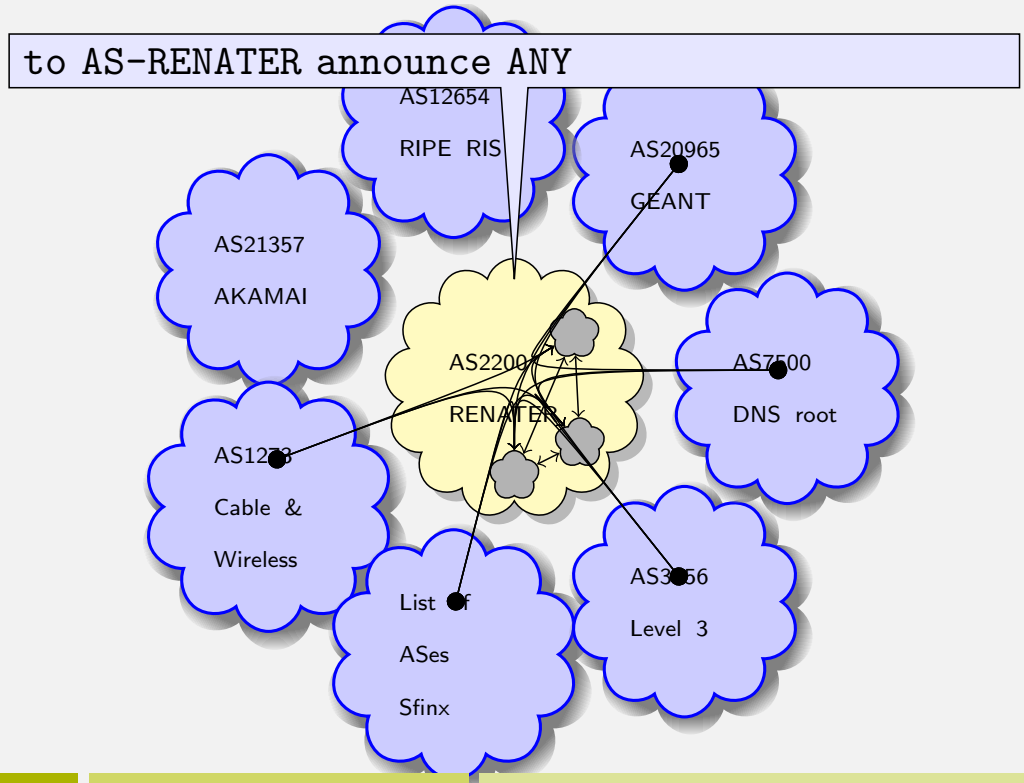
TELECOM Bretagne

Network topology from AS2200

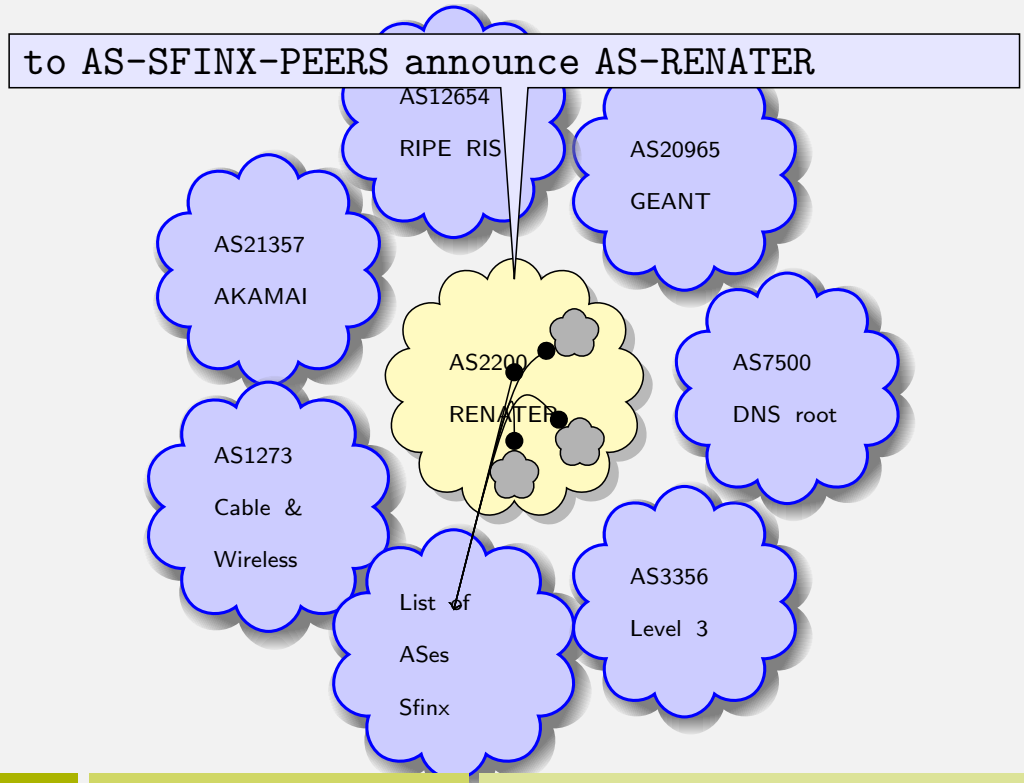


TELECOM Bretagne

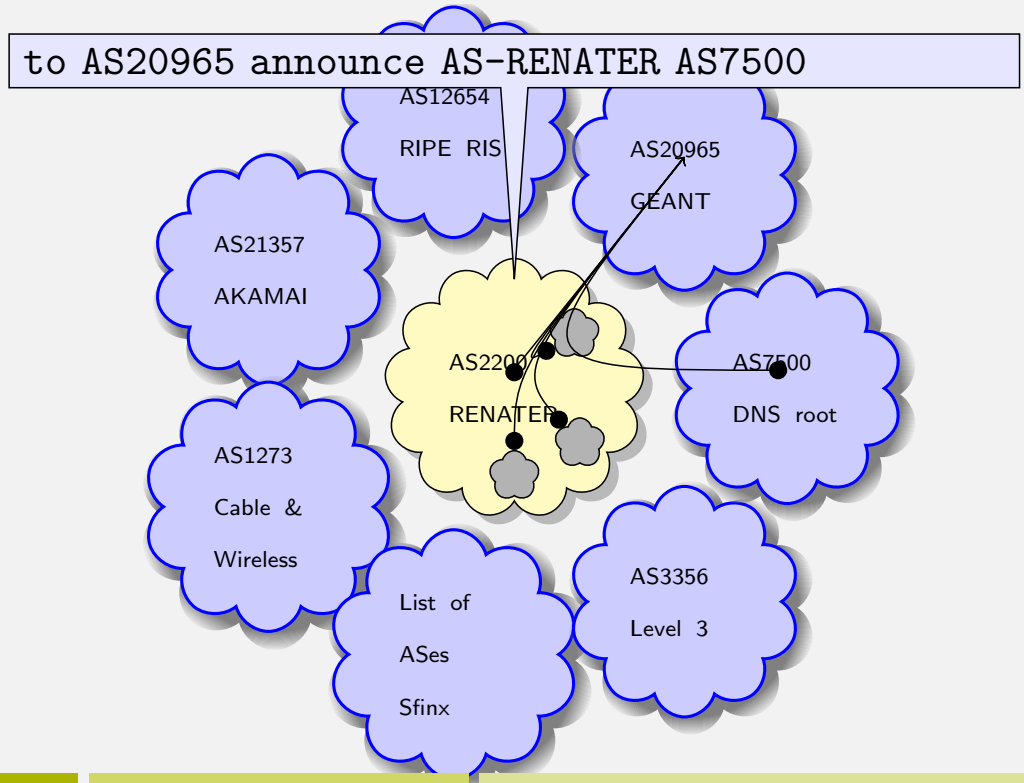
Network topology from AS2200



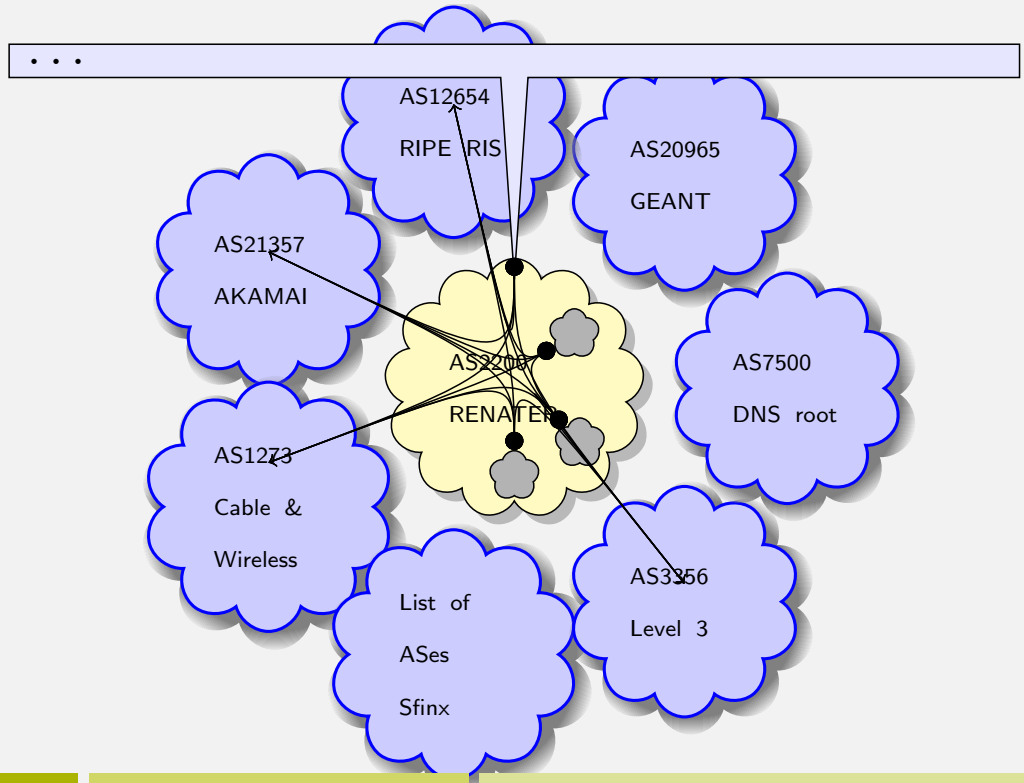
Network topology from AS2200



Network topology from AS2200



Network topology from AS2200



Traceroute

- Traceroute sends a packet with TTL/HL=1
- First router discards packet and sends an ICMP message
- DNS reverse query allows to know router's name
- `whois` dabase query allows to know ASN
- Also called NANOG traceroute
  - see `ftp://ftp.login.com/pub/software/traceroute/`
  - `apt-get install traceoute-nanog`
- `-A` allows to display ASN

Traceroute

BGP ▶ AS Peering

```
# traceroute -A www.icu.ac.kr
traceroute to www.icu.ac.kr (143.248.116.26), 64 hops max, 40 byte packets
1  mgs-c5.ipv6.rennes.enst-bretagne.fr (192.108.119.190) [AS2200]  0 ms  0 ms  0 ms
2  * * *
3  te4-1-caen-rtr-021.noc.renater.fr (193.51.189.54) [AS2200] [MPLS: Label 227 Exp 0]  7 ms  7 ms  7 ms
4  te4-1-rouen-rtr-021.noc.renater.fr (193.51.189.46) [AS2200] [MPLS: Label 348 Exp 0]  7 ms  7 ms  7 ms
5  te0-0-0-1-paris1-rtr-001.noc.renater.fr (193.51.189.49) [AS2200]  42 ms  7 ms  7 ms
6  renater.rt1.par.fr.geant2.net (62.40.124.69) [AS20965]  7 ms  7 ms  7 ms
7  so-3-0-0.rt1.lon.uk.geant2.net (62.40.112.106) [AS20965]  14 ms  14 ms  14 ms
8  so-2-0-0.rt1.ams.nl.geant2.net (62.40.112.137) [AS20965]  22 ms  22 ms  25 ms
9  xe-2-3-0.102.rtr.newy32aoa.net.internet2.edu (198.32.11.50) [<NONE>]  106 ms  106 ms  106 ms
10 ge-0-0-0.0.rtr.chic.net.internet2.edu (64.57.28.72) [<NONE>]  133 ms  144 ms  133 ms
11 kreonet2-abilene.kreonet.net (134.75.108.45) [AS1237]  188 ms  188 ms  188 ms
12 134.75.108.209 (134.75.108.209) [AS1237]  303 ms  302 ms  302 ms
13 supersiren.kreonet.net (134.75.20.20) [AS1237/AS9318]  302 ms  302 ms  302 ms
14 kaist-gw.kaist.ac.kr (143.248.119.1) [AS1781/AS9318]  295 ms  295 ms  295 ms
15 143.248.119.85 (143.248.119.85) [AS1781/AS9318]  295 ms  295 ms  295 ms
16 143.248.116.253 (143.248.116.253) [AS1781/AS9318]  295 ms  296 ms  296 ms
17 143.248.116.218 (143.248.116.218) [AS1781/AS9318]  296 ms  297 ms  296 ms
18 iccweb.kaist.ac.kr (143.248.116.26) [AS1781/AS9318]  296 ms  296 ms  296 ms
```

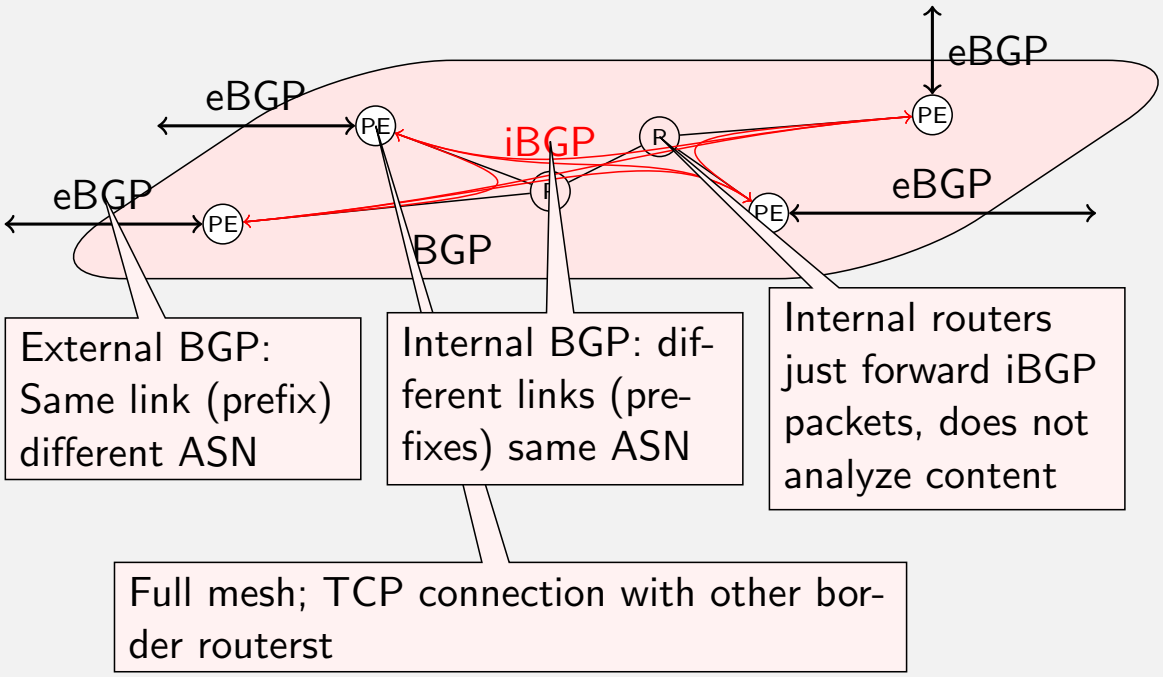
BGP

Border Gateway Protocol

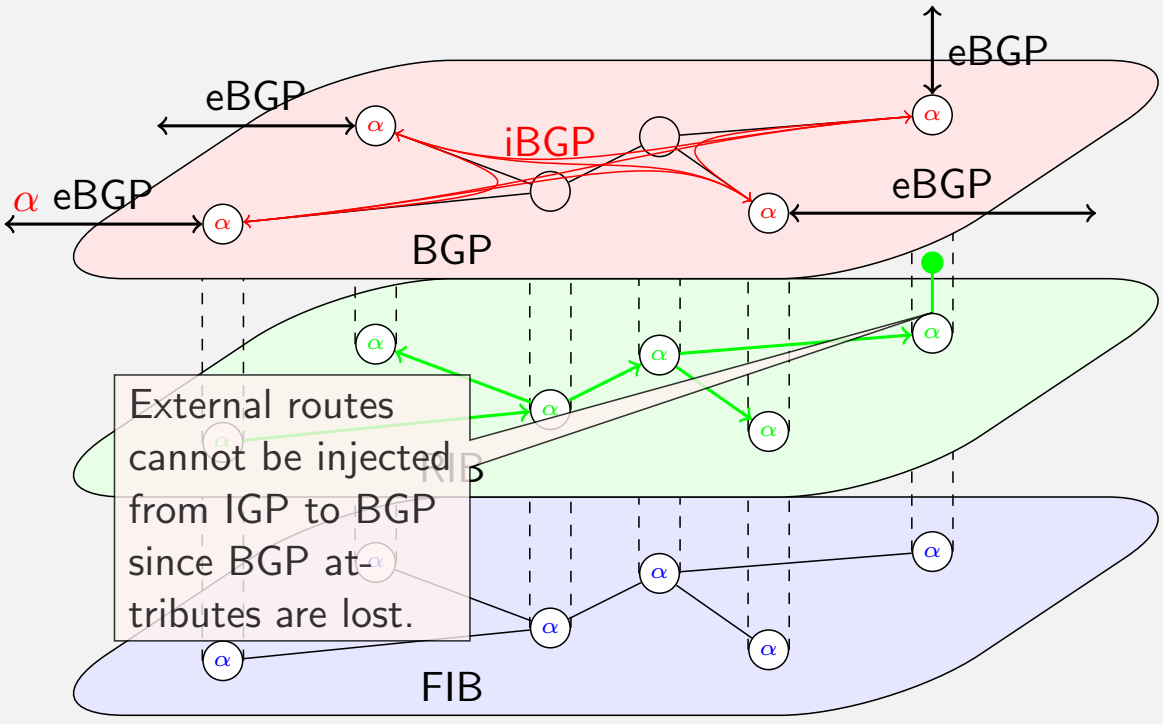
MP-BGP

- BGP: Border Gateway Protocol (RFC 4271)
  - Version 4 supports CIDR
  - MP-BGP (RFC 4760) is a smooth evolution to support IPv6, MPLS, Multicast, VPN, ...
- Based on TCP (port 179) to exchange information
  - point-to-point
  - reliable exchange
- Four types of messages
  - OPEN
    - Agree on values such as ASN
    - MP-BGP defines capabilites
  - UPDATE: Prefix add or withdraw
  - NOTIFICATION: Error messages
  - KEEPALIVE: Periodically sent

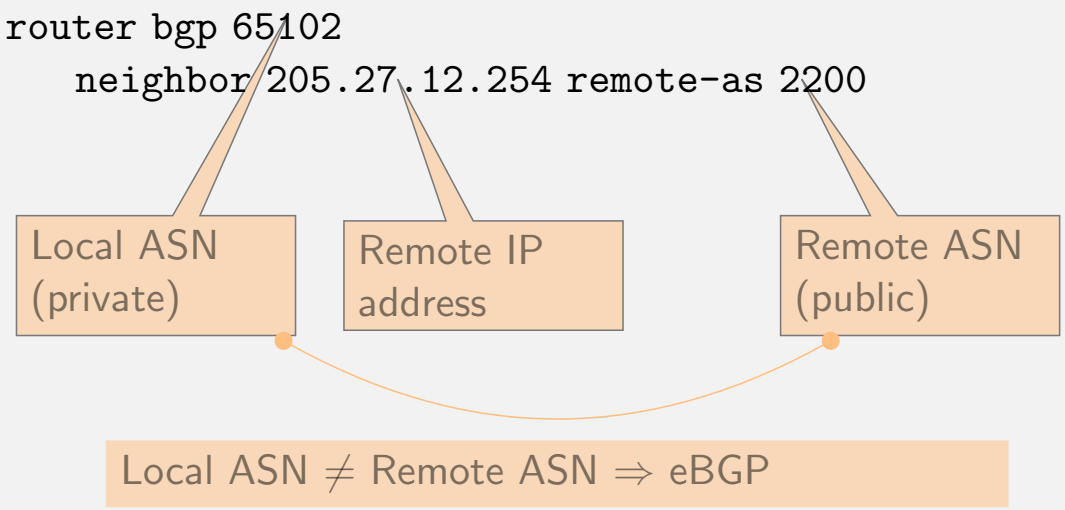
Principles



Principles



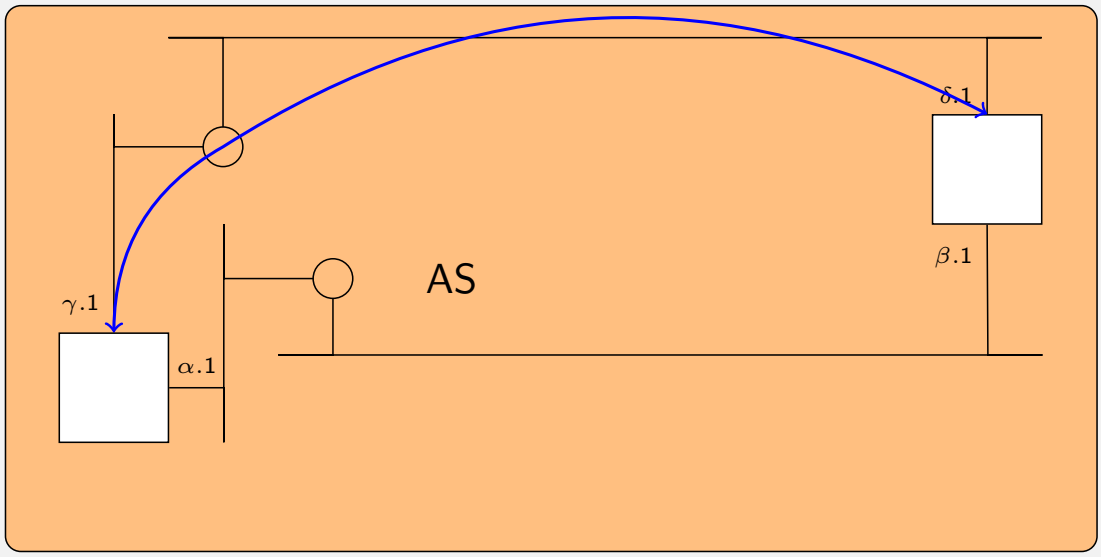
BGP configuration example



Loopback address

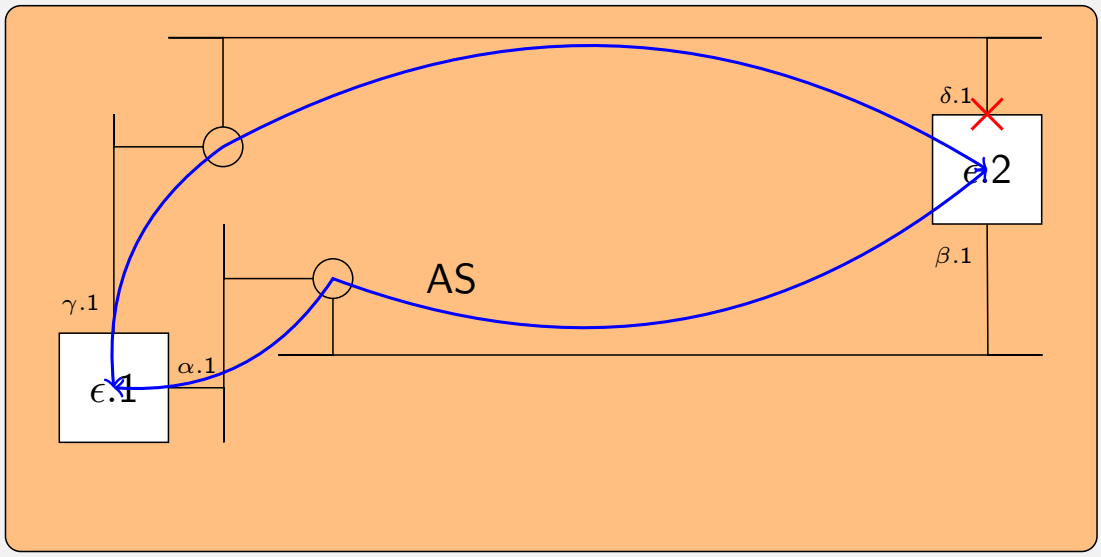
- BGP uses TCP
- Both peers need to configure other end address
- A router may have several interfaces
- if one interface is down, some iBGP peers may have to be configured again
- loopback address is used to give an address to the router (not to the interface).
- loopback addresses are announced on IGP

