

UMTS

Thi-Mai-Trang Nguyen  
LIP6-UPMC

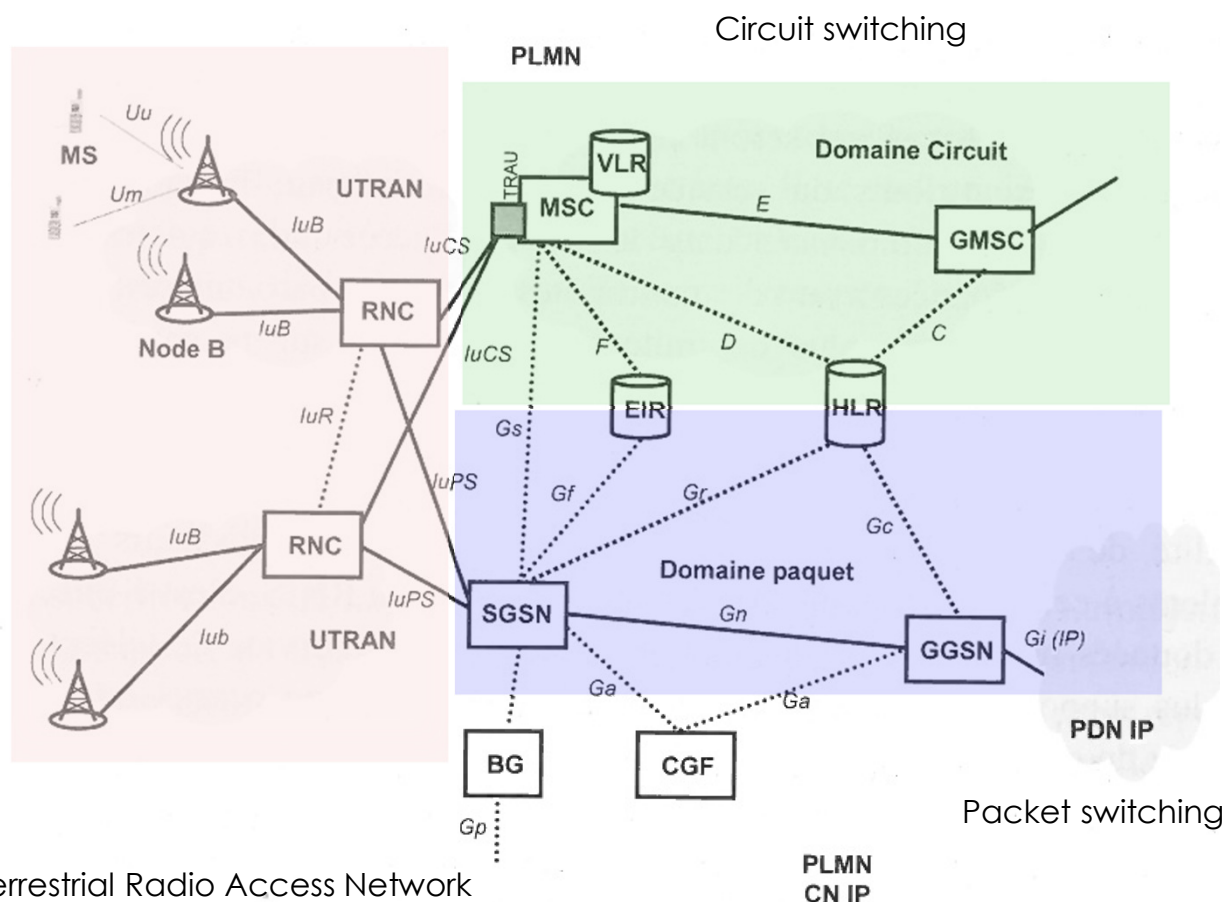
# Plan

- Introduction
- CDMA – Access network
- Core network
- IMS
- SIP protocol

# UMTS

- Universal Mobile Telecommunications System (UMTS) is the 3G technology proposed by Europe
  - **R99** (finalised in 1999, = Rel-3) – new air interface based on CDMA + core network based on GSM/GPRS
  - **Rel-4** TD-SCDMA for UTRA-TDD
  - **Rel-5** IP based core network + IMS (IP-based Multimedia Services) + HSDPA
  - **Rel-6** HSUPA
  - **Rel-7** HSPA+ (High Speed Packet Access +)