Loopback address




```
router bgp 1234
    neighbor δ.1 remote-as 1234
```


Loopback address




```
router bgp 1234
    neighbor ϵ.1 remote-as 1234
```

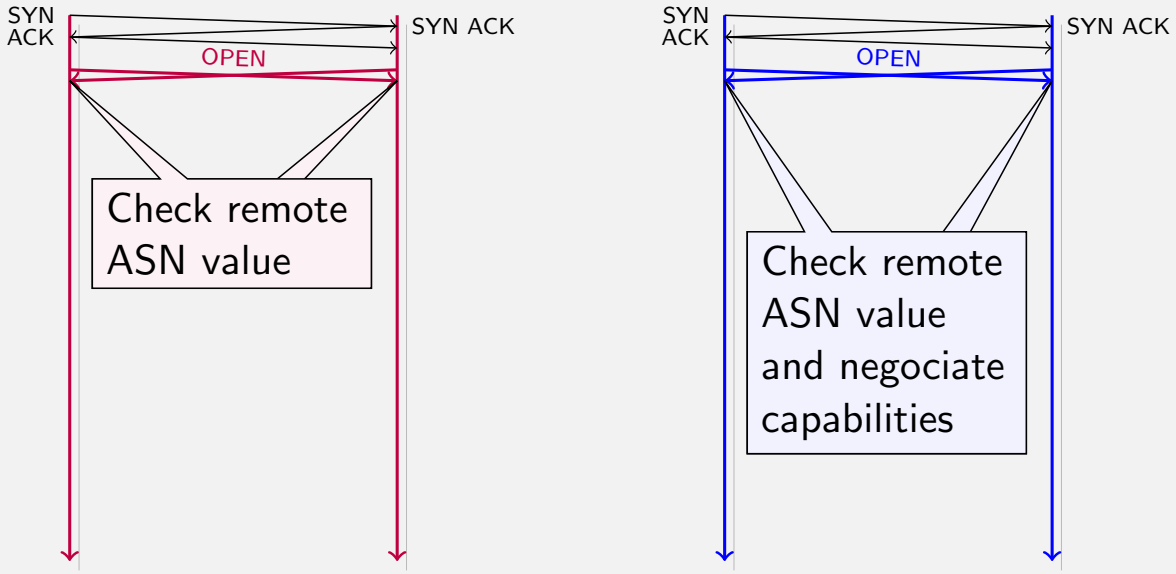
BGP

BGP Attributes

BGP attributes

1	ORIGIN	$m.w.$	RFC 4271
2	AS_PATH	$m.w.$	RFC 4271
3	NEXT_HOP	$m.w.$	RFC 4271
4	MULTI_EXIT_DISC	$o.\bar{t}.$	RFC 4271
5	LOCAL_PREF	$w.o.$	RFC 4271
6	ATOMIC_AGGREGATE	$w.o.$	RFC 4271
7	AGGREGATOR	$w.t.$	RFC 4271
8	COMMUNITY	$o.t.$	RFC 1997
9	ORIGINATOR_ID	$o.\bar{t}.$	RFC 4456
10	CLUSTER_LIST	$o.\bar{t}.$	RFC 4456
14	MP_REACH_NLRI	$o.\bar{t}.$	RFC 4760
15	MP_UNREACH_NLRI	$o.\bar{t}.$	RFC 4760
17	AS4_PATH	$o.t.$	RFC 4893
18	AS4_AGGREGATOR	$o.t.$	RFC 4893

BGPv4 versus MP-BGP

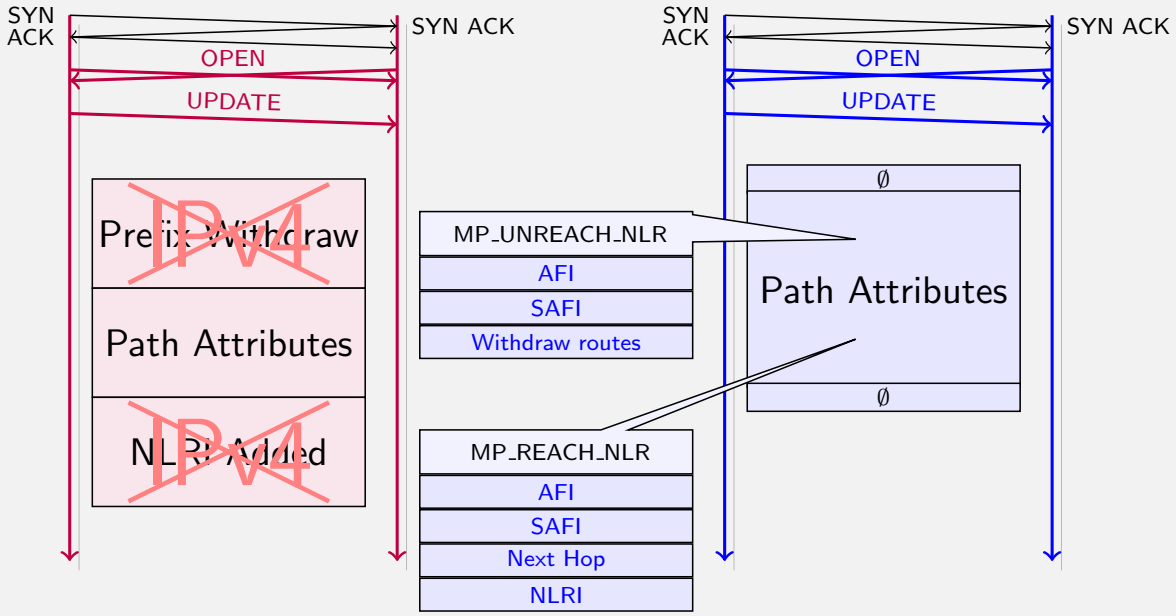


MP-BGP capabilities

- AFI : Address Family Identifier [1]
  - 1: IPv4
  - 2: IPv6
- SAFI: Subsequent Address Family Identifiers [2]
  - 1: unicast
  - 2: multicast
  - 4: MPLS
  - 65: Support for 4-octet ASN
  - 67: BGP 4over6
  - 68: BGP 6over4

[1] http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml
[2] http://www.iana.org/assignments/safi-namespace

BGPv4 versus MP-BGP



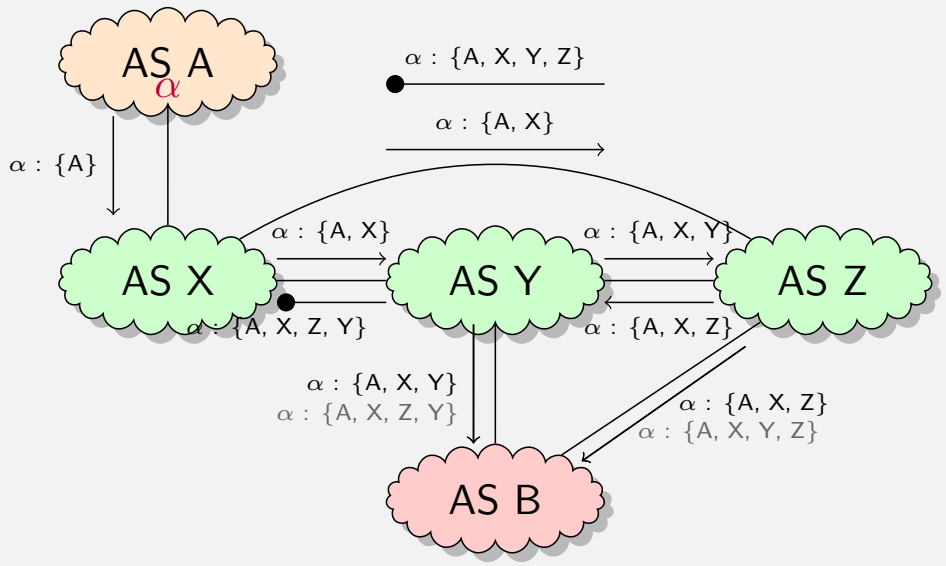
BGP attributes

1	ORIGIN	*m.w.*	RFC 4271
2	AS_PATH	*m.w.*	RFC 4271
3	NEXT_HOP	*m.w.*	RFC 4271
4	MULTI_EXIT_DISC	*o.$\bar{t}$.*	RFC 4271
5	LOCAL_PREF	*w.o.*	RFC 4271
6	ATOMIC_AGGREGATE	*w.o.*	RFC 4271
7	AGGREGATOR	*w.t.*	RFC 4271
8	COMMUNITY	*o.t.*	RFC 1997
9	ORIGINATOR_ID	*o.$\bar{t}$.*	RFC 4456
10	CLUSTER_LIST	*o.$\bar{t}$.*	RFC 4456
14	MP_REACH_NLRI	*o.$\bar{t}$.*	RFC 4760
15	MP_UNREACH_NLRI	*o.$\bar{t}$.*	RFC 4760
17	AS4_PATH	*o.t.*	RFC 4893
18	AS4_AGGREGATOR	*o.t.*	RFC 4893

AS_PATH and ORIGIN

- Every AS add its ASN in the AS_PATH
- Used to detect loops
  - can not work for iBGP since the ASN remains the same
- AS_PATH length is also used to select best annoucement
  - the shorter, the best
  - this metric does not take into account either the number of routers nor the links quality (bandwidth, ...)
  - An AS may add several time its ASN to tell that the route should not be selected (backup)
  - Other policies may be used to select appropriate route
- ORIGIN tells how the route was injected into BGP
  - IGP=BGP, EGP (obsolete), unknown
  - Mainly used during EGP/BGP transition.

AS_PATH



Example from Canada (edited)

AS_PATH NEXT_HOP ORIGIN Shortest AS_PATH selected

```
route-views.optus.net.au>sh ip bgp 192.108.119.0
BGP routing table entry for 192.108.119.0/24, version 25940801
Paths: (3 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  7474 7473 1273 2200
    203.13.132.53 from 203.13.132.53 (172.26.32.13)
      Origin IGP, localpref 100, valid, external
  7473 1273 2200
    202.160.242.71 from 202.160.242.71 (202.160.242.71)
      Origin IGP, localpref 100, valid, external, best
  7474 7473 1273 2200
    203.13.132.35 from 203.13.132.35 (172.26.32.42)
      Origin IGP, localpref 100, valid, external
```

Shortest AS_PATH selected

Cable & Wireless RENATER

Optus Communications Singapore Telecommunications

TELECOM Bretagne

BGP attributes

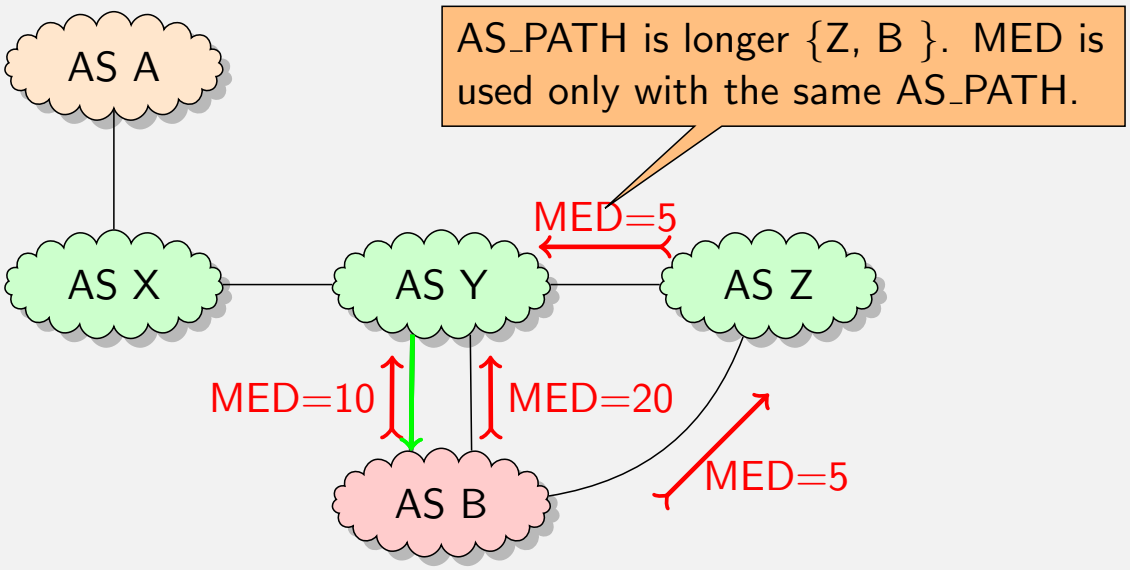
1	ORIGIN	*m.w.*	RFC 4271
2	AS_PATH	*m.w.*	RFC 4271
3	NEXT_HOP	*m.w.*	RFC 4271
4	MULTI_EXIT_DISC	*o.$\bar{t}$.*	RFC 4271
5	LOCAL_PREF	*w.o.*	RFC 4271
6	ATOMIC_AGGREGATE	*w.o.*	RFC 4271
7	AGGREGATOR	*w.t.*	RFC 4271
8	COMMUNITY	*o.t.*	RFC 1997
9	ORIGINATOR_ID	*o.$\bar{t}$.*	RFC 4456
10	CLUSTER_LIST	*o.$\bar{t}$.*	RFC 4456
14	MP_REACH_NLRI	*o.$\bar{t}$.*	RFC 4760
15	MP_UNREACH_NLRI	*o.$\bar{t}$.*	RFC 4760
17	AS4_PATH	*o.t.*	RFC 4893
18	AS4_AGGREGATOR	*o.t.*	RFC 4893

TELECOM Bretagne

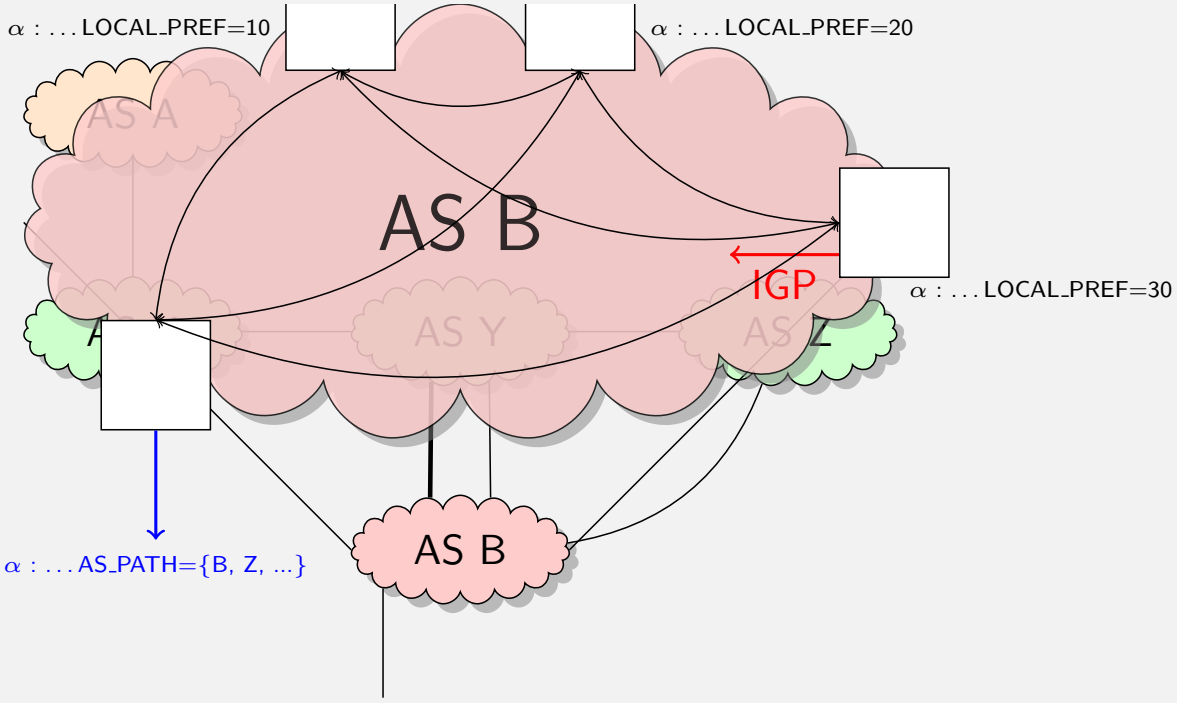
MULTI_EXIT_DISC and LOCAL_PREF

- MULTI_EXIT_DISC is used to signal another as which BGP peering is the best
  - used between two ASes
  - the smaller, the better
- LOCAL_PREF is used inside an AS to assign a cost to a NRLI
  - used internally
  - the higher, the better
- Duplication of ASN in AS_PATH can also be used to make alternative path less attractive.
  - preferences can be propagated on all ASes in the AS_PATH.

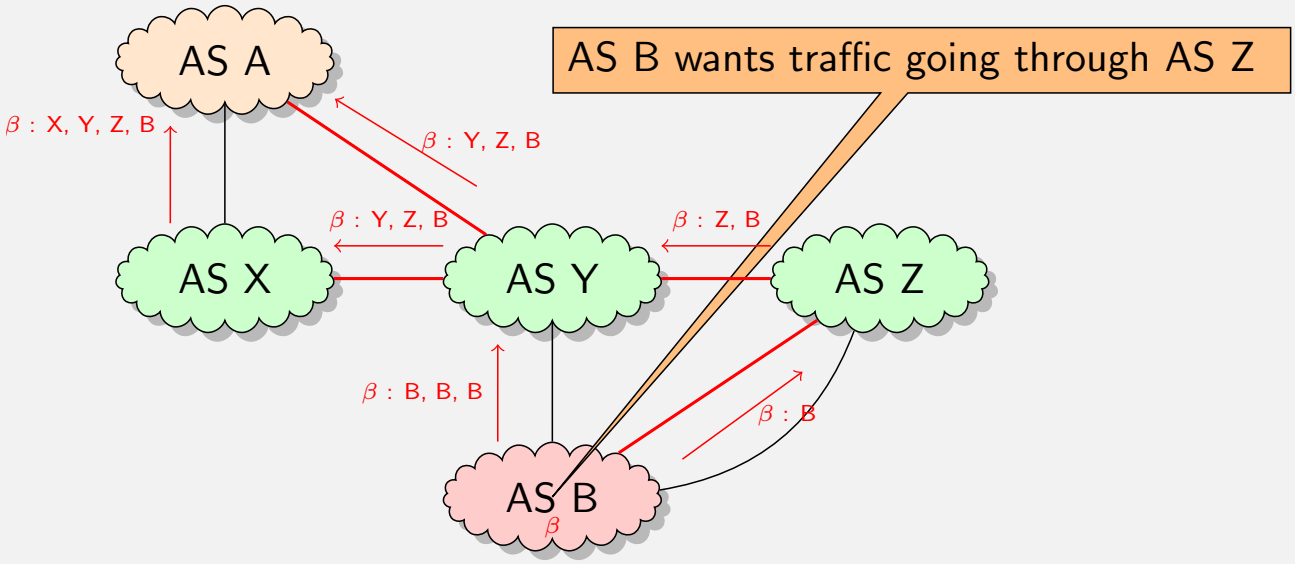
MULTI_EXIT_DISC



LOCAL_PREF



AS_PATH



BGP attributes

1	ORIGIN	*m.w.*	RFC 4271
2	AS_PATH	*m.w.*	RFC 4271
3	NEXT_HOP	*m.w.*	RFC 4271
4	MULTI_EXIT_DISC	$o.\overline{t}.$	RFC 4271
5	LOCAL_PREF	*w.o.*	RFC 4271
6	ATOMIC_AGGREGATE	*w.o.*	RFC 4271
7	AGGREGATOR	*w.t.*	RFC 4271
8	COMMUNITY	*o.t.*	RFC 1997
9	ORIGINATOR_ID	$o.\overline{t}.$	RFC 4456
10	CLUSTER_LIST	$o.\overline{t}.$	RFC 4456
14	MP_REACH_NLRI	$o.\overline{t}.$	RFC 4760
15	MP_UNREACH_NLRI	$o.\overline{t}.$	RFC 4760
17	AS4_PATH	*o.t.*	RFC 4893
18	AS4_AGGREGATOR	*o.t.*	RFC 4893

Community

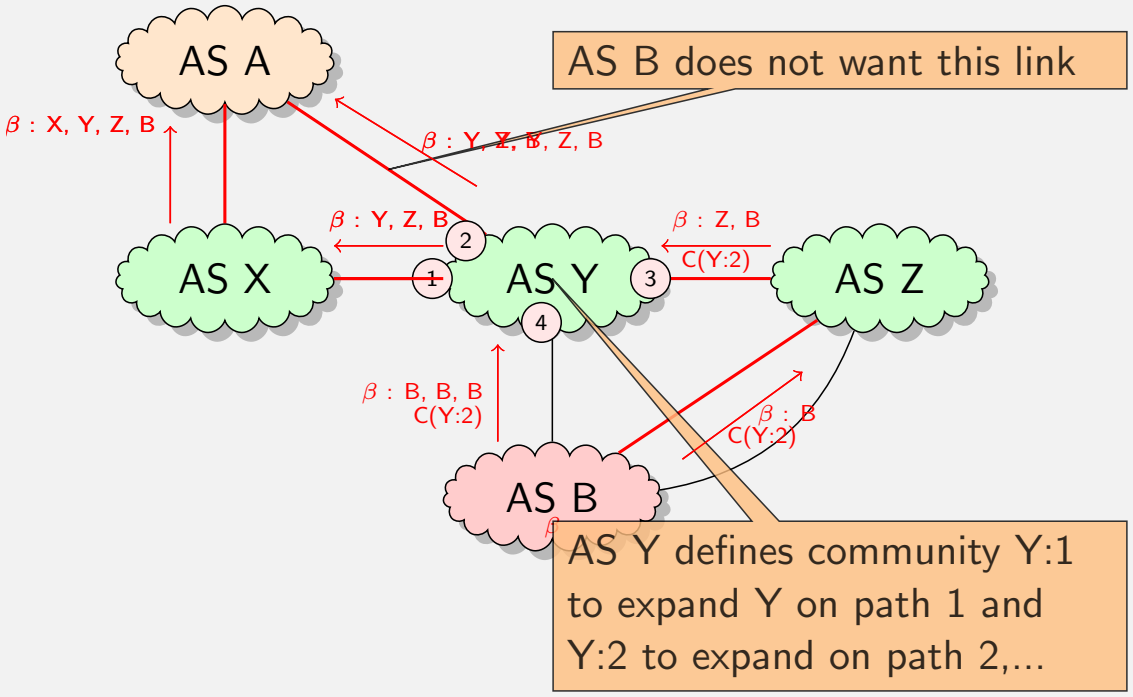
- Initially developed to facilitate identification of a group of prefixes
  - the same Community attribute is used to identify this group
  - peer just has to look for this attribute to select processing
- Some behaviors have been standardized
  - NO_EXPORT (`0xFFFFFF01`)
  - NO_ADVERTISE (`0xFFFFFF02`)
  - NO_EXPORT_SUBCONFED (`0xFFFFFF03`)
- Providers can develop their own community attribute
  - `ASN:xxxx` where `xxxx` is defined by provider
  - For 32 byte ASN, see RFC 4360 .
  - can be used either to
    - add information about the processing (ISP → site) or
    - to control how announcement is processed (ISP ← site)

Example from Level3

Paths: (3 available, best #1, table Default-IP-Routing-Table)
Advertised to peer-groups:
INTERNAL
1273 2200
195.66.224.182 from 195.66.224.182 (195.2.1.105)
Origin IGP, metric 0, localpref 100, valid, external, best
Community: 1273:12250 2200:1000 2200:2200 5459:1 5459:60
1273 2200, (received-only)
195.66.224.182 from 195.66.224.182 (195.2.1.105)
Origin IGP, metric 0, localpref 100, valid, external
Community: 1273:12250 2200:1000 2200:2200
1273 2200
195.66.232.239 from 195.66.232.239 (195.66.232.239)
Origin IGP, metric 0, localpref 100, valid, internal
Community: 1273:12250 2200:1000 2200:2200 5459:3 5459:6

See `http://www.cw.net/outcommunities.shtml`:
This format is 1273:SRCCC
S=1 customer route R=2 Europe 250=France

Example: Community Attribute



BGP attributes

1	ORIGIN	*m.w.*	RFC 4271
2	AS_PATH	*m.w.*	RFC 4271
3	NEXT_HOP	*m.w.*	RFC 4271
4	MULTI_EXIT_DISC	*o.t.*	RFC 4271
5	LOCAL_PREF	*w.o.*	RFC 4271
6	ATOMIC_AGGREGATE	*w.o.*	RFC 4271
7	AGGREGATOR	*w.t.*	RFC 4271
8	COMMUNITY	*o.t.*	RFC 1997
9	ORIGINATOR_ID	*o.t.*	RFC 4456
10	CLUSTER_LIST	*o.t.*	RFC 4456
14	MP_REACH_NLRI	*o.t.*	RFC 4760
15	MP_UNREACH_NLRI	*o.t.*	RFC 4760
17	AS4_PATH	*o.t.*	RFC 4893
18	AS4_AGGREGATOR	*o.t.*	RFC 4893

TELECOM Bretagne's prefixes


```
route-server-east.gt.ca>sh ip bgp 192.108.119.0
BGP routing table entry for 192.108.119.0/24, version 171516980
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  1273 2200
    66.59.190.227 from 66.59.190.227 (66.59.190.227)
      Origin IGP, metric 0, localpref 110, valid, internal, best
      Community: 6539:3 6539:305 6539:520 6539:1400

...

route-server-east.gt.ca>sh ip bgp 193.52.74.0
BGP routing table entry for 193.52.0.0/16, version 171516968
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  1273 2200, (aggregated by 2200 193.51.178.10)
    66.59.190.227 from 66.59.190.227 (66.59.190.227)
      Origin IGP, metric 0, localpref 110, valid, internal, atomic-aggregate, best
      Community: 6539:3 6539:305 6539:520 6539:1400

...
```

Prefix allocated before 1994, belongs to TELECOM Bretagne

As is in routing table

CIDR prefix allocated after 1994 by Renater

Agregated in routing table

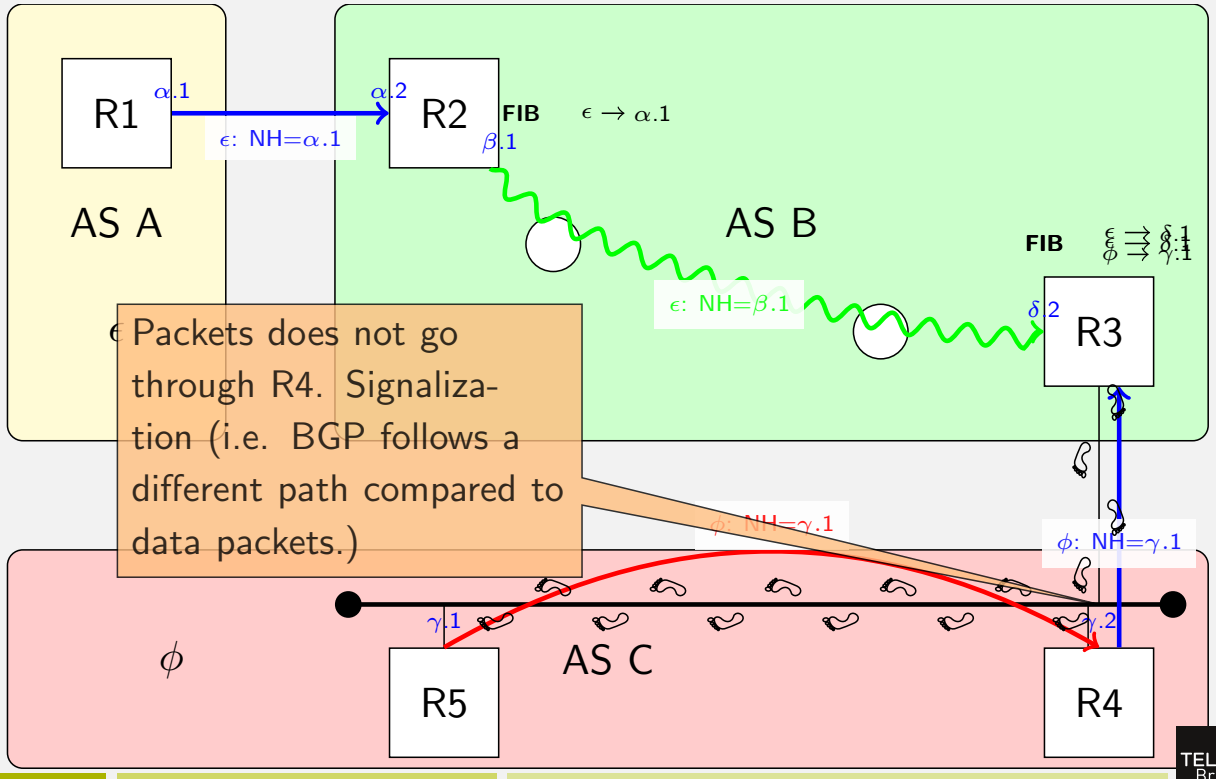
BGP attributes

1	ORIGIN	*m.w.*	RFC 4271
2	AS_PATH	*m.w.*	RFC 4271
3	NEXT_HOP	*m.w.*	RFC 4271
4	MULTI_EXIT_DISC	*o.t.*	RFC 4271
5	LOCAL_PREF	*w.o.*	RFC 4271
6	ATOMIC_AGGREGATE	*w.o.*	RFC 4271
7	AGGREGATOR	*w.t.*	RFC 4271
8	COMMUNITY	*o.t.*	RFC 1997
9	ORIGINATOR_ID	*o.t.*	RFC 4456
10	CLUSTER_LIST	*o.t.*	RFC 4456
14	MP_REACH_NLRI	*o.t.*	RFC 4760
15	MP_UNREACH_NLRI	*o.t.*	RFC 4760
17	AS4_PATH	*o.t.*	RFC 4893
18	AS4_AGGREGATOR	*o.t.*	RFC 4893

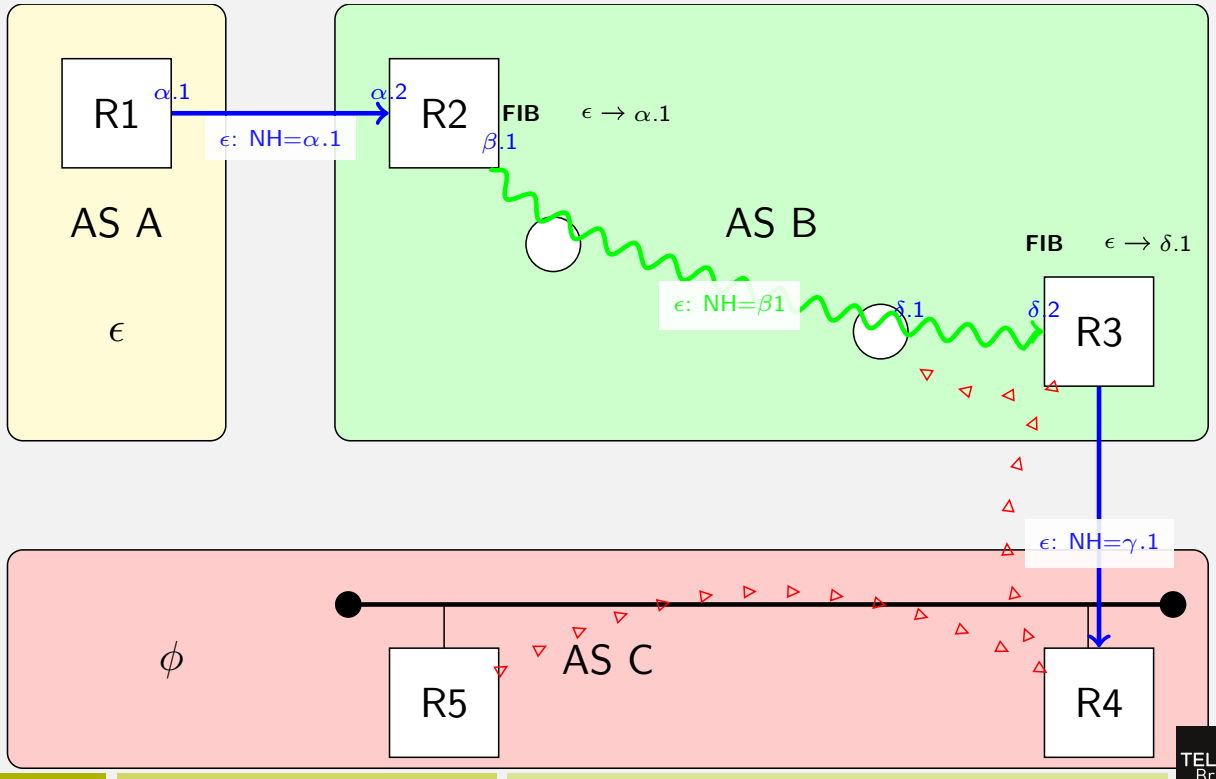
Next Hop Attribute

- Used to populate FIBs NH value is not always pushed in the FIB
- NH contains the IP address of the router (or interface) of the sending router
- For eBGP value is directly pushed since eBGP is a connection on a prefix shared by both ends
  - A router may not change NH value if the sender shares the same prefix.
- with iBGP, BGP NH and IGP NH are different. For iBGP, since some IGP routers may be between both ends, interactions with IGP SPT is needed do find FIB Next Hop.

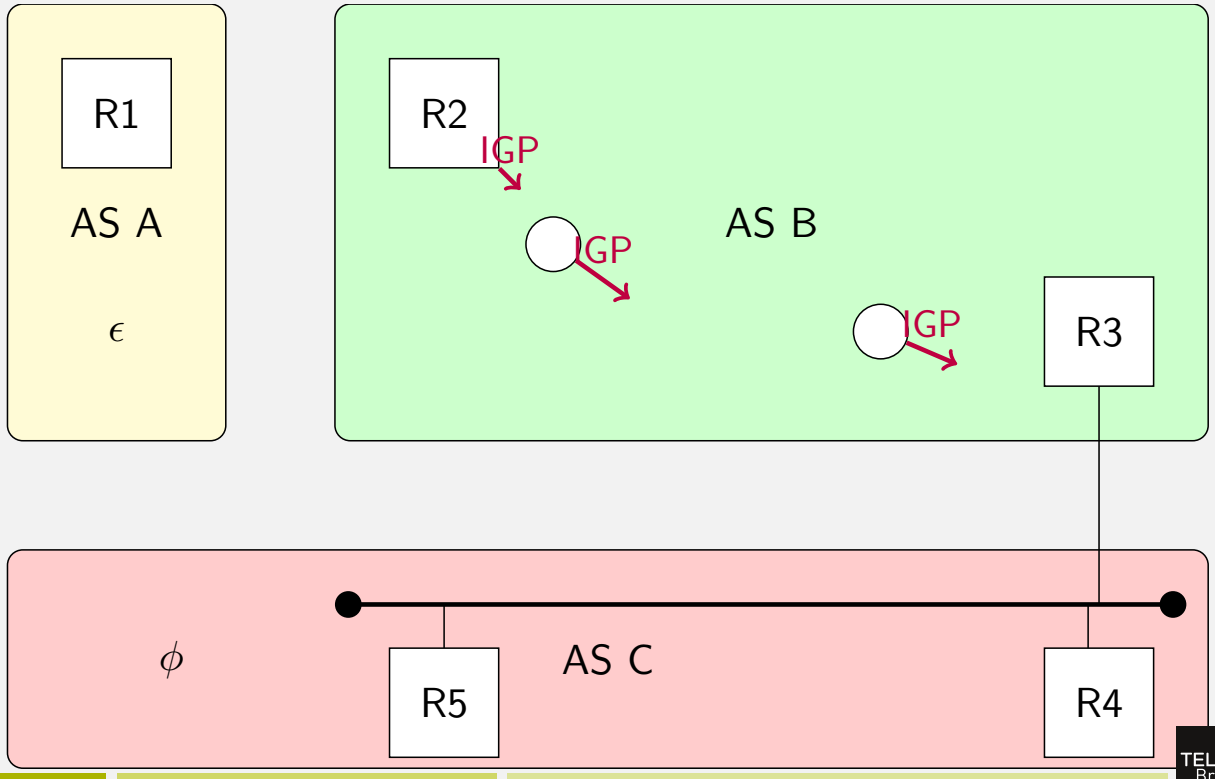
Next Hop different behaviors



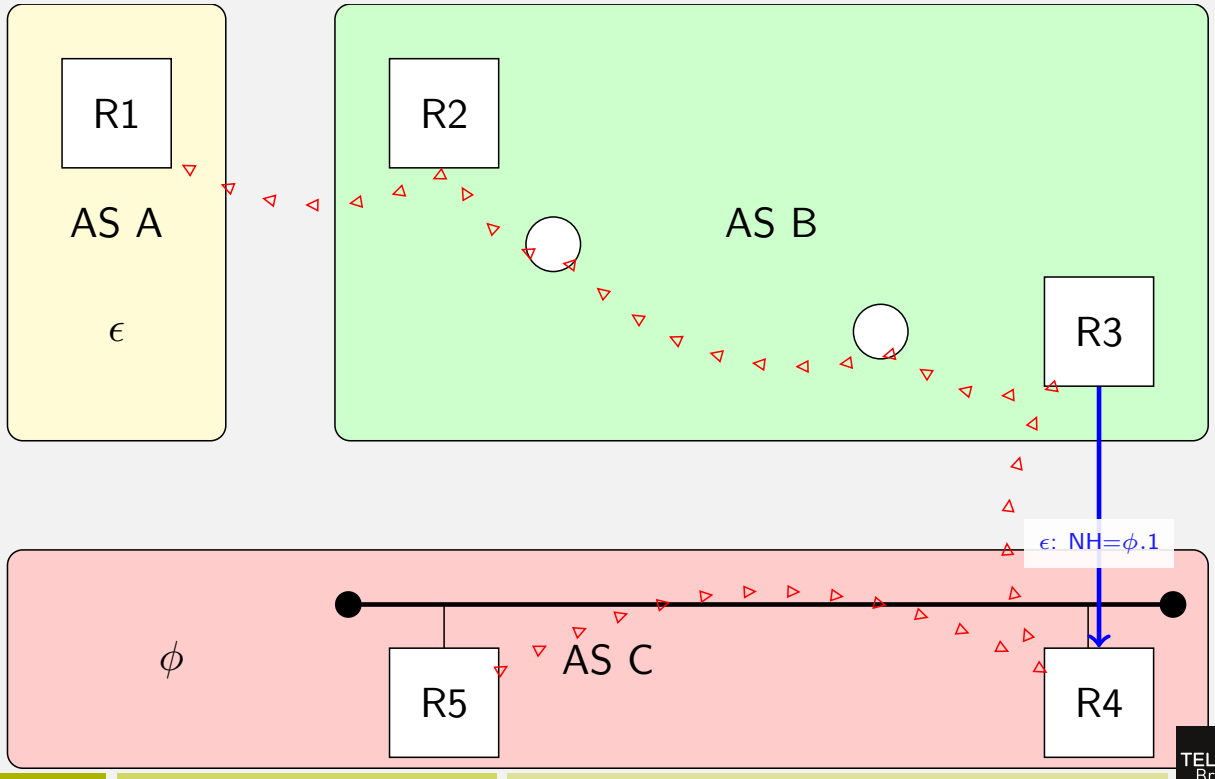
Synchronization



Synchronization



Synchronization



Route Selection

1. Reject bad announcements:
   - 1.1 ASN is in Path
   - 1.2 NEXT_HOP inaccessible
   - 1.3 filtering based on prefix or AS_PATH
2. Select:
   - 2.1 highest local weight (local cost of BGP peering) otherwise
   - 2.2 highest LOCAL_PREF otherwise
   - 2.3 shortest AS_PATH otherwise
   - 2.4 prefix originated by the router (cisco's `aggregate` or `network BGP`) otherwise
   - 2.5 lowest origin type (IGP $<$ EGP $<$?) otherwise
   - 2.6 lowest multi-exit discriminator otherwise
   - 2.7 eBGP over iBGP paths otherwise
   - 2.8 closest IGP neighbor otherwise
   - 2.9 lowest IP address, in BGP router ID or ORIGINATOR_ID otherwise
   - 2.10 shortest route reflection cluster list
   - 2.11 lowest neighbor address

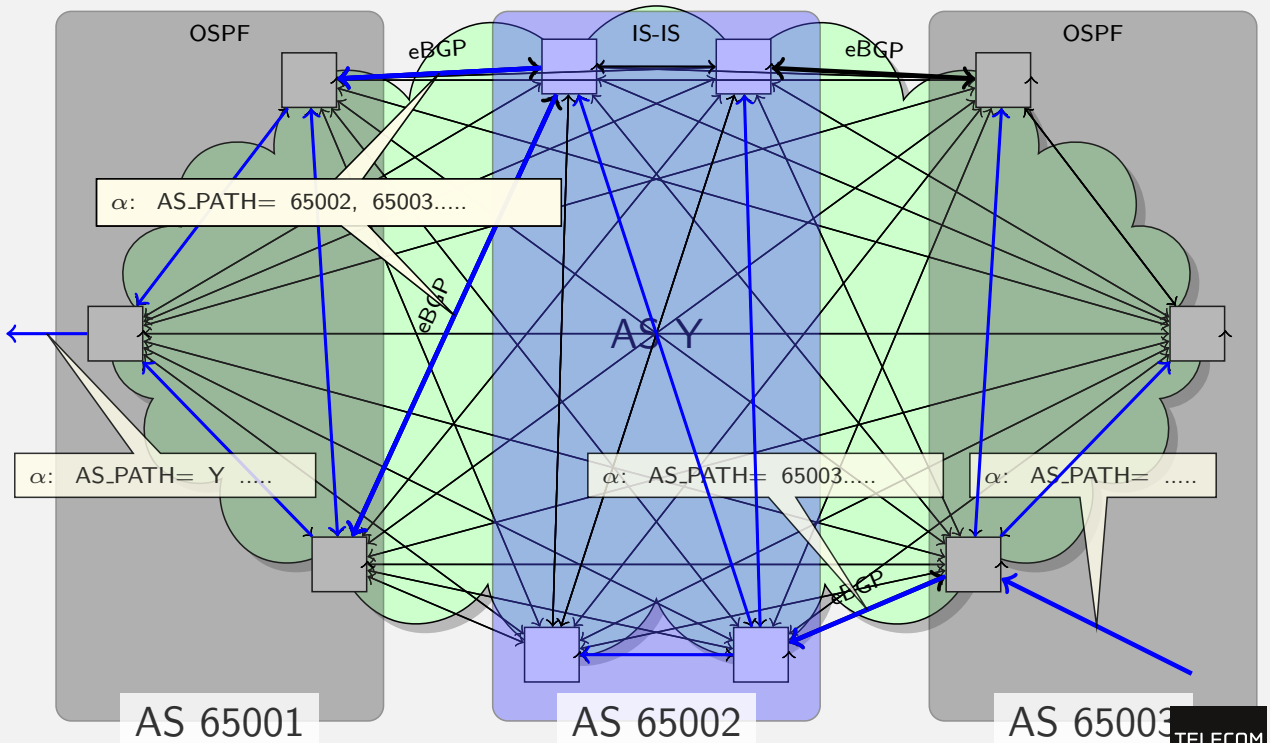
BGP

BGP for Large Networks

Confederation and Route Reflectors

- AS_PATH allows to detect loops between ASes.
- Inside an AS this is not possible since ASN is the same
- Original specifications impose a mesh network:
  - each BGP router must peer with all others
  - information learned through iBGP cannot be send to other peers inside the AS.
- The number of TCP connection is $O(n^2)$
- Confederation breaks the AS into small private ASes
- Router Reflectors (RR) modify iBGP behavior to reduce the number of peering (RFC 4456)

Confederation



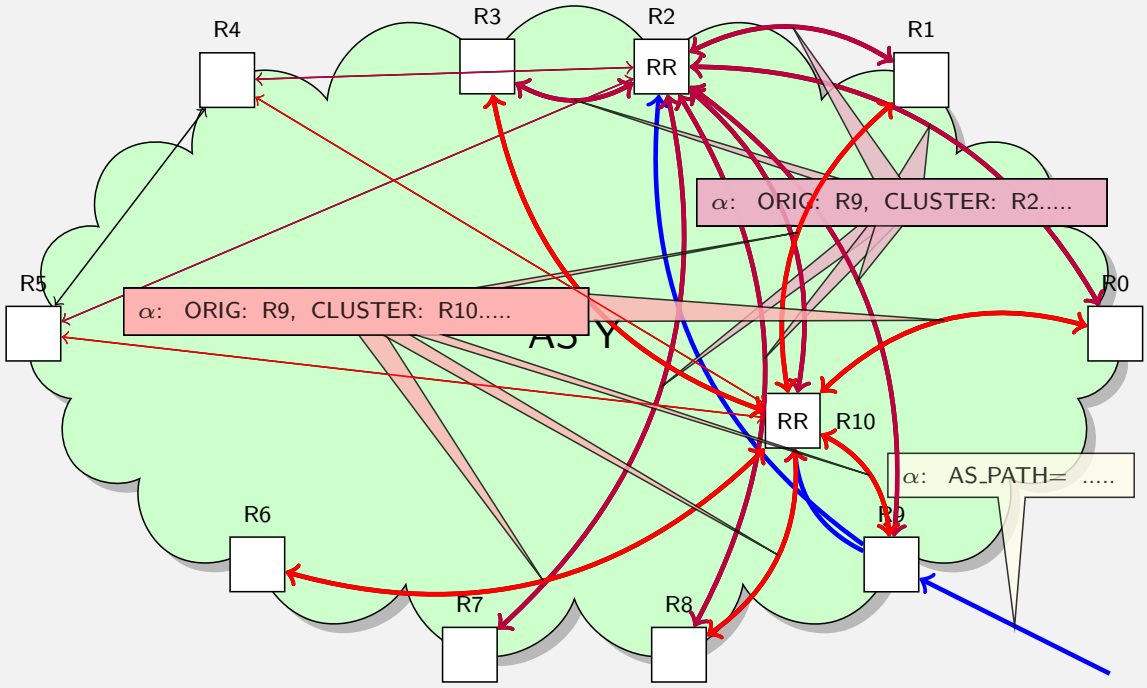
BGP attributes

1	ORIGIN	*m.w.*	RFC 4271
2	AS_PATH	*m.w.*	RFC 4271
3	NEXT_HOP	*m.w.*	RFC 4271
4	MULTI_EXIT_DISC	*o.$\bar{t}$.*	RFC 4271
5	LOCAL_PREF	*w.o.*	RFC 4271
6	ATOMIC_AGGREGATE	*w.o.*	RFC 4271
7	AGGREGATOR	*w.t.*	RFC 4271
8	COMMUNITY	*o.t.*	RFC 1997
9	ORIGINATOR_ID	*o.$\bar{t}$.*	RFC 4456
10	CLUSTER_LIST	*o.$\bar{t}$.*	RFC 4456
14	MP_REACH_NLRI	*o.$\bar{t}$.*	RFC 4760
15	MP_UNREACH_NLRI	*o.$\bar{t}$.*	RFC 4760
17	AS4_PATH	*o.t.*	RFC 4893
18	AS4_AGGREGATOR	*o.t.*	RFC 4893

Route Reflector RFC 4456

- With RR, iBGP becomes transitive:
  - iBGP announcements received are sent to other peers, ...
  - ... if the announcement is better
- Several RR can be deployed in an AS.
- To avoid loops, two new attributes are defined:
  - ORIGINATOR_ID: Orign of the route in the local AS
  - CLUSTER_LIST: List of RR ID (used as AS_PATH to avoid loops)
- RR may modify BGP, especially if IGP metric is taken into account to select routes

Route Reflectors

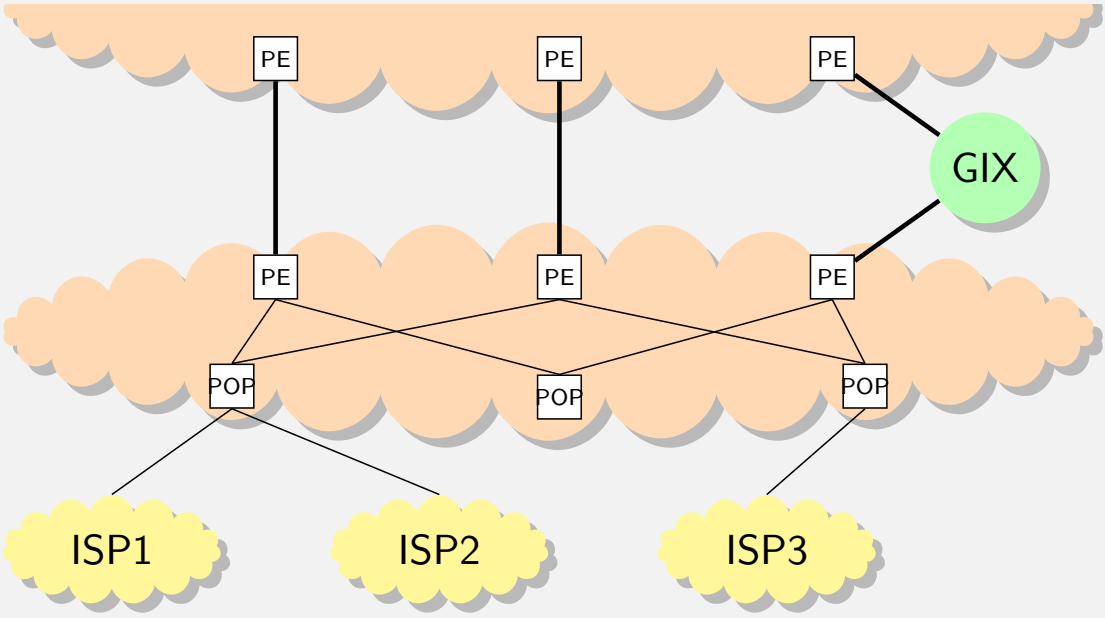


TELECOM Bretagne

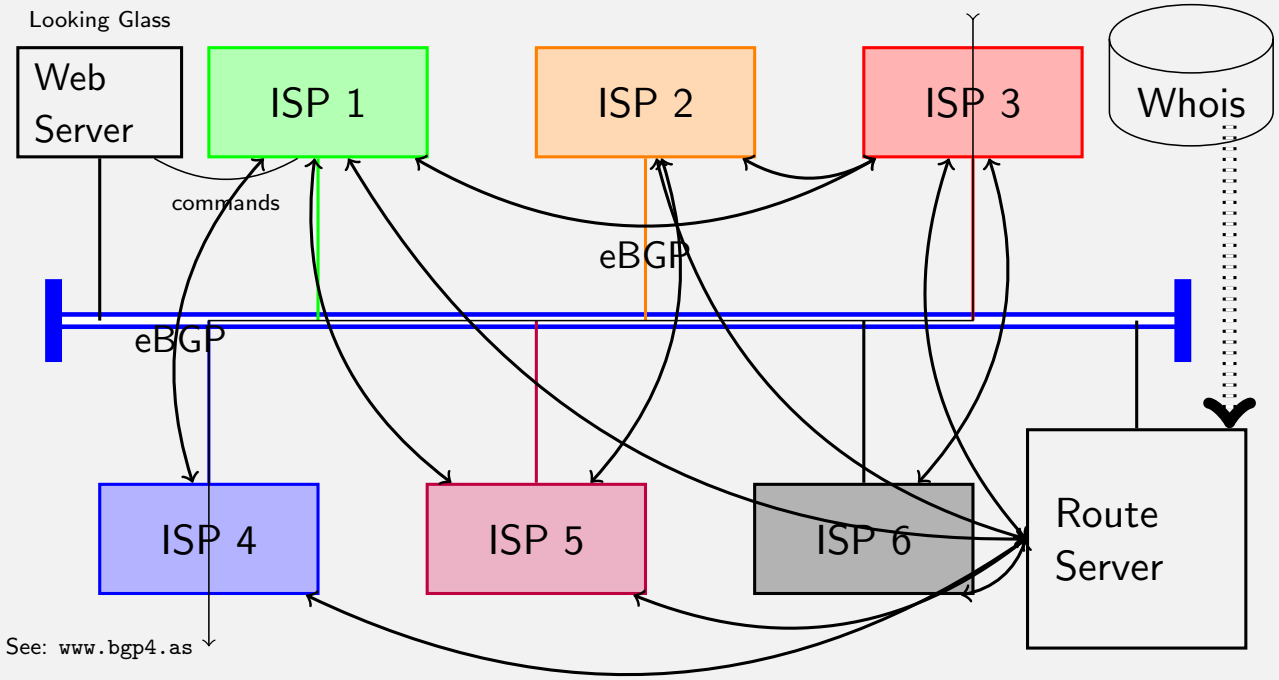
BGP

Network Architecture

Network Architecture



Global Inter eXchange

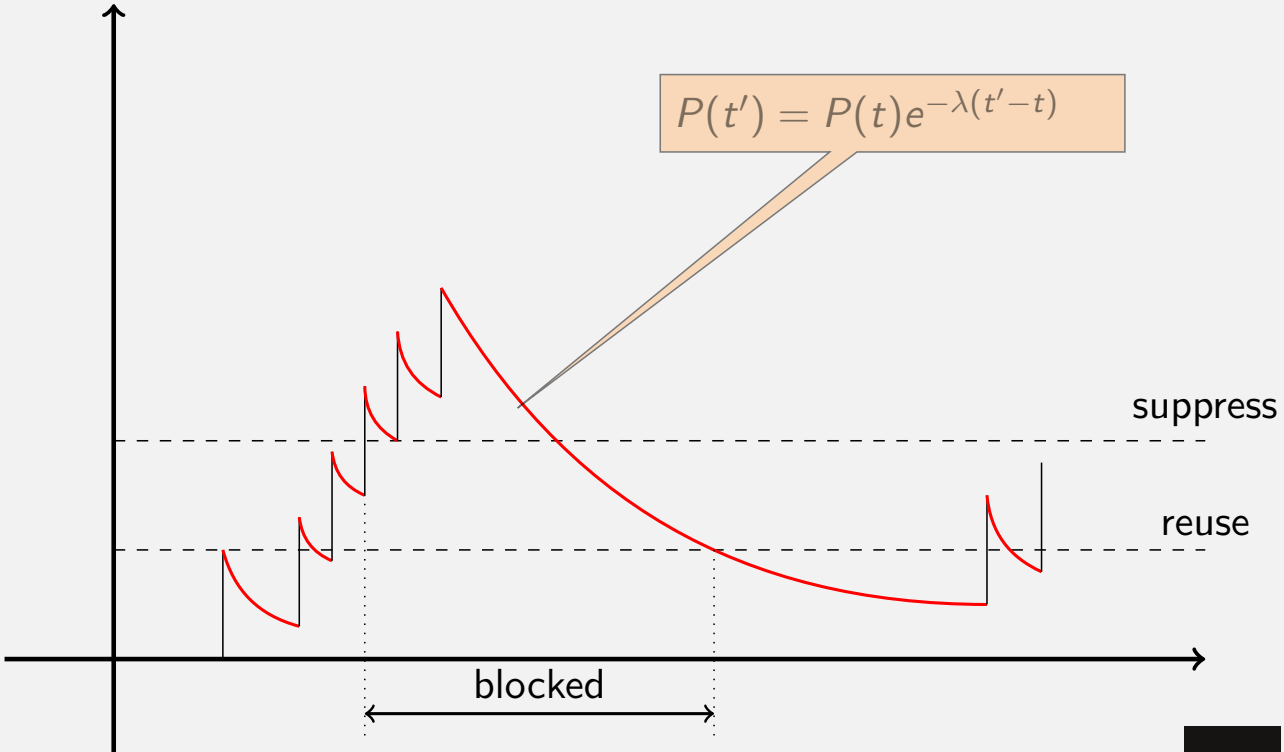


BGP Limits: Flapping

- in the 90s, some bogus implementation causes a lot of route withdraw and announcement.
  - called Flapping
  - this was propagated to all ASes, using router CPU
- Damping has been proposed (see RFC 2439)
  - When the route oscillate too much, the information is not propagated
  - Network is stabilized but,
  - network becomes unreachable
- Currently, it is not a good solution (see RIPE 378)
  - Some normal events can be amplified and lead to a route withdraw

BGP Limits: Flapping

BGP ▶ Network Architecture



BGP Limits: Security

- Provider exchange BGP announces
- It is impossible to control all announced prefixes
  - bogon/martian can be announced by some servers
- but a bad provider may inject a legal prefix
- see youtube attack:
  - http://www.ripe.net/news/study-youtube-hijacking.html
- IETF working group sidr studies the possibility to use PKI to authenticate NLRI origin.