# UMTS architecture



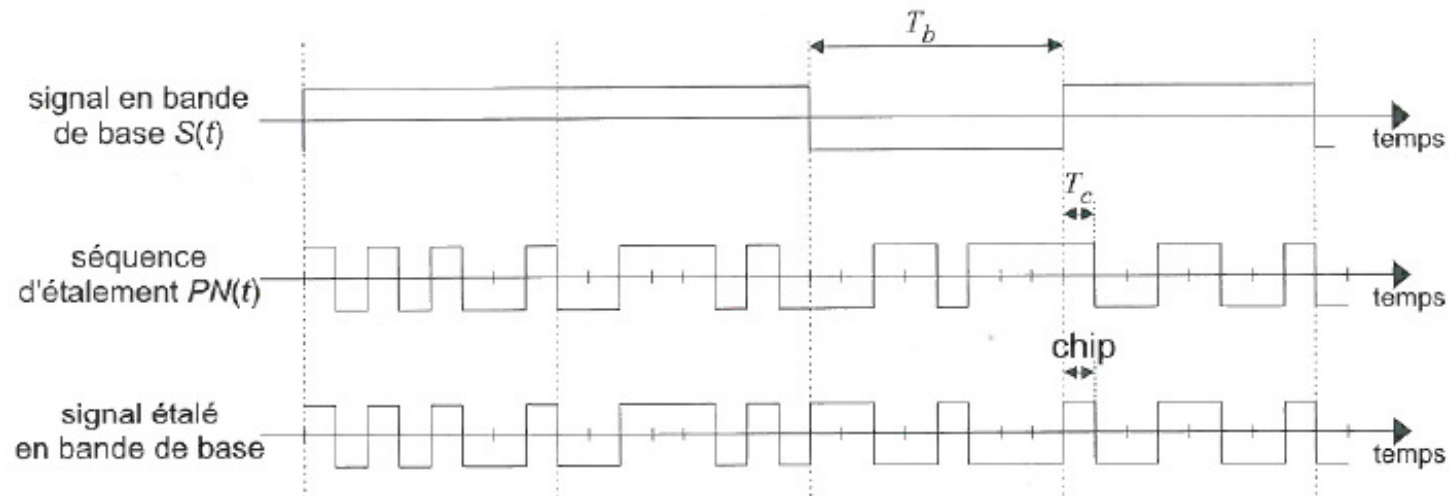
Universal Terrestrial Radio Access Network  
CDMA (Code Division Multiple Access)

# Spread-spectrum

- Principle
  - a signal generated with a particular bandwidth is spread in the frequency domain, resulting in a signal with a wider bandwidth
- Advantage
  - Bonne résistance aux bruits