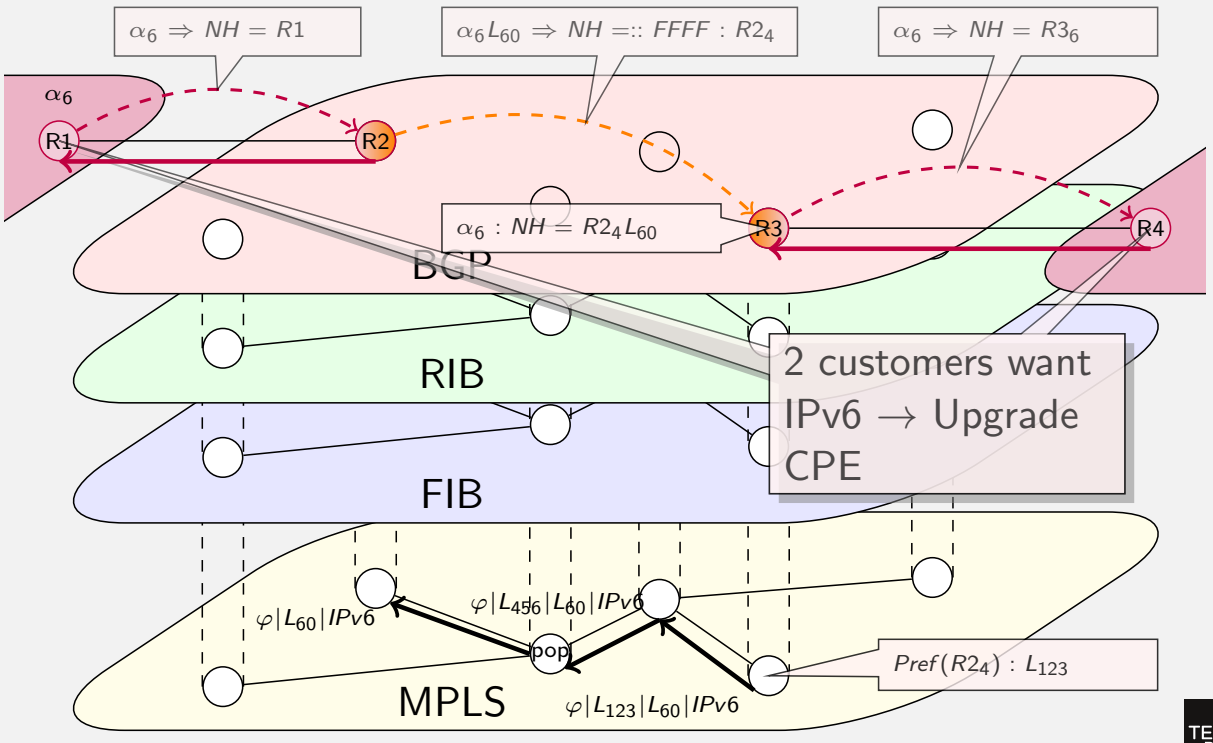
BGP

VPN

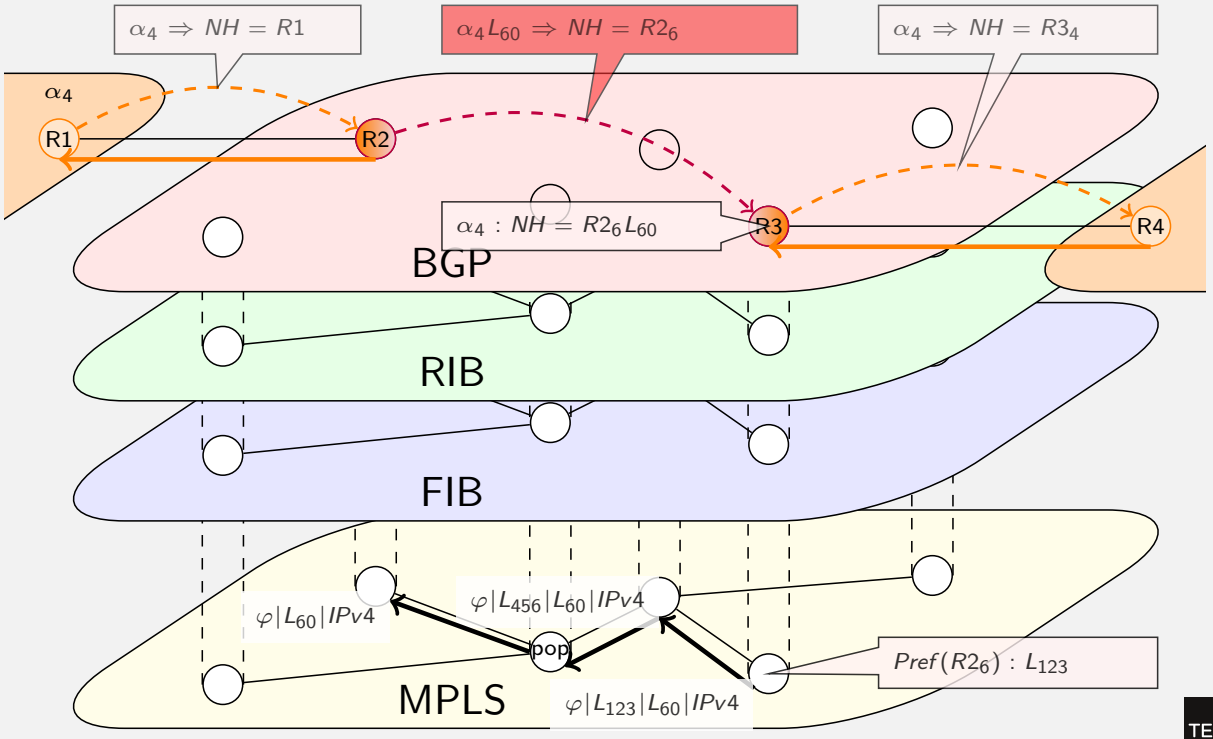
VPN

- BGP can isolate routing inside and outside an AS
  - only PE have to know both world prefixes
  - P know only inside route
- use to limit the number of entries in routing table
- Protocol family inside and outside can be different
  - private IPv4 — IPv4
  - IPv6 — IPv4
  - IPv4 — IPv6
- MPLS is used:
  - L3 VPN: used to route packets other a provider infrastructure
  - L2 VPN: used to bridge frames other a provider infrastructure
  - L1 VPN : use GMPLS to activate optical circuit on provider infrastructure

6PE



Softwires Mesh



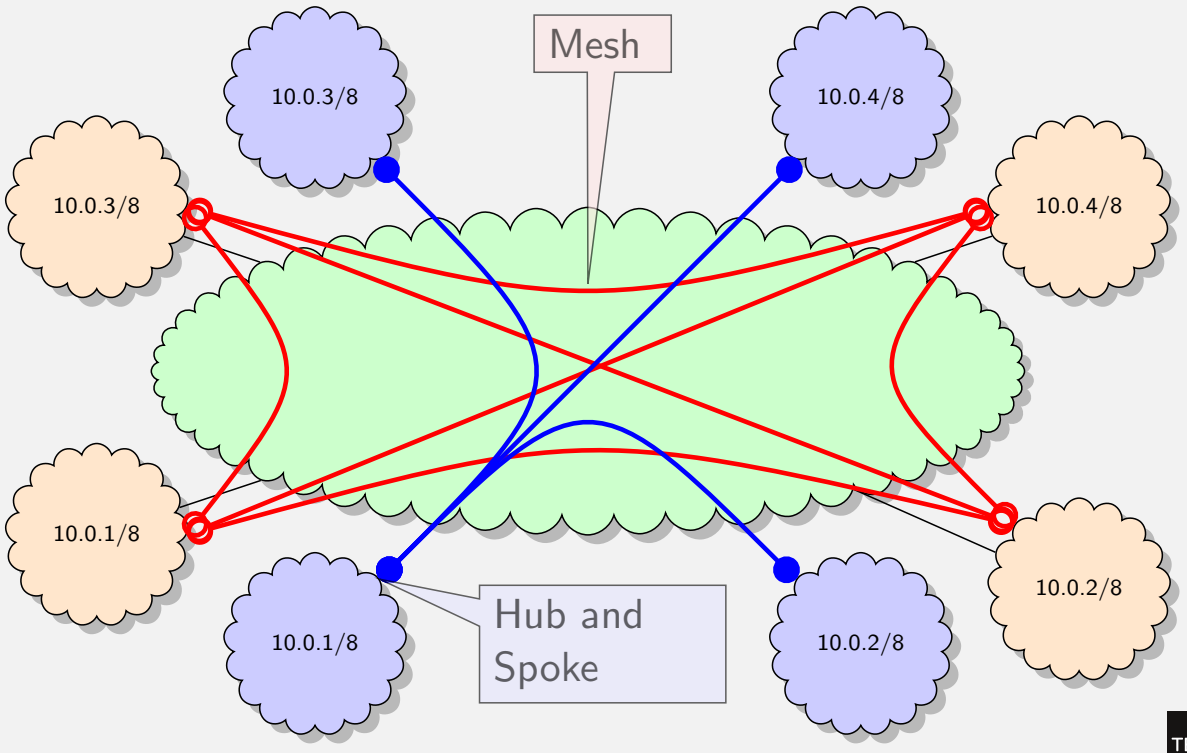
6PE versus Softwires Mesh

- MP-BGP: (RFC 4760)

  `The Network Layer protocol associated with the Network Address of the Next Hop is identified by a combination of <AFI, SAFI> carried in the attribute.`

- no AFI/SAFI defined for 6PE and Softwires
  - 6PE:
    - NLRI is IPv6
    - NH is IPv4
    - use IPv4 mapped addresses
  - Softwires Mesh:
    - NLRI is IPv4
    - NH is IPv6
    - Change the MP-BGP RFC (RFC 5549)

L3 VPN

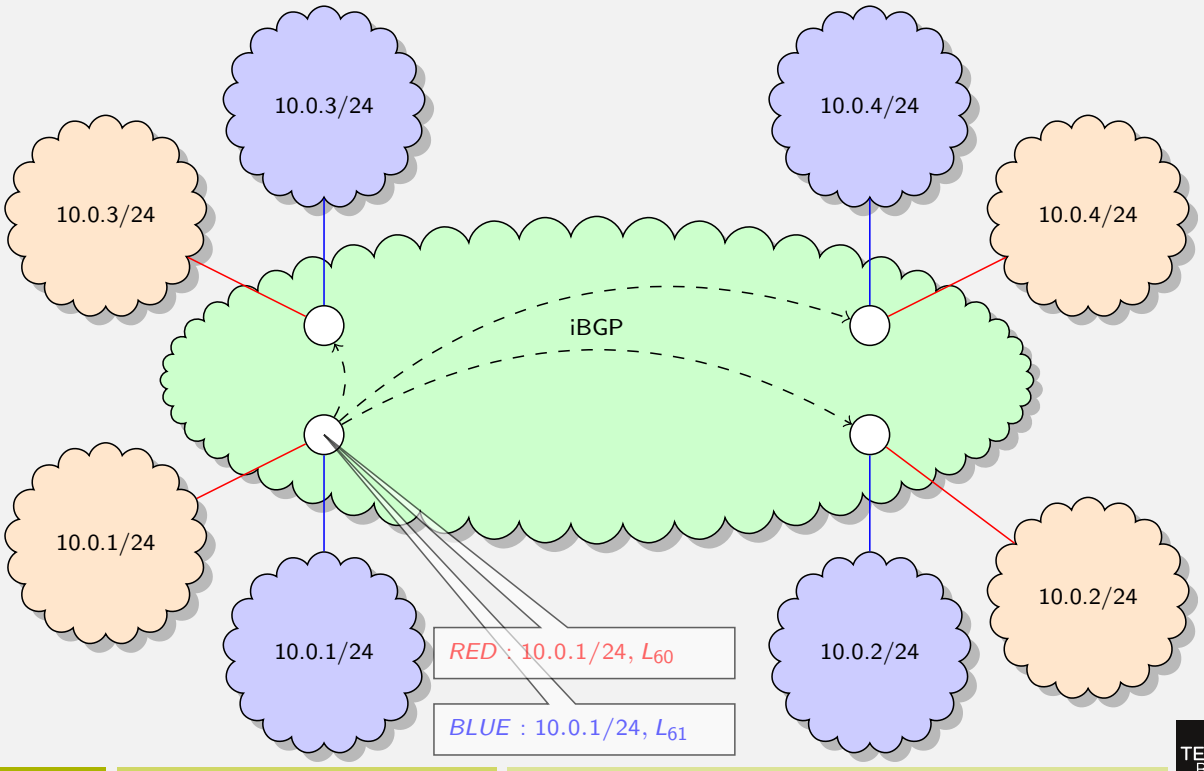


Customer VPN

- Based on tunneling:
  - IP over iP (protocol 4 or 41)
  - IPsec
  - GRE
  - L2TP
- Topology is chosen by the customer:
  - Full Mesh: a tunnel to every other locations
  - Hub and Spoke: All traffic is sent to a central site
  - any other intermediate topology
- VPN can also be managed by the provider (RFC 4364):
  - simplicity for companies
  - traffic cannot leak from one company to the other, even is the same private address space is used.

- Route Distinguishers:
  - used be the provider to identify private addresses.
  - coded on 8 bytes (2 bytes for type)
    - type 0: public Autonomous System Number (2 bytes) + customer value (4 bytes)
    - type 1: public IPv4 address (4 bytes) + customer value (2 bytes)
    - type 2: public Autonomous System Number (4 bytes) + customer value (2 bytes)
  - sent in NLRI with prefix and MPLS Label

L3 VPN



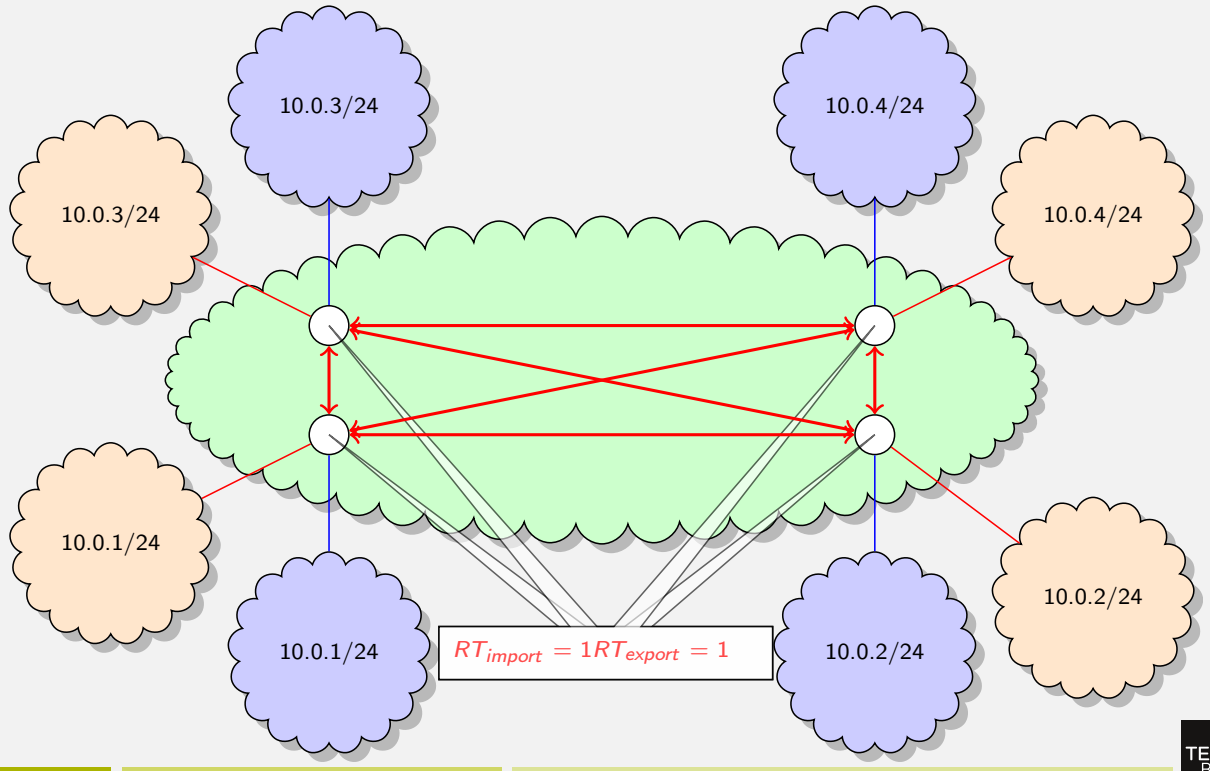
L3 VPN: Topology control

- each BGP router must tag information with a value (export) and be configure to accept some values (import).
  - if export value = import value ⇒ Full Mesh
  - if all except one export on a value and one import this value ⇒ Hub and Spoke
- route target (import/export) is carried in Community Attribute

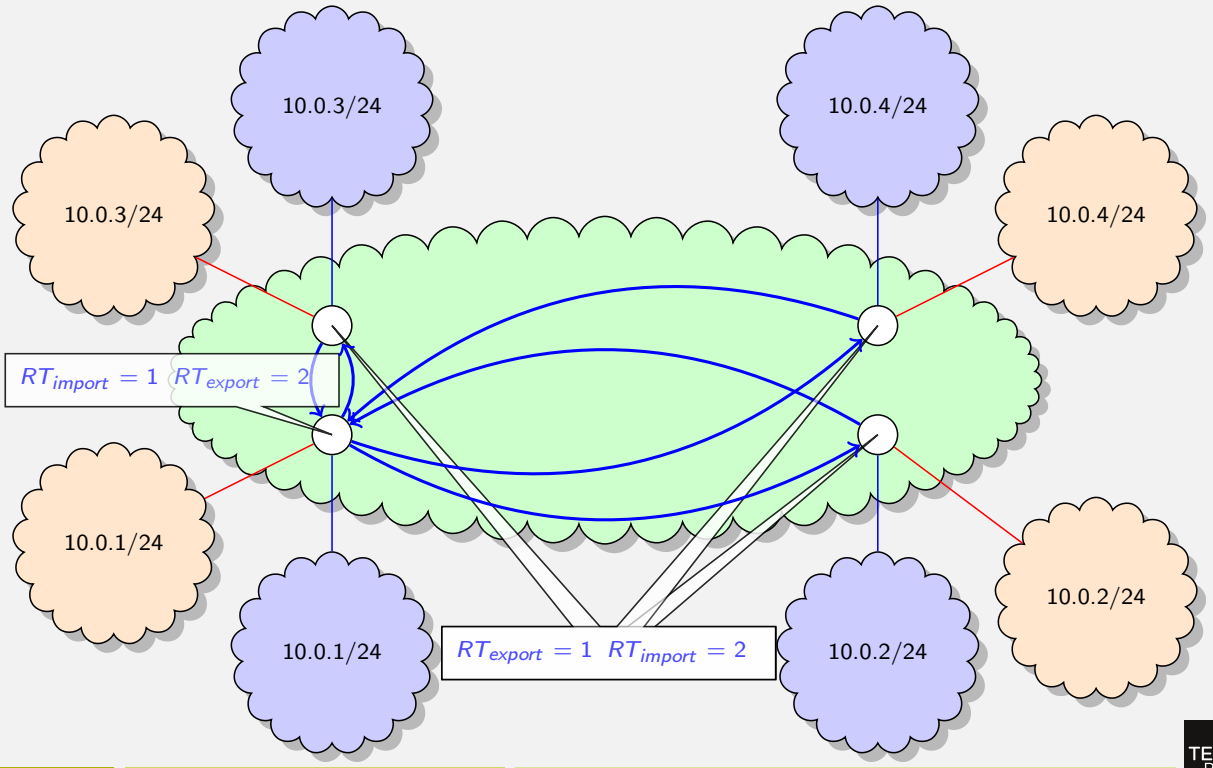
Routing Protocol

A routing protocol (IGP) MUST be installed on VPN to build connectivity inside the company.

L3 VPN


10.0.3/24
10.0.4/24
10.0.3/24
10.0.4/24
10.0.1/24
10.0.2/24
10.0.1/24
10.0.2/24
$RT_{import} = 1 RT_{export} = 1$

L3 VPN


10.0.3/24
10.0.4/24
10.0.3/24
10.0.4/24
$RT_{import} = 1$ $RT_{export} = 2$
10.0.1/24
10.0.2/24
10.0.1/24
10.0.2/24
$RT_{export} = 1$ $RT_{import} = 2$

References

- BGP Tutorial: `http://www.cl.cam.ac.uk/~tgg22/talks/BGP_TUTORIAL_ICNP_2002.ppt`

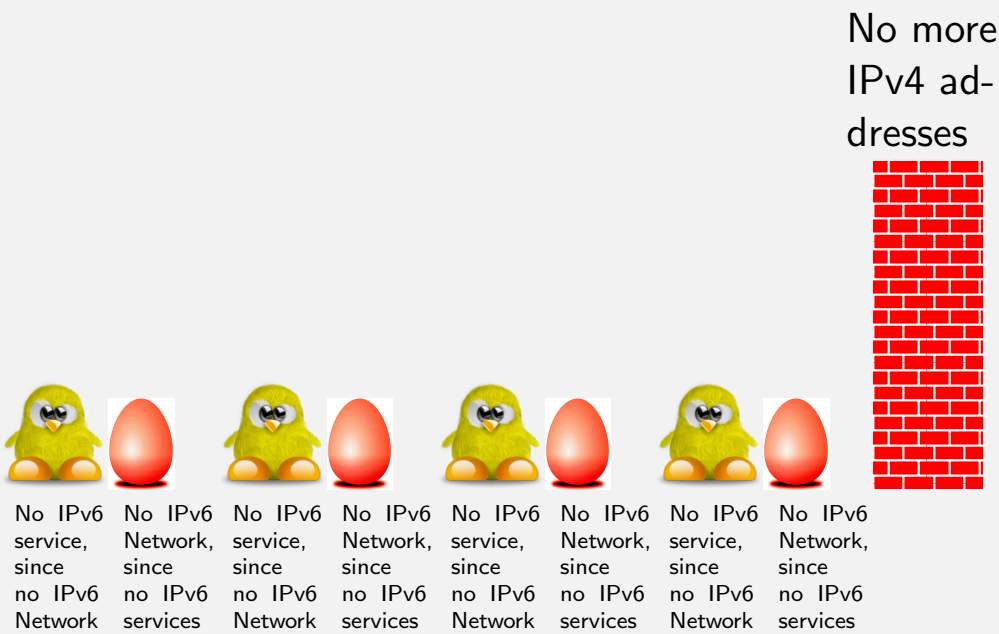
Integration

Why IPv6 Integration ?

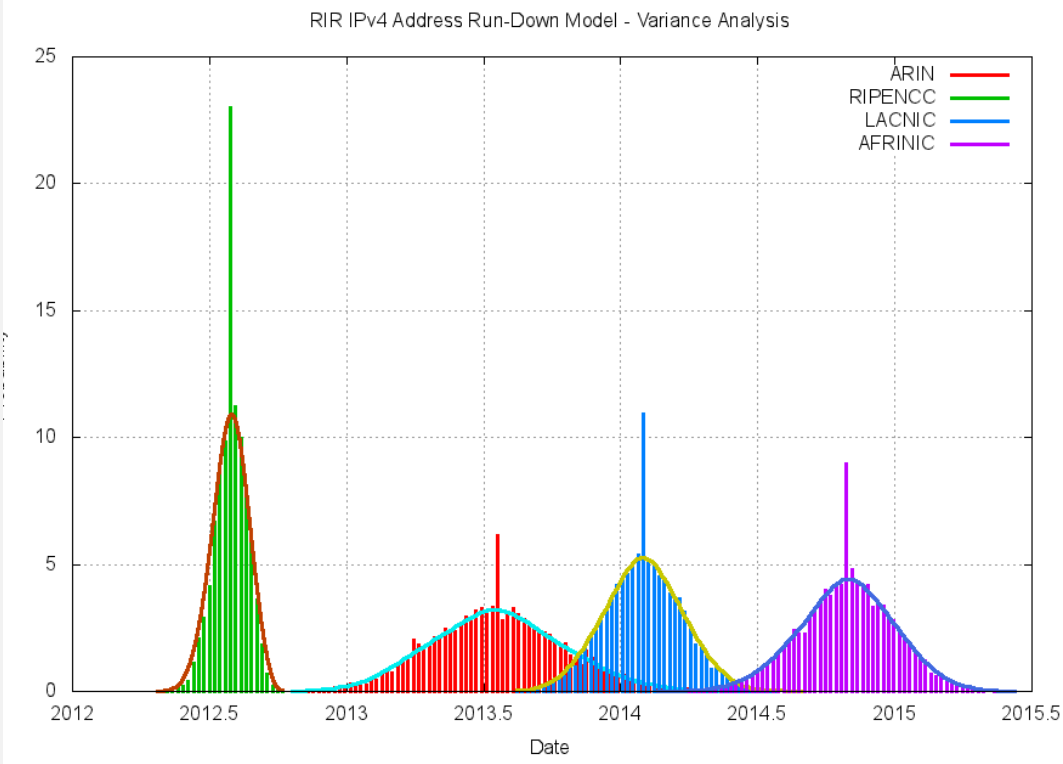
Why Integration?

- IPv4 and IPv6 are incompatible
  - Different packet format
  - Prefixes are different
- No backward compatibility, but management is very similar.
- IETF planned to deploy IPv6 then make IPv4 disappeared
  - but Metcalf's law was on IPv4 side.
  - Content on IPv4, so few actors moved.
  - Not a complete chain so access is difficult.
- Some Integration mechanisms are dangerous

Chicken Egg Problem ?



Where is IPv4?



Source http://www.potaroo.net/tools/ipv4/

TELECOM Bretagne

Not completely true

- OSes have integrated IPv6
  - Window 7, iOS, Linux,...
- Some applications are compatible with IPv6
  - see W http://en.wikipedia.org/wiki/Comparison_of_IPv6_application_support
- Routers have integrated IPv6
  - Cisco, Juniper, ALU,...
- but the chain is not complete, so IPv6 is not fully available
- An address is not only used to forward packet
  - Allocation procedures
  - Management (size is different)
  - ...
- IPv6 is new. Test products before production!

TELECOM Bretagne

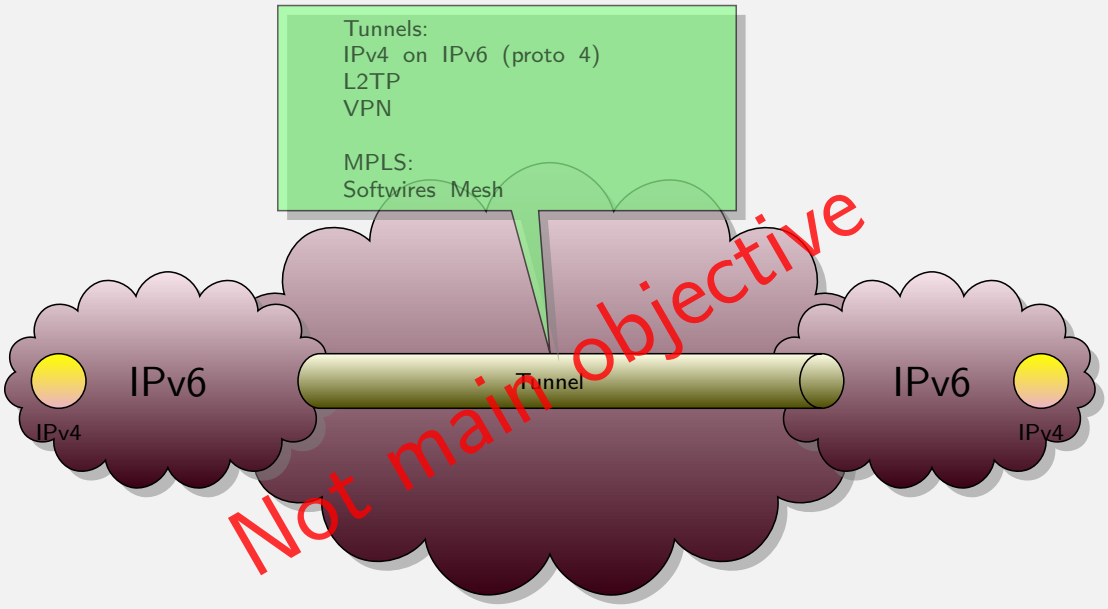
Integration

6 generic scenarios

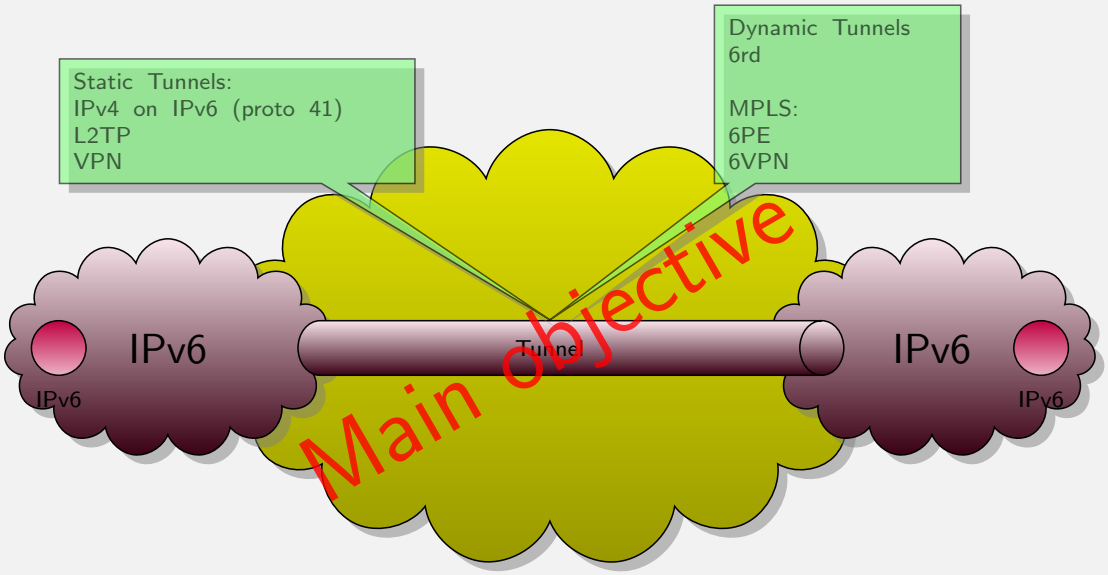
An IPv4 system connects to an IPv4 system through an IPv4 network

An IPv6 system connects to an IPv6 system through an IPv6 network

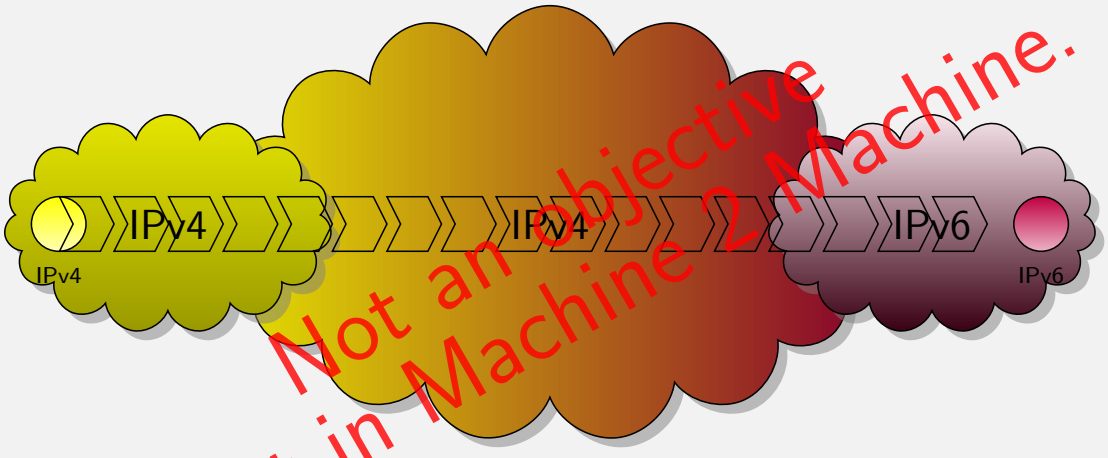
An IPv4 system connects to an IPv4 system through an IPv6 network



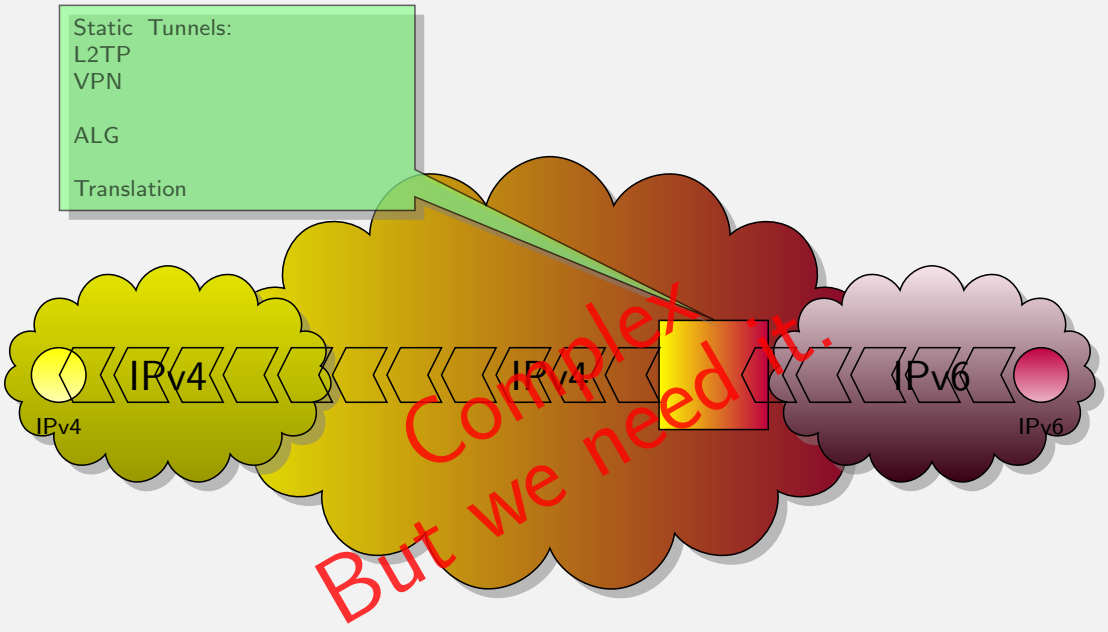
An IPv6 system connects to an IPv6 system through an IPv4 network



An IPv4 system connects to an IPv6 system



An IPv6 system connects to an IPv4 system



Integration

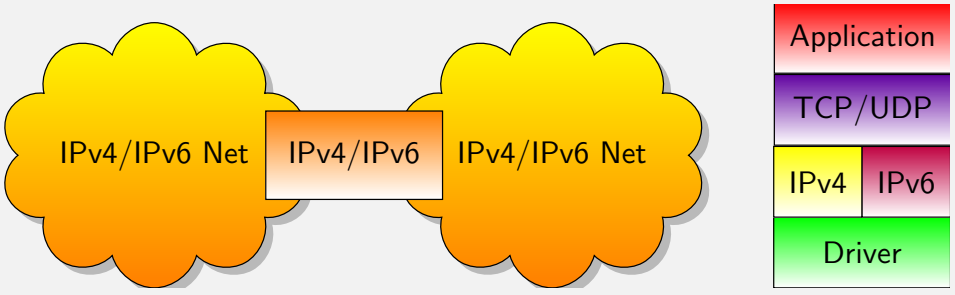
Tools overview

Rough Classification of Transition/Integration Mechanisms

- v6-v6 or v4-v4 Communication
  - Dual-Stack: v4 and v6 are fully available end-to-end
- Tunneling
  - v4 communication through a v6 network or vice versa
  - **automatic** vs **configured** (manual) tunnels
- v4-v6 co-existence/cross-communication
  - Translation
    - Header / protocol / port (v6→v4 and v4→v6)
    - Stateless *vs* Stateful
  - Relays / Application Level Gateways (ALG)

Dual-Stack Approach (RFC 4213)

- IPv4 and IPv6 running on the same box
- Especially useful for "Legacy" (existing) networks
  - *V6-fied* (legacy) IPv4 servers can provide the same service over IPv6 transport for new IPv6-only clients (web, mail, ftp, ssh...)
  - *V6-fied* (legacy) IPv4 clients can query new IPv6-only servers

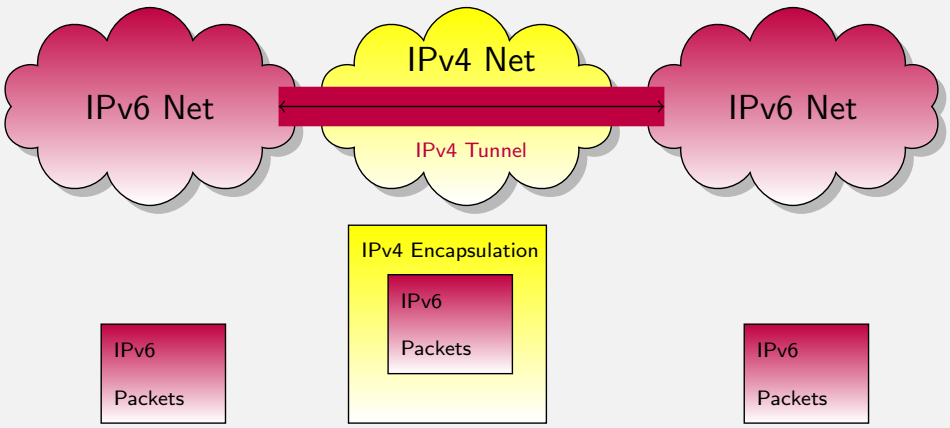


- But...
  - At least one IPv4 address is required for every node
  - ⇒ Alone, this approach does not fix the issue of IPv4 space exhaustion!
  - ⇒ Need to manage both protocols

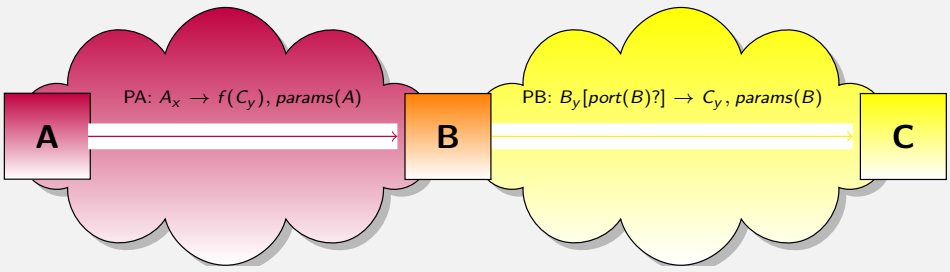
Generic Approach for "Tunneling"

2 types of tunnels:

- Automatic Tunnels
  - Examples : 6to4, Teredo, ISATAP, 6PE/MPLS...
- Configured Tunnels
  - Manual, "Tunnel Broker"
- IP on IP cannot be NATed

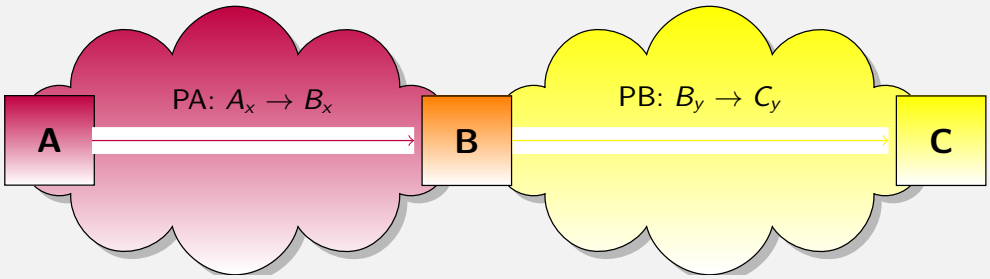


Generic Approach for "Translation"



- $(x, y) \in \{(6, 4), (4, 6)\}$
- A is IPv_x-only, C is IPv_y-only
- A sends a packet PA to C
  - Source address: A_x
  - Destination address: $C_x = f(C_y)$ (an IPv_x mapped to C_y)
- Packet PA is intercepted by B, the translation box supporting both IPv_x and IPv_y
- Packet PA is translated into packet PB, later sent to C
  - Source address: B_y from the "shared pool", potentially with a new port(B)
  - Destination address: C_y

Generic Approach for ALGs (”proxy”)



- $(x, y) \in \{(6, 4), (4, 6)\}$
- A is an IPv_x-only client; C is IPv_y-only server
- A sends to B a packet PA containing a request targeting C
  - Source address: A_x
  - Destination address: B_x
- B is a proxy supporting both IPv_x and IPv_y
- B sends to C a **new packet** PB, *proxying* A?s request
  - Source address: B_y
  - Destination address: C_y
- Examples: proxy web/ftp/DNS/mail...

TELECOM Bretagne

Integration

Scenarios

Where to act, what to do exactly?

- For ISPs/Operators
  - Backbone routers, Border routers (peering, transit)
    - Performances, Management
  - Access equipment (wired or wireless)
    - Prefix Allocation
- For users (individuals, enterprise, campus. . .):
  - LAN (routers if any)
  - Firewalls
  - Connectivity (CPE, PE)
  - Getting through their v4 ISP or bypassing it
- For everybody:
  - OS (local and distant)
  - Network applications or applications invoking the network even transiently

IPv6 is not mandatory everywhere to start Integration

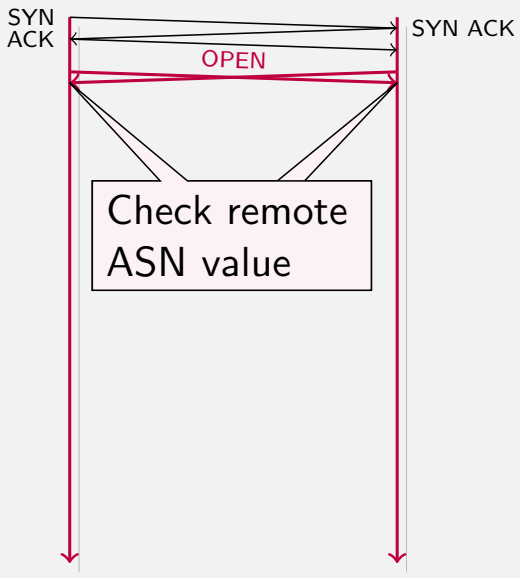
Integration

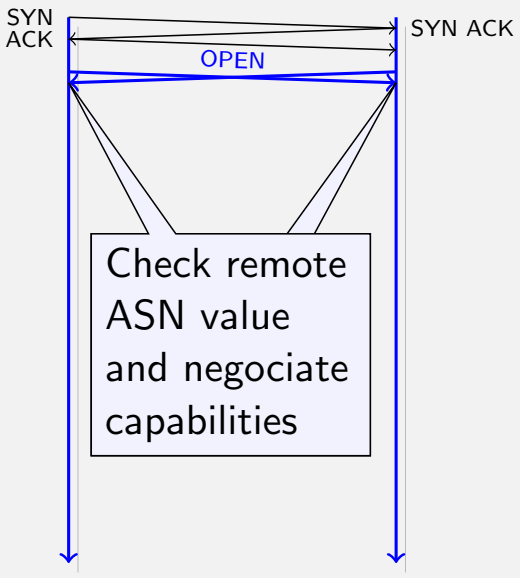
Backbone operator

Backbone operators

- Forward IPv6 as fast as IPv4
- Some old routers forward IPv6 in the supervision card
  - bad performances
- Tunnel is not a good solution
  - bad performances due to encapsulation
- MPLS is your friend.
  - L2VPN
  - 6PE
  - 6VPN
- Few have the opposite problem:
  - How to carry IPv4 traffic on an IPv6 backbone
  - Softwires mesh

BGPv4 versus MP-BGP





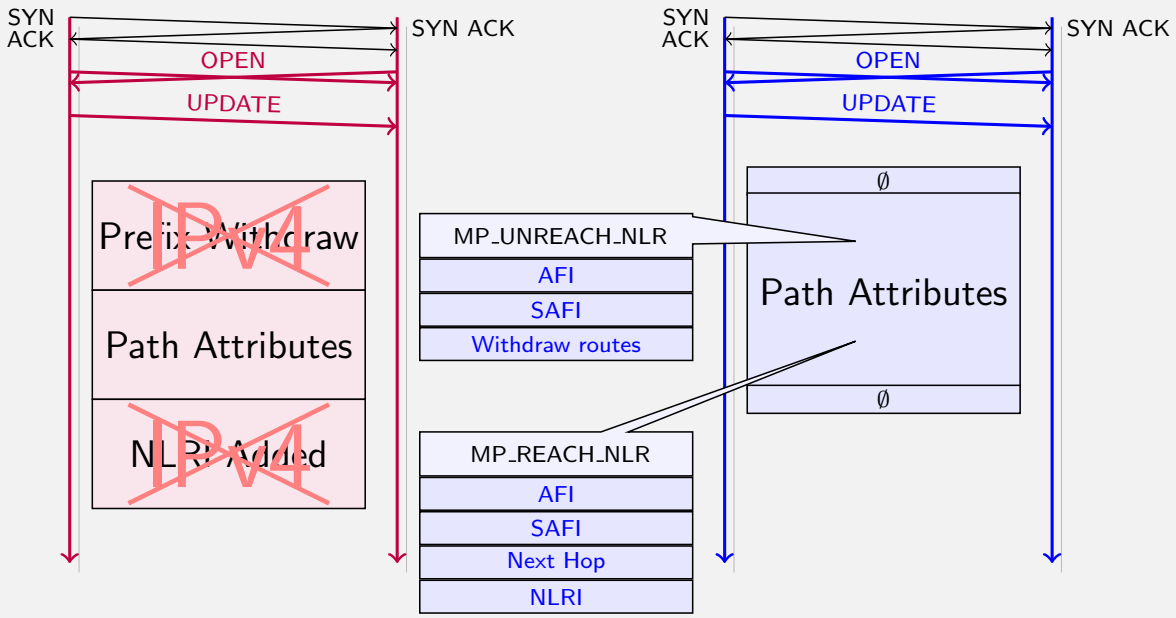
MP-BGP capabilities

- AFI : Address Family Identifier [3]
  - 1: IPv4
  - 2: IPv6
- SAFI: Subsequent Address Family Identifiers [4]
  - 1: unicast
  - 2: multicast
  - 4: MPLS
  - 65: Support for 4-octet ASN
  - 67: BGP 4over6
  - 68: BGP 6over4

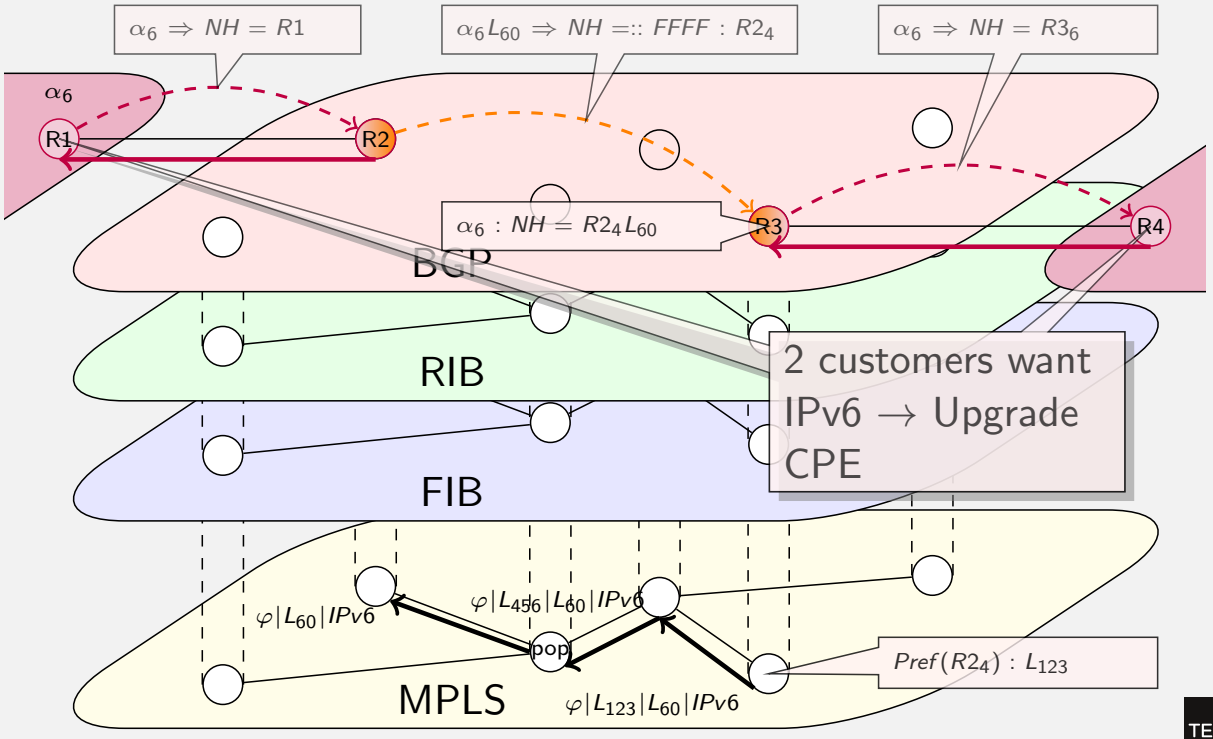
[3] http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml

[4] http://www.iana.org/assignments/safi-namespace

BGPv4 versus MP-BGP

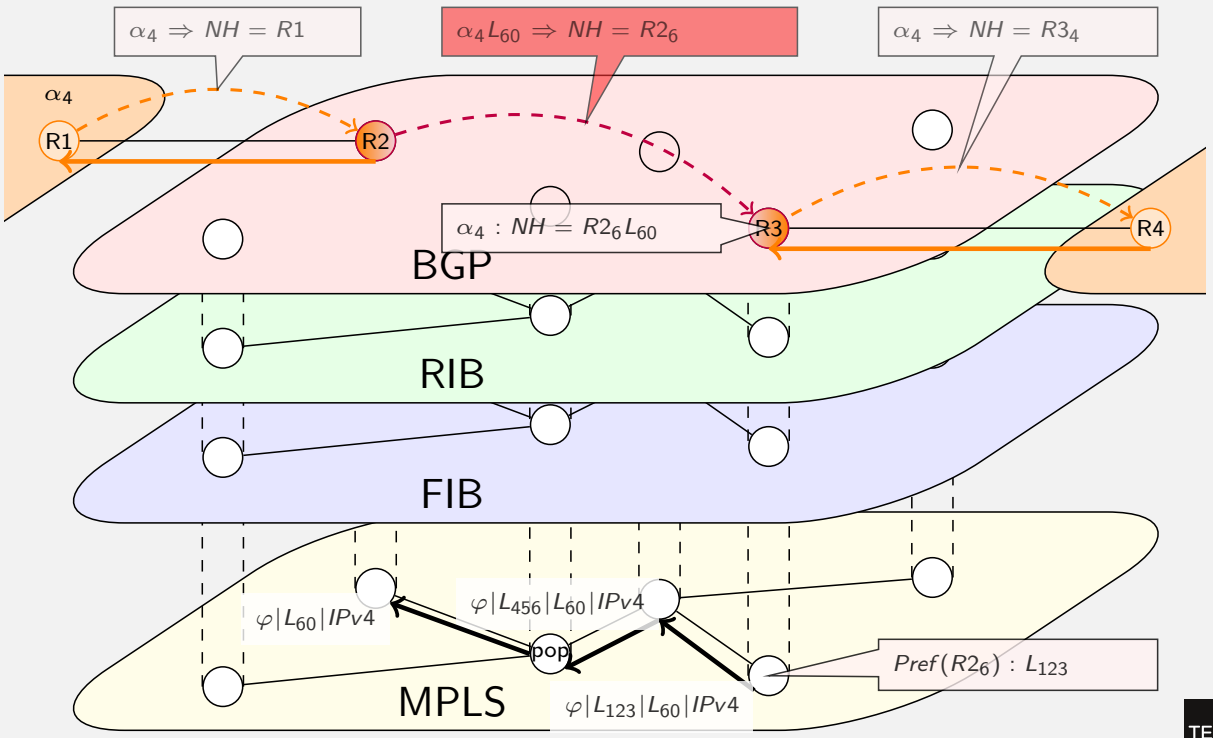


6PE



TELECOM Bretagne

Softwires Mesh



TELECOM Bretagne

6PE versus Softwires Mesh

- MP-BGP: (RFC 4760) `The Network Layer protocol associated with the Network Address of the Next Hop is identified by a combination of <AFI, SAFI> carried in the attribute.`
- no AFI/SAFI defined for 6PE and Softwires
  - 6PE:
    - NLRI is IPv6
    - NH is IPv4
    - use IPv4 mapped addresses (`::FFFF:IPv4`)
  - Softwires Mesh:
    - NLRI is IPv4
    - NH is IPv6
    - Change the MP-BGP RFC (RFC 5549)

IPv6 is here, at least at tier 1 level

- Tier 1: Sprint, Cable & Wireless, Level 3, ...
- Tier 2: France Télécom,
- GIX:

Integration

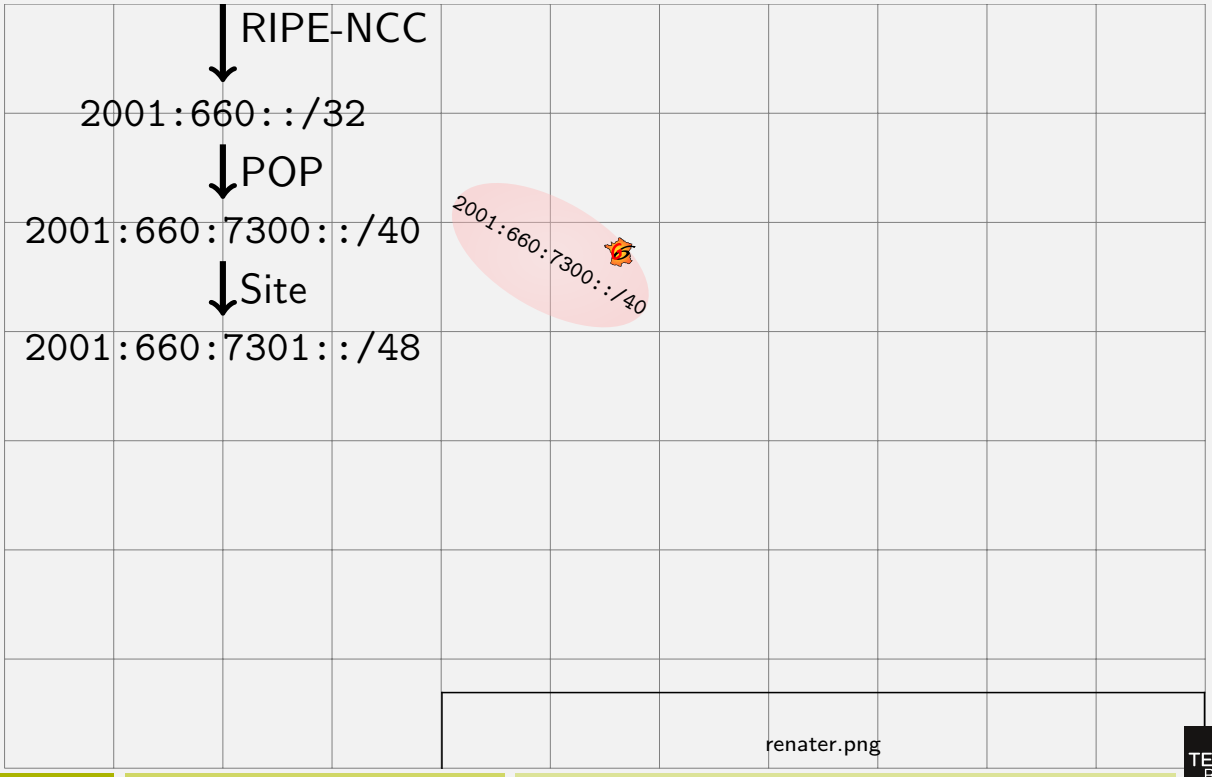
Internet Access Provider

ISP

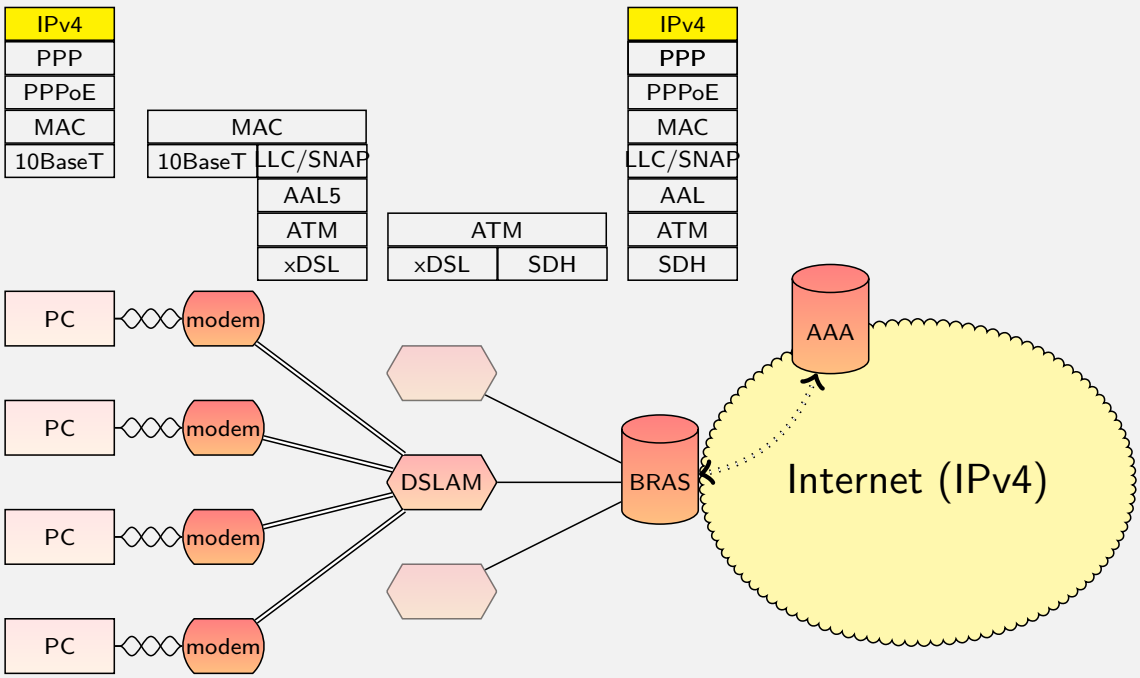
- Performances in forwarding (not so strict)
  - may use tunnels
- Allocate IPv6 prefixes
  - Lawfull IP address identification.
- May suffer from IPv4 shortage
- Different strategies exist

TELECOM Bretagne

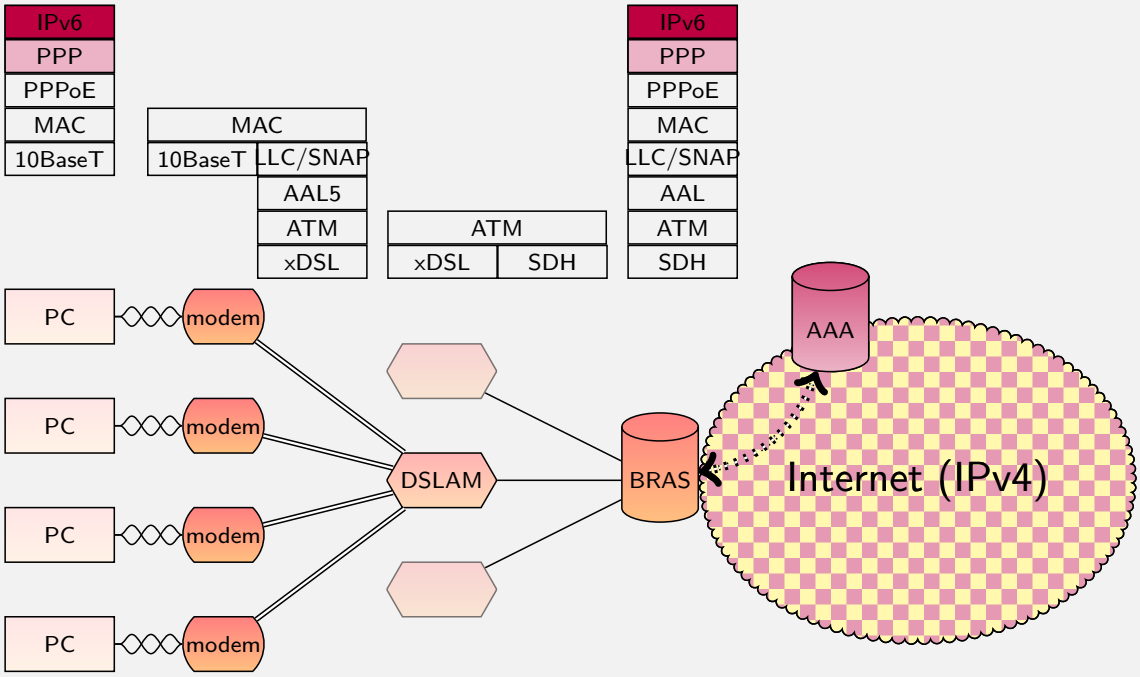
Define an addressing plan (Renater case study)



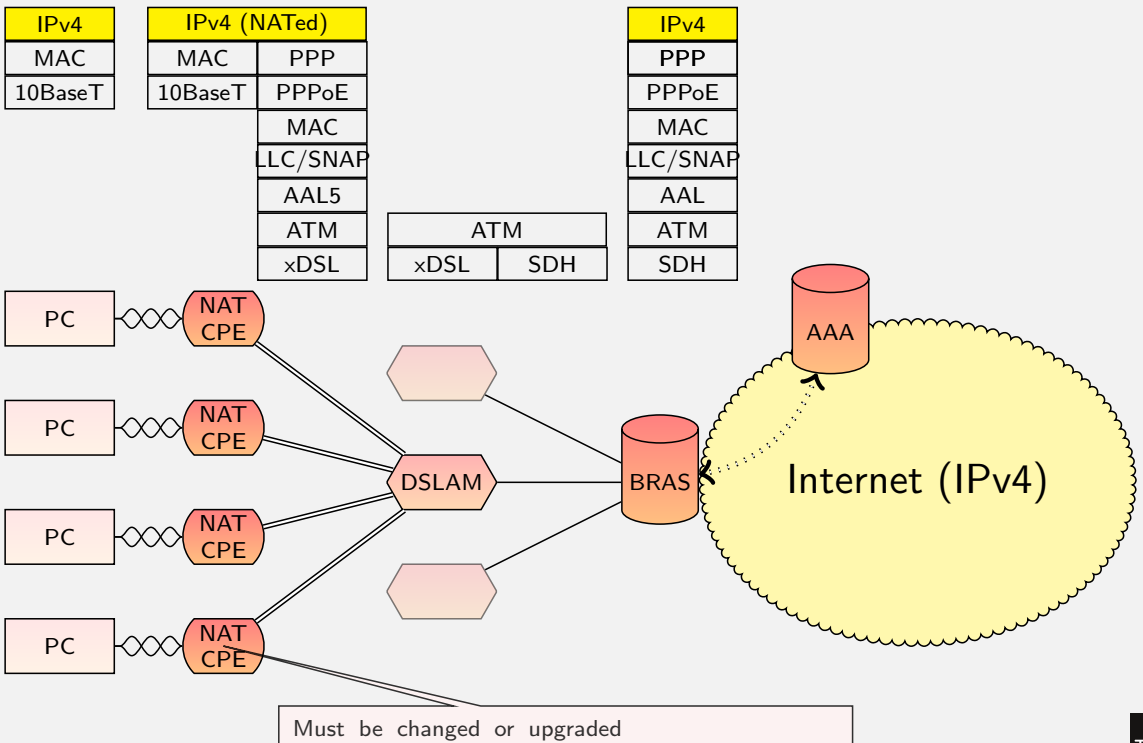
ADSL Architecture



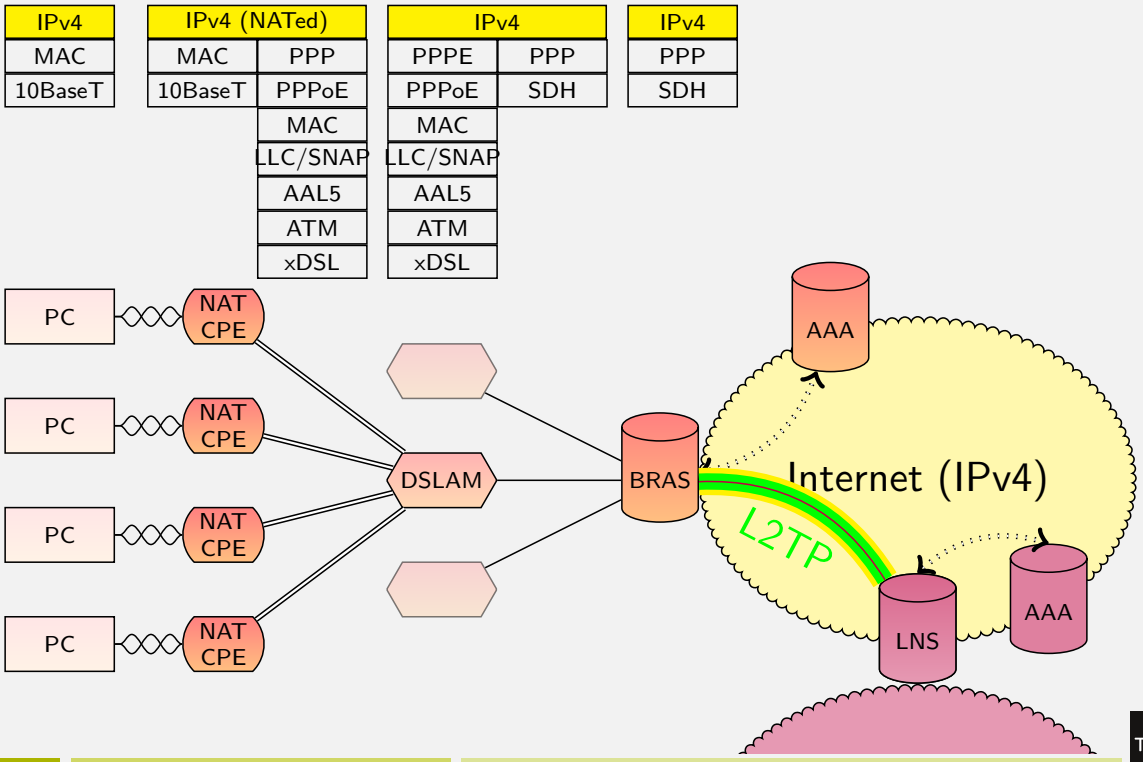
ADSL Architecture



ADSL Architecture (Box or CPE)



ADSL Architecture (3rd Generation DSLAM)



Comments I

L'intégration d'IPv6 dans les réseaux xDSL n'est pas aussi simple qu'elle peut apparaître au premier abord. En effet, basiquement un réseau ADSL est un réseau de niveau 2. Un ordinateur va utiliser l'encapsulation PPP pour transporter des trames IP vers un vers un modem ADSL qui joue le rôle de pont et transmet la trame sur le réseau téléphonique DSLAM (*Digital subscriber line access multiplexer*). A son tour, le DSLAM se contente de ponter et de multiplexer les trafics vers un routeur B-RAS (*Broadband Remote Access Server*). Pour que l'ordinateur ait accès à IPv6, il faut bien entendu qu'il ait une pile IPv6 et que PPP l'intègre et à l'autre extrémité, il faut que le B-RAS soit également compatible avec cette version du protocole et et que le réseau de l'opérateur soit également IPv6.
Même dans ce cas simple, il faut pourvoir intégrer les fonctionnalité de AAA pour authentifier les utilisateurs et configurer son équipement. En IPv4, tout passe par PPP. L'ordinateur de l'utilisateur répond à un challenge envoyé par le B-RAS. Ce dernier interroge un serveur AAA pour savoir si l'authentification est correcte. Dans un second temps, toujours via PPP, l'ordinateur est configuré avec une adresse IPv4 et généralement l'adresse du résolveur de nom pour le DNS. En IPv6, PPP après l'authentification ne configure que les adresses Lien-Local. Il faut donc que le B-RAS affecte un préfixe, via DHCPv6, à l'utilisateur dans lequel il auto-configurera son adresse IPv6. Le serveur peut retourner le préfixe a attribuer à l'utilisateur pour garantir un stabilité dans son adressage (RFC 4818).

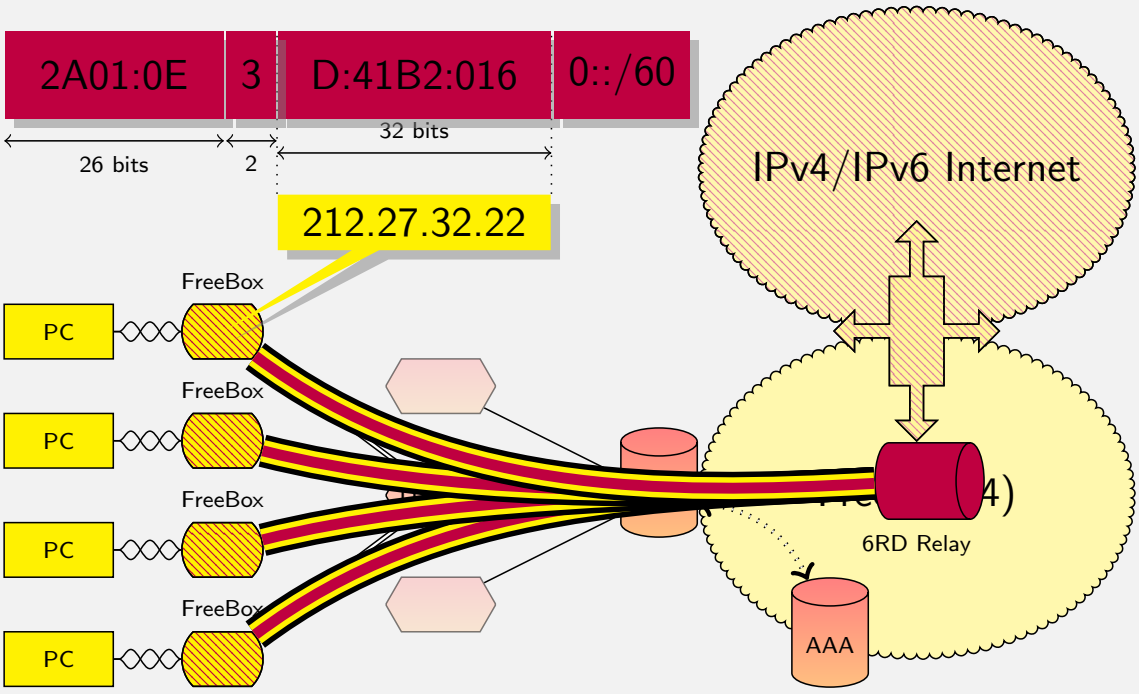
Mais en réalité, l'architecture est plus complexe. Tout d'abord l'ordinateur de l'utilisateur est derrière un CPE (inclus dans les box en France) qui contient des fonctions de NAT et de DHCP pour permettre à plusieurs équipements de se connecter. Il faut donc que cet équipement puisse accepter de l'IPv6, ce qui est rarement le cas. Plusieurs situations existent. Quand l'utilisateur est propriétaire de son CPE, il faut qu'il en achète un autre. S'il appartient à un opérateur (cas des box) il faut que ce dernier mette à jour le firmware.
L'utilisation de tunnel IP dans IP est délicate car il manque les numéros de port pour permettre au NAT de fonctionner.

Depuis plusieurs années, les opérateurs ont regroupé les fonctions de DSLAM et de B-RAS dans un même équipement. Cela a plusieurs avantages, en particulier de mieux optimiser la gestions de flux multicast des flux de

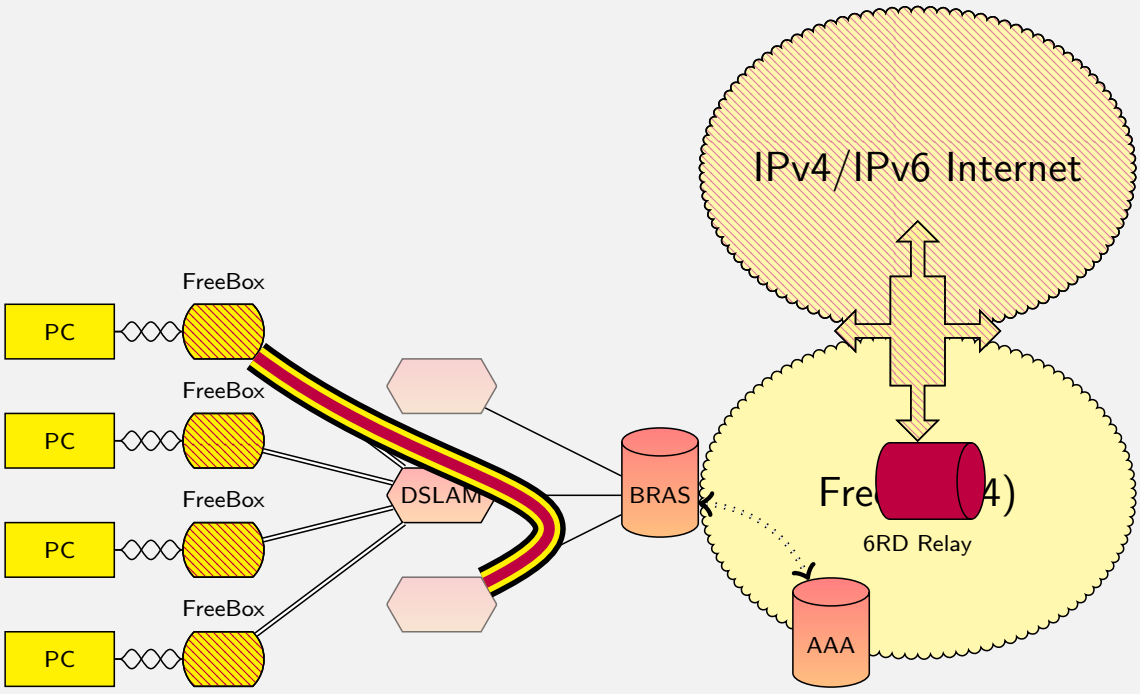
Comments II

télévision. Par contre, pour permettre de l'IPv6 natif, il faut que le DSLAM puisse le traiter. Une alternative consiste faire fonctionner le B-RAS comme un pont et envoyer les trames PPP en utilisant l'encapsulation L2TP (PPP/L2TP/UDP/IP) vers un autre routeur (appelé LAC: *L2TP Access Concentrator* sur le transparent) qui procède à l'authentification.

Free - 6rd (RFC 5969)

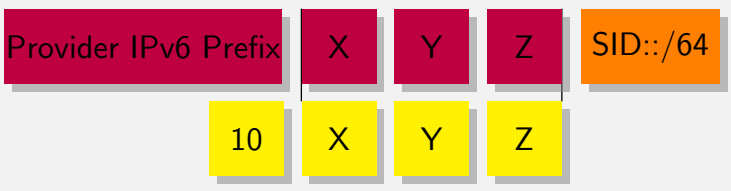


Free - 6rd (RFC 5969)



6rd

- Core network or DSLAM are not changed:
  - only some 6RD relays and CPE modification.
- IPv6 prefixes are stable if IPv4 addresses are stable
- No need to manage/log IPv6 prefixes since IPv4 prefix is embedded
- 6RD relay is not used for internal traffic
- Deployed in Free Network in 2007 in 5 weeks.
- DHCPv4 option to setup 6RD relays (6RD Relays, and prefix lengths)
- Can work with IPv4 private addresses.

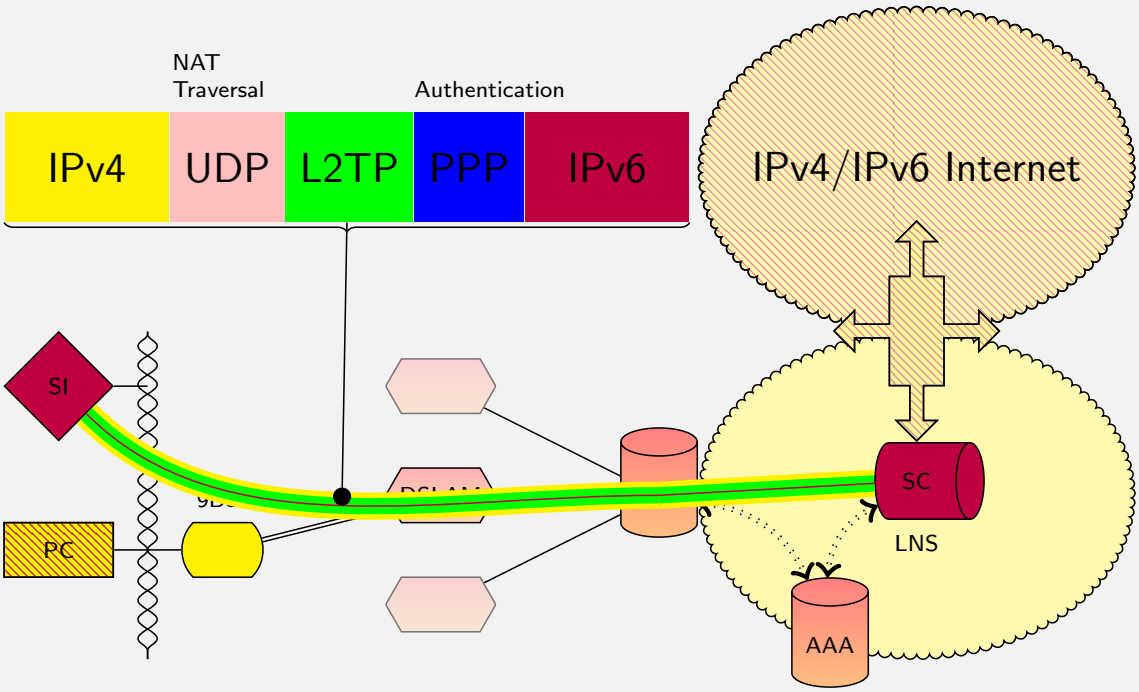


Comments I

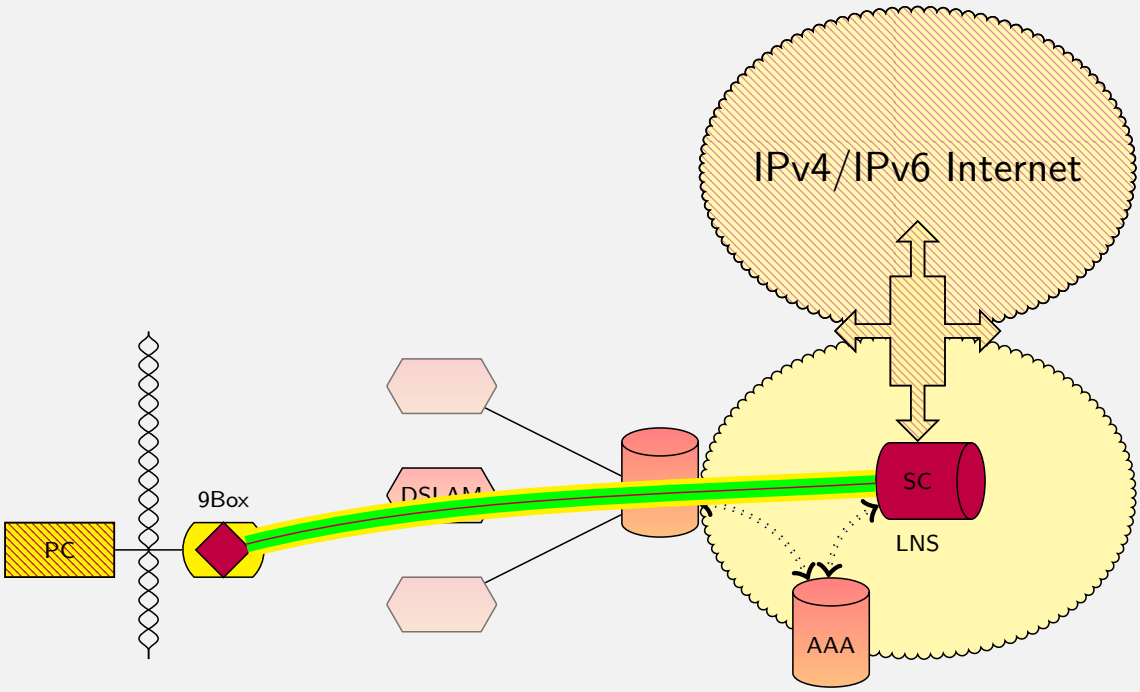
Le technologie 6RD (*Rapid Deployment*) a été introduite pour la première fois en 2007 dans le réseau de l'opérateur français Free. Sa simplicité a permis de la mettre en œuvre dans le réseau de cet opérateur en moins de 5 semaines. Elle se base sur la technologie 6to4 déjà existante que nous verrons par la suite, mais qui souffrait d'une mauvaise qualité de service.
L'opérateur met en place un tunnel qui permet de gérer IPv6 dans IPv4 (protocole 41) et doit modifier les box (CPE) de ses utilisateurs pour y introduire également une interface pour les tunnels.
Les préfixes IPv6 sont déduits des adresses IPv4 attribués à la box. L'opérateur y concatène sont préfixe IPv6.
Dans le cas de Free, le préfixe `2A01:0E00::/26` a été attribué par RIPE-NCC. Free réserve 2 bits pour avoir un `/28` qui sera plus lisible car aligné sur les chiffres du préfixe. La valeur 3 (`11` en binaire) est utilisé pour ce mécanisme.
Le préfixe de 6RD est donc `2A01:E30::/28`. On ajoute ensuite les 32 bits de l'adresse IPv4 allouée à l'interface externe de la box, on obtient donc un `/60` de la forme `2A01:E3X:XXXX:XXX0::/60`. L'utilisateur dispose donc de 4 bits pour numéroter ses SID soit 16 valeurs possibles. La Box choisit un SID et annonce normalement le préfixe sur le réseau de l'utilisateur. Les équipements qui ont activé IPv6 construisent leur adresse.
Comme l'adresse IPv6 dépend de l'adresse IPv4, il n'est pas nécessaire d'avoir des mécanismes de gestion supplémentaires pour IPv6. Ainsi, si une demande légale d'identification d'un abonné est demandée pour une adresse IPv6, il suffit de se baser sur la partie IPv4.

Le RFC 5969 prévoit une option DHCPv4 pour configurer le CPE de l'opérateur avec l'adresse des relais 6RD ainsi que les longueurs des préfixes IPv4 et IPv6. Ainsi, si l'opérateur utilise un adressage privé ou si son préfixe IPv6 est trop long, il n'est pas nécessaire de mettre l'intégralité de l'adresse IPv4 dans le préfixe 6RD, il suffit juste d'y mettre les bits correspondant à la partie variable de l'adresse IPv4.

SFR: Softwires: H&S Architecture RFC 5571



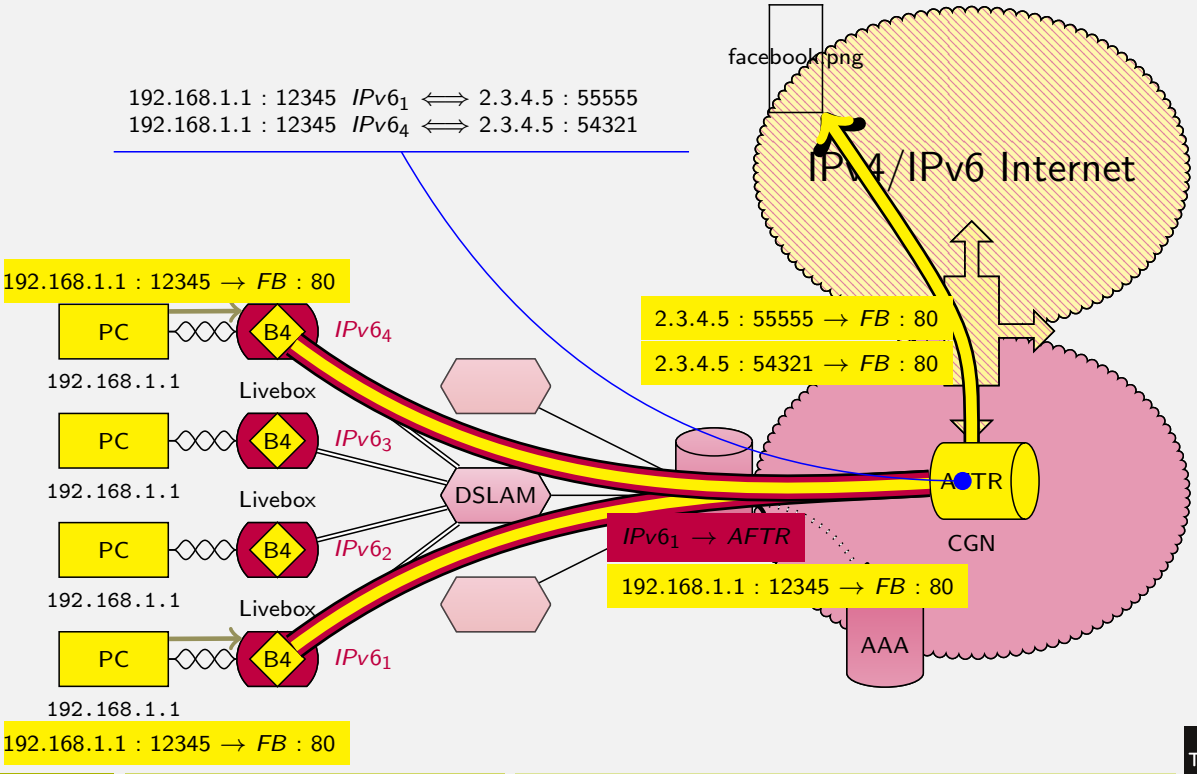
SFR: Softwires: H&S Architecture RFC 5571



Comments I

La technique Softwires Hub & Spoke utilise les tunnels L2TP. Dans la version de base, un équipement (appelé SI: *Softwires Initiator*) est mis dans le réseau local de l'utilisateur. Celui-ci contacte un concentrateur (SC: *Softwires Concentrator*). L'intérêt de cette technologie est de n'utiliser que des protocoles déjà standardisés. Le RFC 5571 définit les profiles d'utilisation. Le fait d'utiliser UDP permet de traverser les NAT. Les messages de *keepalive* de L2TP et de PPP permettent de garder les contextes NAT ouverts même lorsqu'il n'y a pas de trafic. L'utilisation de PPP permet d'authentifier l'utilisateur et donc de lui fournir toujours le même préfixe. Ainsi, si l'opérateur renumérote périodiquement la box, le tunnel L2TP tombe, mais est rapidement réouvert et le préfixe IPv6 reste le même.
Le SI peut être intégré à la box. Cela permet de traverser les DSLAM qui ne sont qu'IPv4.

France Télécom/Orange: Native + CGN



TELECOM Bretagne

France Télécom/Orange: Native + CGN

Integration ▶ Internet Access Provider

- Carrier Grade NAT deals with IPv4 address exhaustion:
  - No IPv4 address for the infrastructure
  - An IPv4 address is shared among several users
    - A user consumes about 300 port numbers
    - Less is needed (2 or 3 users per address)
- Less scalable than user NAT
  - More traffic from different users
  - for incoming traffic must map a port number to an IPv6 address
- Must take into account:
  - UPnP: Send UPnP traffic to CGN (see Port Control Protocol)
  - Static Mapping: Web page on AFTER
- Legal identification is complex:
  - Log per flow
  - Need IPv4 address, port number and time.

TELECOM Bretagne

Comments I

Cette architecture impose le déploiement d'IPv6 jusqu'à chez l'utilisateur. Le trafic IPv4 sera encapsulé dans de l'IPv6. Les CGN consistent à mettre un NAT au cœur du réseau plutôt que chez l'utilisateur. De cette manière, il est possible de partager une adresse IPv4 entre plusieurs utilisateurs. L'architecture se compose d'un équipement B4 (*Basic Bridging BroadBand*) va simplement encapsuler le trafic IPv4 sortant vers un équipement AFTR (*Address Family Transition Router*) qui effectuera la traduction de l'adresse privée en adresse publique. L'avantage de cette solution est de faire disparaître les adresses IPv4 de l'infrastructure, elles pourront être redistribuées aux clients. De plus le partage d'une adresse IPv4 par plusieurs utilisateurs permet de moins gaspiller de cette ressource rare.
Cette traduction est un peu plus complexe que dans un NAT traditionnel, car il faut associer au numéro de port sortant l'adresse IPv6 de l'équipement B4 en plus de l'adresse privée de la source et le numéro de port qu'elle a choisi. Quand un paquet revient à l'AFTR, celui-ci à partir du port destination retrouve l'adresse du B4, l'adresse privée de la machine et le numéro de port. Cette opération est relativement complexe, surtout si les débits sont relativement élevés.
Un utilisateur moyen consomme environ 300 ports (il faut prendre en compte qu'un port utilisé pour une connexion TCP n'est libéré que 2 minutes après la fermeture de la connexion). On pourrait donc arriver à un multiplexage de 200 clients par adresse IPv4. Mais ces valeurs sont irréalistes. Si un opérateur alloue la même adresse à deux utilisateurs, il double le nombre de clients.

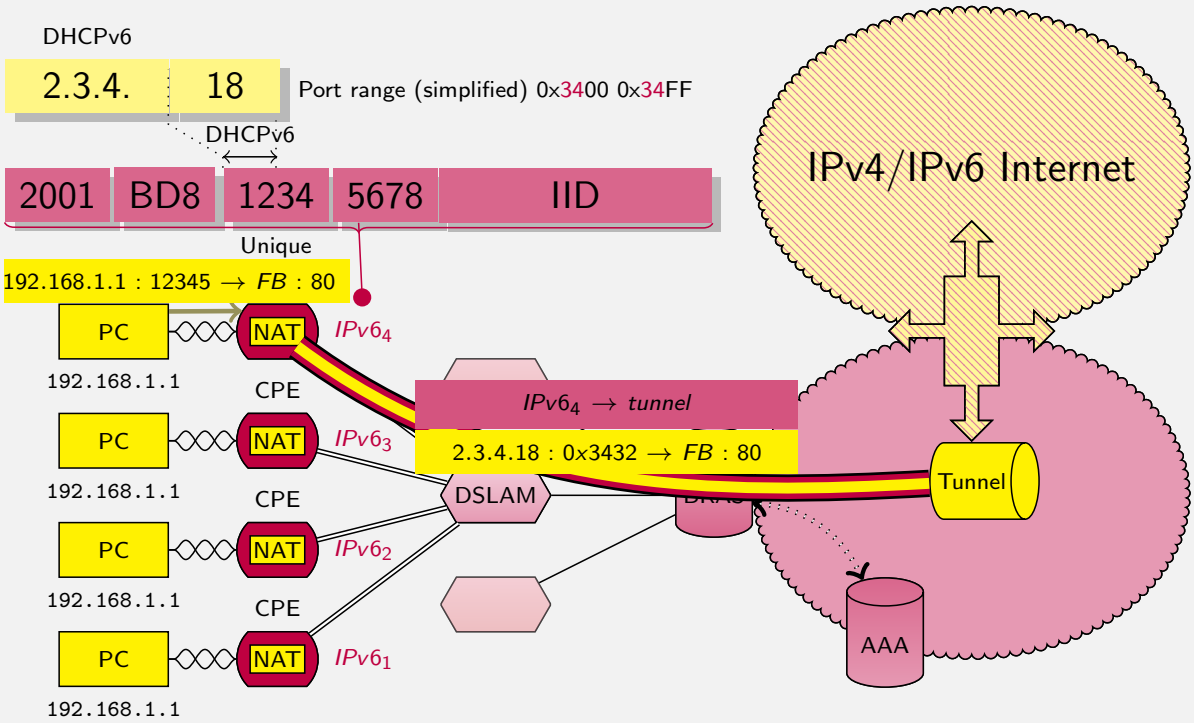
Par contre cette solution a des inconvénients. Dans les architectures UPnP très utilisées par les jeux en lignes ou des applications comme bittorrent, un message en diffusion est émis par les stations pour trouver et donner des ordres aux NAT. Comme le NAT ne se trouve plus sur le réseau local, il faut définir un protocole pour permettre aux ordres UPnP d'atteindre le CGN; Port Control Protocol est en cours de définition à l'IETF.
Un utilisateur peut vouloir mettre en place chez lui un serveur web. Déjà, il ne peut plus compter sur le port bien connu 80 pour mettre en place son service, car il sera partagé entre plusieurs utilisateurs. Il devra donc demander un autre numéro de port et le mettre dans les URL. Le CGN doit disposer d'une interface de configuration pour garantir une affectation stable des ces valeurs.

Comments II

Finalement, pour les aspects légaux, la gestion du CGN est complexe, en effet une adresse IP ne reflète plus un seul utilisateur, mais un groupe. Il faut donc connaître l'heure à laquelle le trafic a été capturé et le numéro de port utilisé pour remonter à la source et identifier l'utilisateur.

La technique CGN n'est donc qu'une étape intermédiaire, pour amener IPv6 jusqu'à l'utilisateur et doit être utilisée qu'en dernier recours quand le service n'est pas accessible en IPv6.

4rd (main idea)



Comments I

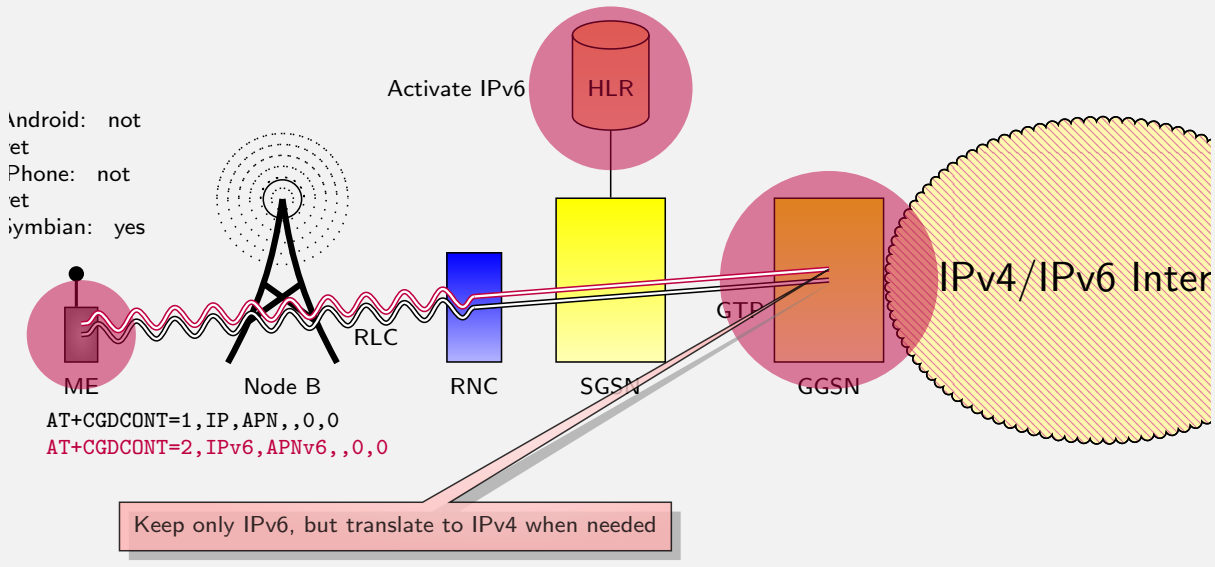
4RD (pour *Residual Deployment*) est une technologie plus jeune de CGN, toujours à l'état de draft à l'IETF, elle est plus simple à mettre en œuvre que CGN. Il s'agit de construire une adresse IPv4 à partir d'informations contenues dans un préfixe IPv6. Ainsi dans l'exemple précédent si un site reçoit le préfixe 2001:DB8:1234::/48. la partie 0x1234 est unique pour ce site (on suppose que l'opérateur dispose d'un /32). Le site aura reçu par DHCPv6 des informations lui donnant le préfixe IPv4 de base (ici 2.3.4/24) et la partie qu'il prendra de l'adresse IPv6 pour compléter l'adresse (ic 0x12, soit 18 en décimal). Le CPE contruit donc l'adresse publique du NAT 2.3.4.18. La partie 0x34 donnera le numéro des ports (en fait ces ports sont répartis sur plusieurs plages pour ne pas favoriser ou défavoriser des utilisateurs). Dans notre exemple simple, tous les ports utilisable commenceront par 0x34XX. Le NAT reste sur le CPE simplifiant l'utilisation des protocoles comme UPnP, il s'agit juste de restreindre les ports utilisables par le NAT.
On voit qu'un autre site recevant le préfixe 2001:DB8:1235::/48 utilisera la même adresse IPv4, mais pas la même plage de numéro de ports.

Ce qui est intéressant dans cette technologie, vient de la gestion des données en retour. En effet, le tunnelier est sans état. S'il reçoit un paquet IPv4 a destination de 2.3.4.18 et sur le port 0X3487, il prend la valeur 18 et le début du numéro de port et peut ainsi construire le préfixe vers lequel les paquets devront être tunnélés.

Integration

3G/LTE

3G data



ME: Mobile Equipment, **RNC**: Radio Network Controller, **SGSN**: Serving GPRS Support Node, **GGSN**: Gateway GPRS Support Node, **HLR**: Home Location Register, **GTP**: GPRS Tunnelling Protocol **RLC**: Radio Link Control

How does it work? (ETSI/3GPP TS 29.061)

donner le chronogramme des échanges pour obtenir un prefixe.

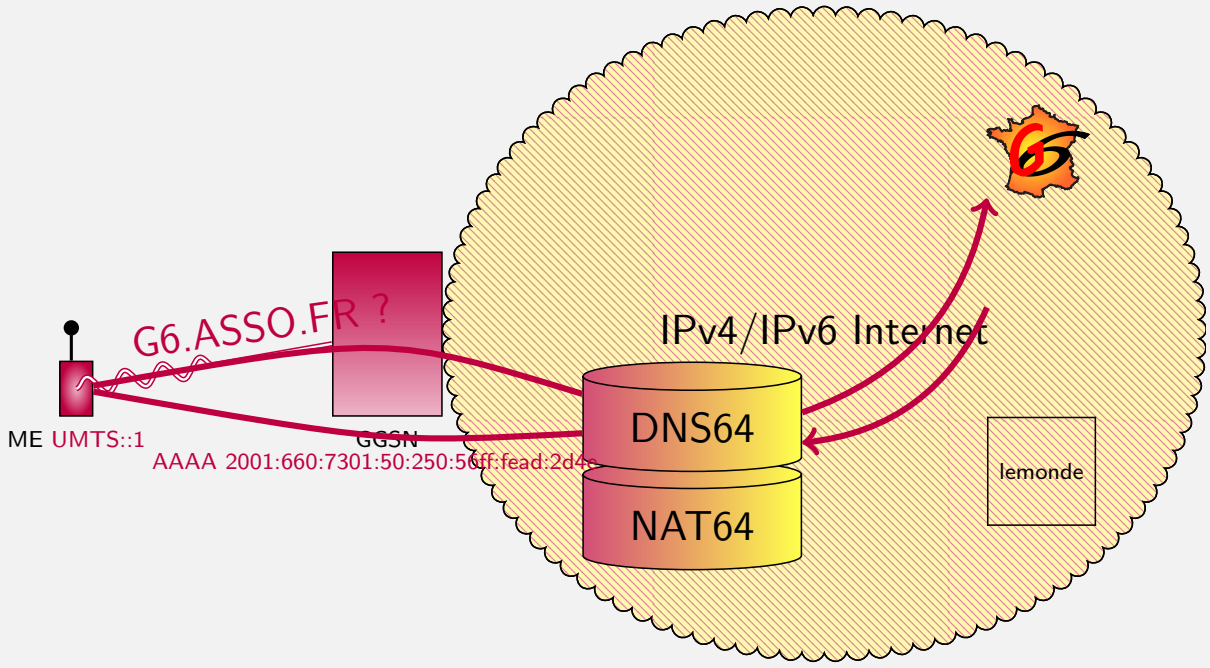
Comments I

D'un point de vue IP, le réseau GRPS/3G est très simple. Le ME (*Mobile Equipment*) correspond par exemple au téléphone portable. Le node B gère la partie transmission. Il est piloté par le RNC (*Radio Network Controller*). Les données sont transportées par le protocole RLC (Radio Link Control) entre le ME et le RNC. Le RNC dialogue avec le SGSN (*Serving GPRS Support Node*) pour les autorisations en liaison avec le HLR (*Home Location Register*). Entre le RNC et le GGSN, un tunnel GTP (*GPRS Tunnelling Protocol*) est établi.

Pour faire de l'IPv6, il faut que le terminal soit IPv6, que le HLR autorise l'accès à ce protocole et que le GGSN dernier routeur avant le réseau Internet accepte cette version du protocole.

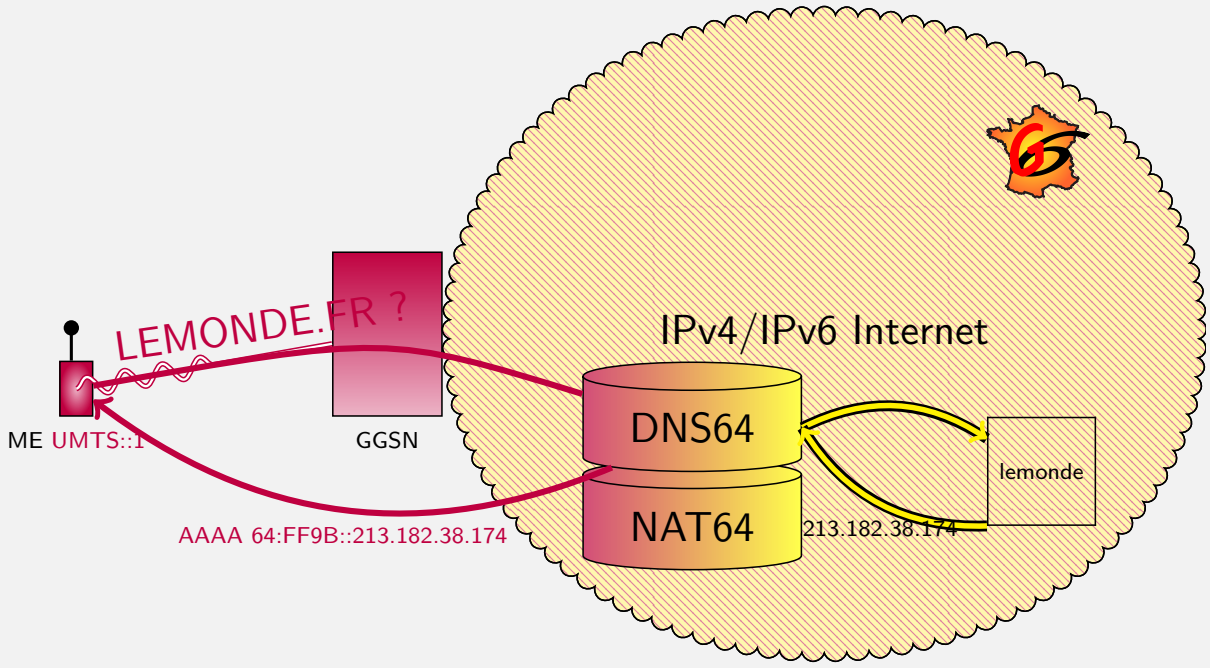
Pour l'instant IPv6 n'est pas intégré dans les piles protocolaires des téléphones les plus modernes. Au niveau le plus bas, l'activation d'IP (on parle de contexte PDP (*Packet Data Protocol*)) peut se faire par des commandes AT. Mais il n'en existe pas pour activer à la fois IPv4 et IPv6 sur un même contexte. L'utilisateur doit donc créer deux contextes, ce qui double le nombre de contextes sur le GGSN. Une solution envisagée actuellement consisterait à ne définir qu'un contexte IPv6 et effectuer une traduction de paquets en sortie pour atteindre les équipements IPv4.

3G data + NAT64/DNS64

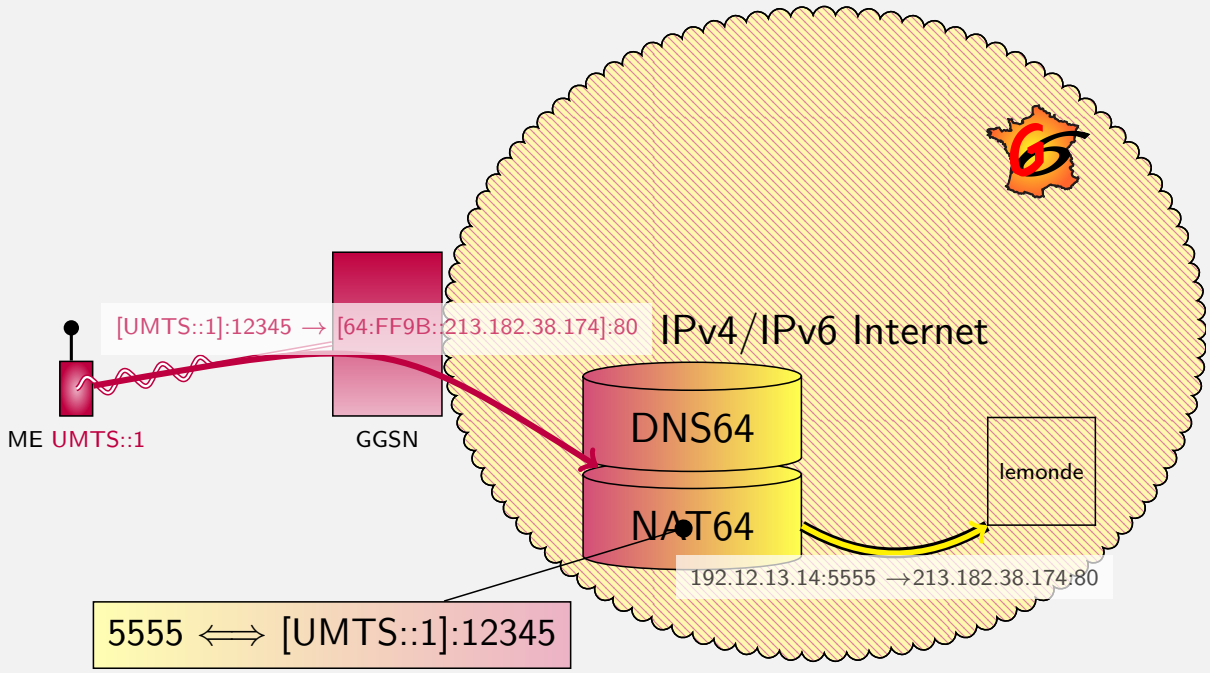


3G data + NAT64/DNS64

Integration ▶ 3G/LTE



3G data + NAT64/DNS64



Comments I

NAT64 fonctionne en deux étapes. Il permet à une machine IPv6 de dialoguer avec une machine IPv4. La machine IPv6 va demander l'adresse IPv6 d'un équipement distant. Comme celui-ci n'est qu'IPv4, il faut mettre dans la chaîne d'interrogation du DNS un équipement qui va traduire les adresses d'une version à l'autre du protocole. Le DNS64 ajoute un préfixe bien connu au début de l'adresse IPv6. Ce préfixe permettra de router les paquets vers un traducteur NAT64. Celui ci pourra retrouver l'adresse IPv4 de la destination. Il devra aussi remplacer l'adresse source pour y mettre à la place une adresse IPv4. Comme dans un NAT traditionnel, le numéro de port servira de référence pour la traduction inverse des paquets en réponse.

Le NAT64 à les même défauts que les NAT44. Si des adresses sont contenues dans les données, elles ne seront pas traduite. Cela le rend incompatible avec des protocoles comme SIP ou le streaming.

Integration

Enterprise

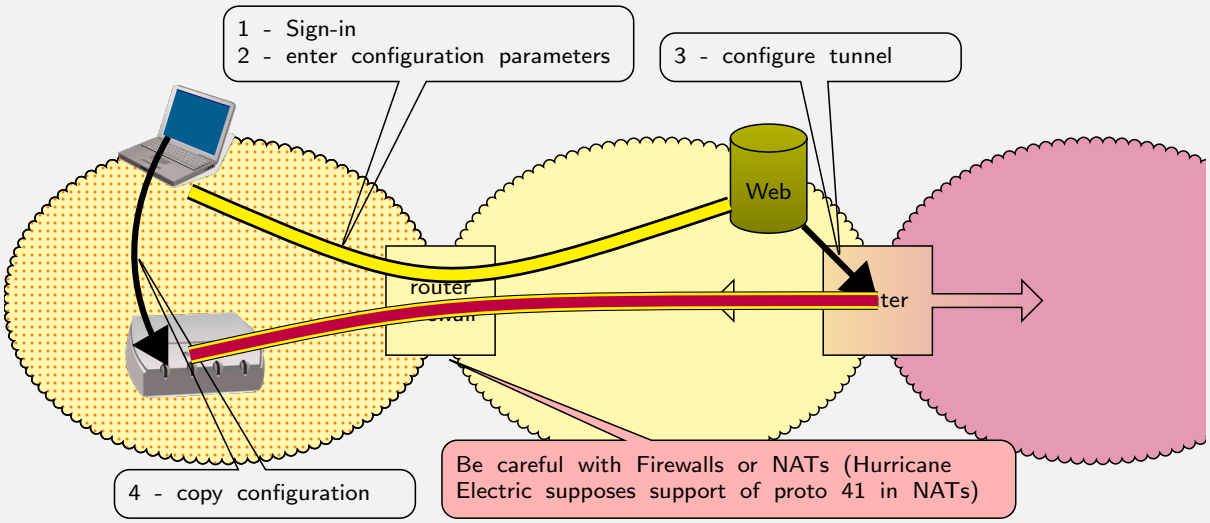
Entreprise Network

- Anticipate: include IPv6 in calls for tenders.
  - RIPE 501 is your friend (http://www.ripe.net/ripe/docs/ripe-501)
- Define your goal:
  - Test: learn about IPv6 or develop products
    - Get temporary connectivity (Tunnel Brokers)
  - V6fy Extranet or/and Intranet
    - Get permanent connectivity and prefix
    - Define addressing plan
    - Define security rules

Tunnel Broker (RFC 3053)

- Hurricane Electric (W tunnelbroker.com)
  - Standard and BGP tunnels
  - Point of Presence in Asia, North America and Europe
- sixxs (W http://www.sixxs.net/main/)
  - Worldwide
- gogo6 (W http://gogonet.gogo6.com/page/freenet6-tunnelbroker)
  - Few Point of Presence
  - in Canada
  - NAT Traversal

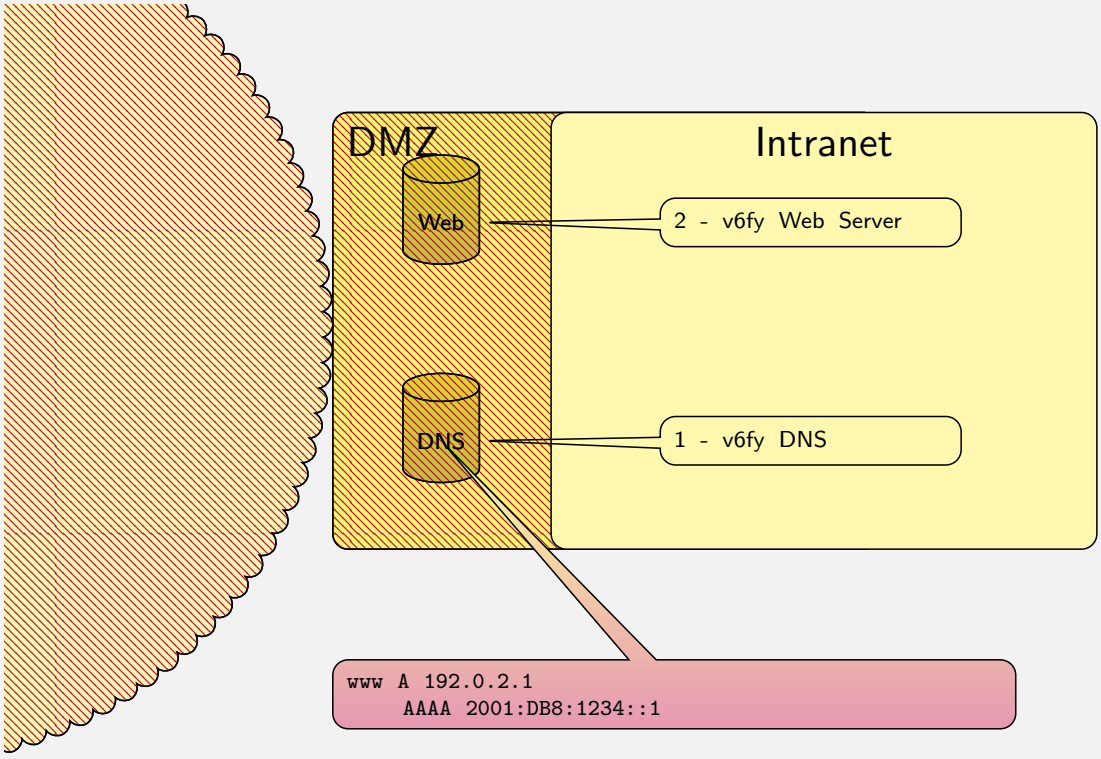
Tunnel Brokers



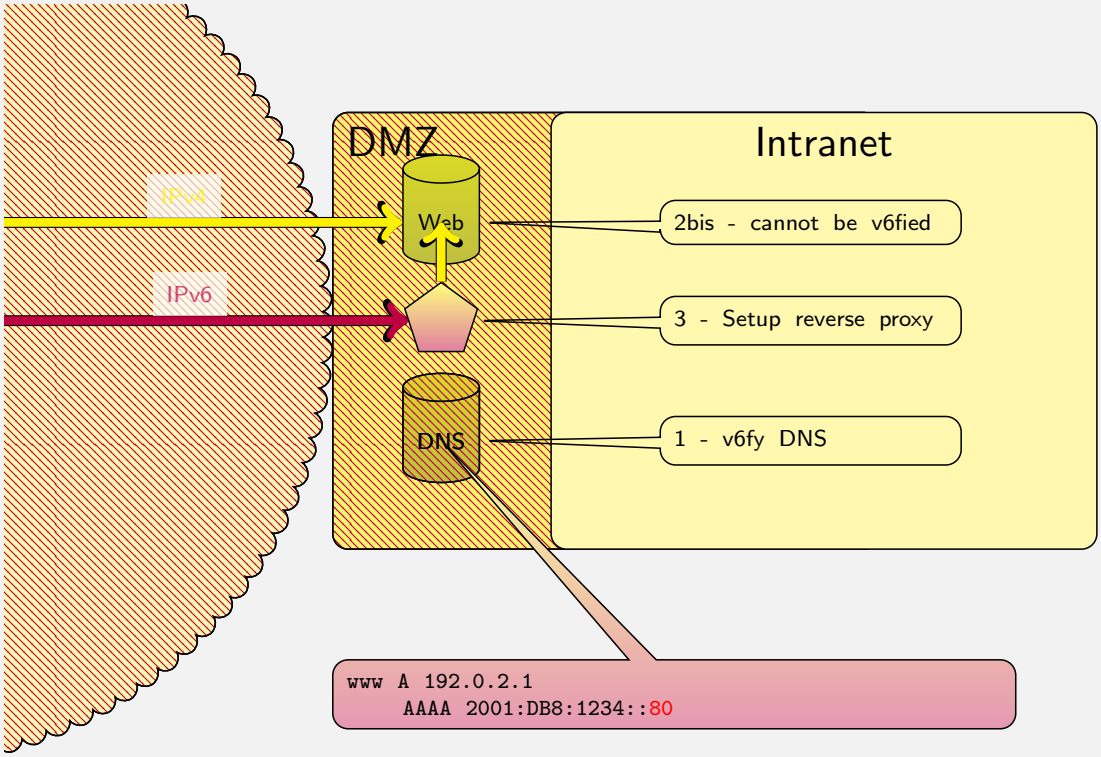
Comments I

Les tunnels brokers sont mis à disposition de la communauté, généralement par des sociétés qui veulent se faire connaître sur le terrain d'IPv6, pour connecter des sites isolés au réseau Internet IPv6. Le principe de fonctionnement est relativement simple. L'utilisateur se connecte sur un serveur web. Après s'être identifié, il peut entrer la configuration de son réseau sur un formulaire. Quand celui-ci est accepté, le serveur web va configurer un routeur une interface tunnel. Le serveur web retourne également à l'utilisateur le script de configuration qu'il devra exécuter sur sa machine.
Suivant les fournisseurs, les points de présence sont plus ou moins loin. Il est préférable de choisir un point relativement proche pour bénéficier d'une bonne qualité de service. L'utilisation d'un NAT peut être un point bloquant pour le déploiement du service.

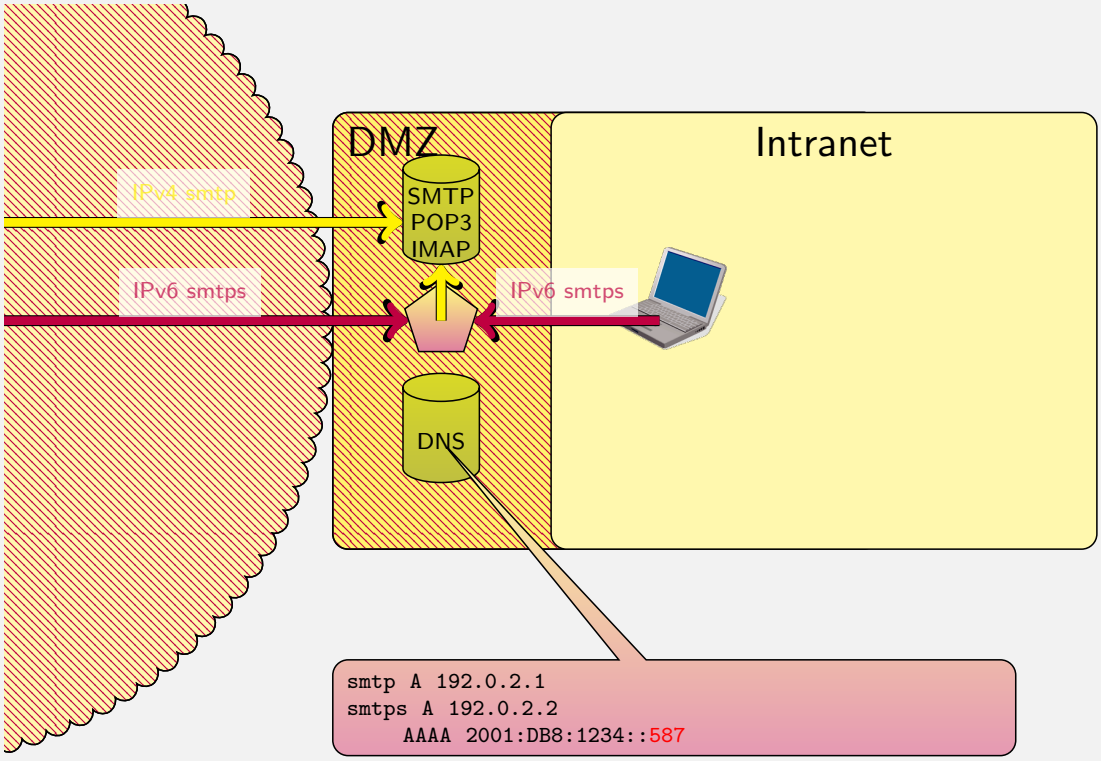
V6fying Extranet (http)



V6fying Extranet (http)



V6fying Extranet (mail)



Comments I

Il est possible de passer d'IPv4 à IPv6 en utilisant des *reverse proxies*. Dans ce cas, on publie dans le DNS l'adresse du proxy, à la place du serveur. Le proxy changera le protocole. Cela fonctionne avec http, mais les protocoles utilisant TLS peuvent être facilement proxifier. On peut donc rendre compatible le web et le mail avec IPv6 sans changer la configuration des serveurs.

Integration

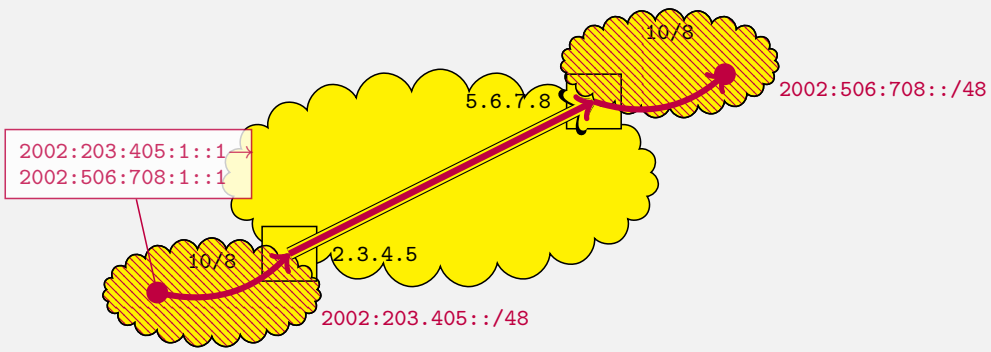
Home network and SOHO

Home Network

- Must (should) be transparent for the end-users
- Last Mile is not currently v6fied
- Wait .... or used Tunnel Brokers
  - DO NOT USE TEREDO OR 6to4
- homenet IETF working group specifies home network behavior for IPv6
  - Today: star topology around single CPE
  - Tomorrow: Mesh network and multi-homing
    - Internet of things
    - smart grid
    - ...

6to4

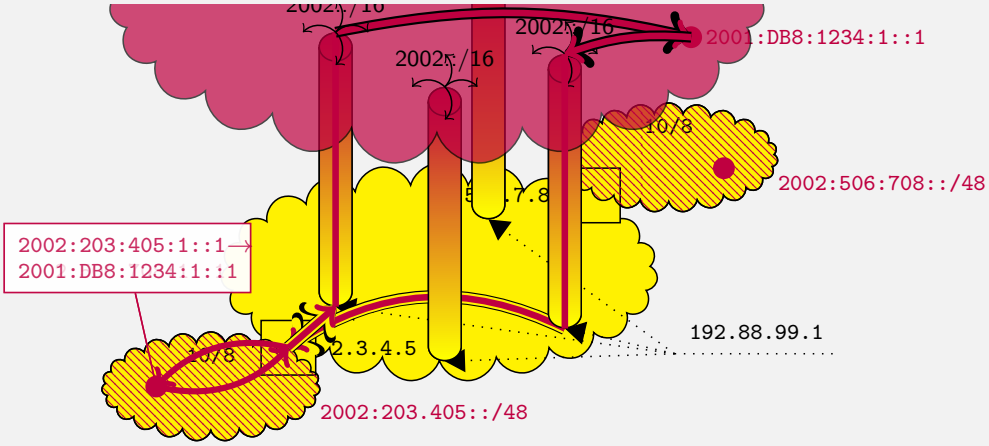
- based on the magic formula 16+32=48
  - 2002::/16 + IPv4 address



- Cannot cross NAT (need to know public address)
- Bad performances.

6to4

- based on the magic formula 16+32=48
  - 2002::/16 + IPv4 address

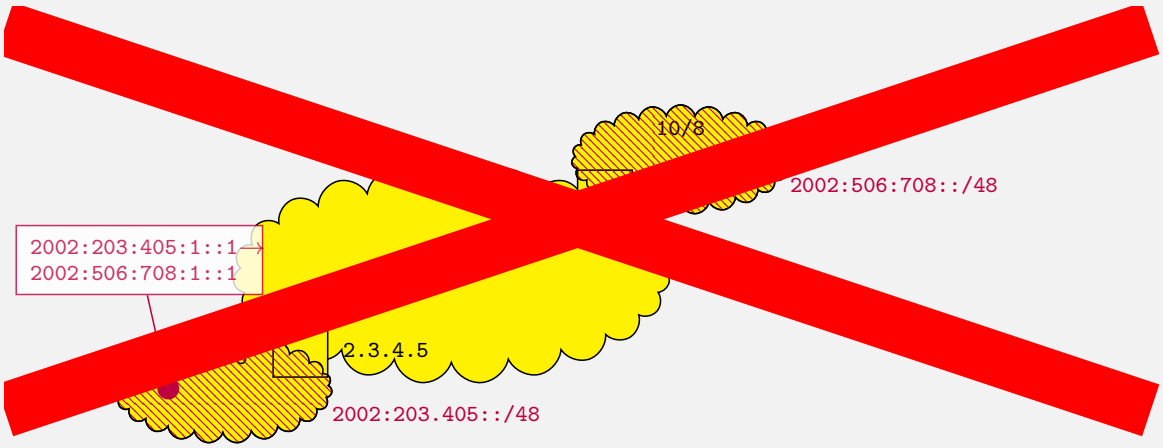


- Cannot cross NAT (need to know public address)
- Bad performances.

6to4

Integration ▶ Home network and SOHO

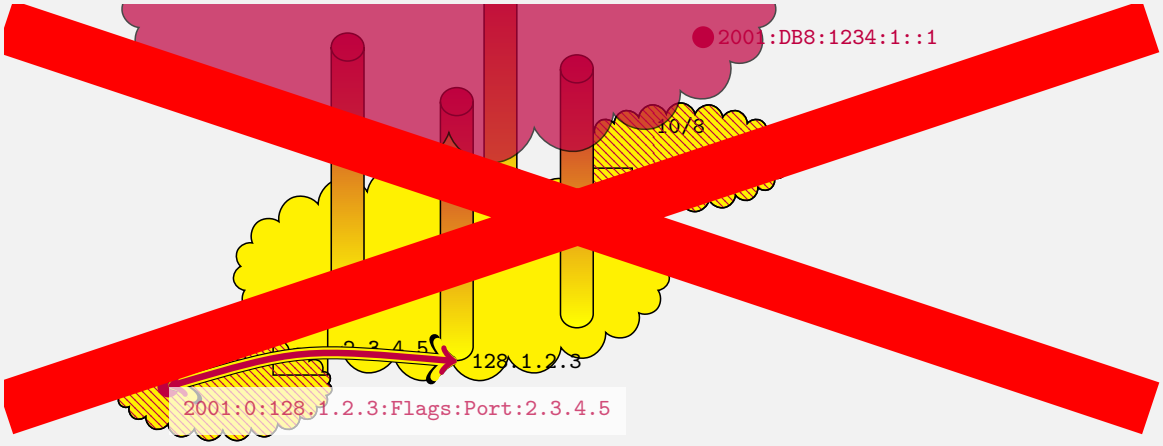
- based on the magic formula 16+32=48
  - 2002::/16 + IPv4 address



- Cannot cross NAT (need to know public address)
- Bad performances.

TEREDO

- Based on NAT Traversal protocol
  - 2001::/32 allocated to this mechanism.



Performances?

- If performances with 6to4 and TEREDO are worst than with IPv4
- What happens if a site decides to activate dual stack on its servers ?
  - Customers will run away
- if IPv6 is dead
  - client starts will IPv6 and then after a long timeout tries IPv4
  - bad performances
- Happy Eyes Ball: try IPv4 and IPv6 simultaneously
- Test the same day IPv6 on main sites
  - Customer will not run away

Performances?

- the 6/8/11: v6Day
  - Good news: nobody notice it
  - 0.3% of IPv6 traffic
- Conclusion: Activating IPv6 do not create troubles
- 6/6/12: IPv6 will be activated on main sites (google, yahoo, facebook, akamai,. . .)
  - Potentially 50% of Internet traffic
  - in reality less since access network is missing