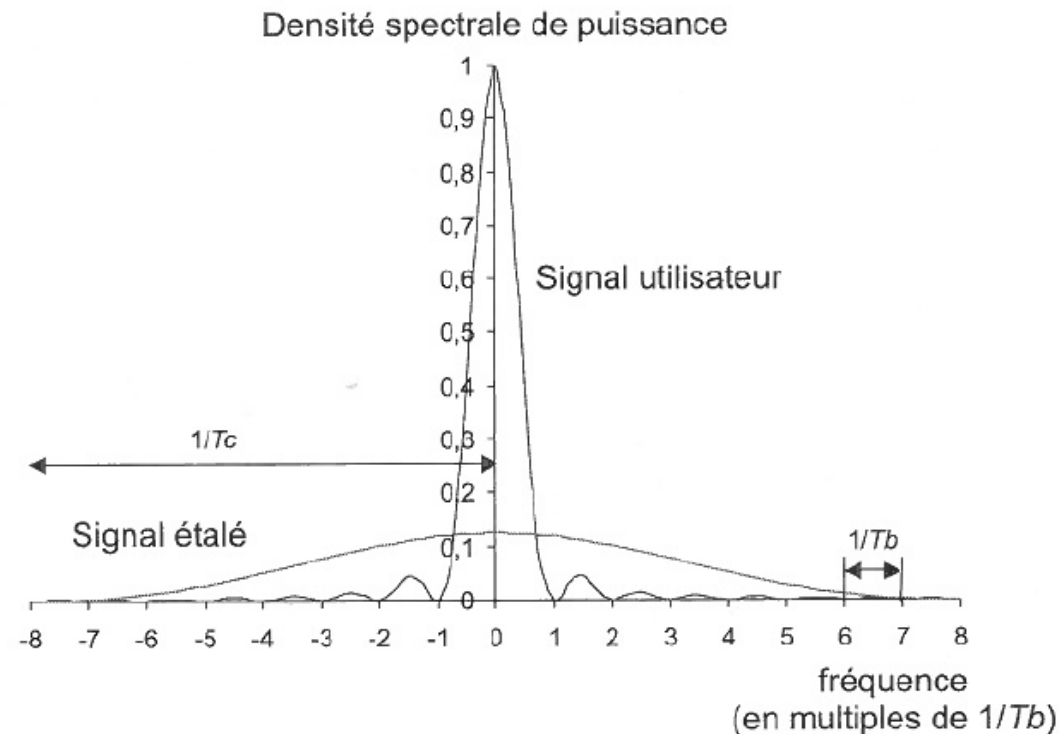
# Direct sequence spread spectrum

- Consider a bit string sent at  $R_b$  [bit/s]
- Spread spectrum consists in multiplying the bit string with a predefined sequence  $PN(t)$  having values +1 or -1 at the rate  $R_c = n * R_b$  [chip/s]
- $n$  is the **spreading factor**
- Each element of the  $PN(t)$  sequence is called a **chip**
- Exemple with  $n = 8$



# Spectrum comparison

- Spread spectrum multiplies the bandwidth of the signal by a factor of  $n$
- The ratio between the signal bandwidth after and before spreading is equal to  $n$



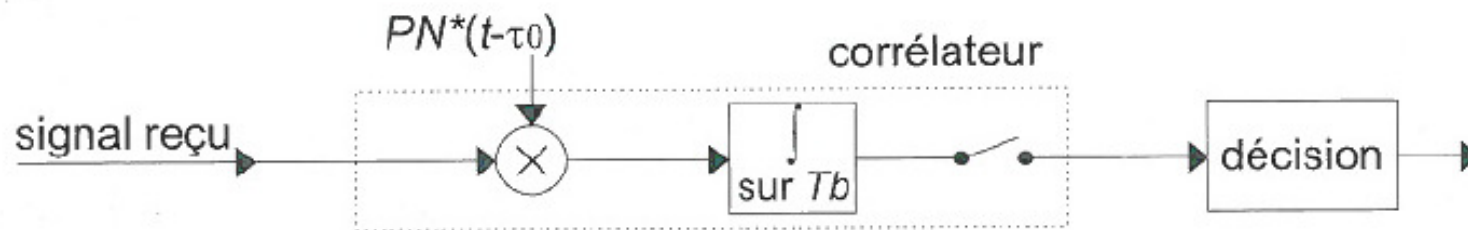
# Spreading gain

- If the signal before spreading has the energy per bit to noise power spectral density ratio  $E_b/N_0$ 
  - The SNR of the signal after spreading is  $(1/n) * (E_b/N_0)$
- To obtain the same SNR, transmission with spread spectrum can tolerate a noise power which is  $n$  times greater than the noise power present in the transmission without spreading
- Example
  - For a binary error rate (BER) of  $10^{-3}$ , we need an SNR of 6 dB (signal power = 4 \* noise power)
  - With a spreading factor  $n = 128$ , the SNR of the spread signal is  $6 - 10\log_{10}(128) = -13$  dB (noise power = 200 \* signal power) → we can have a low BER even when the spread signal is sent completely under the noise
  - $n$  is the **spreading gain**

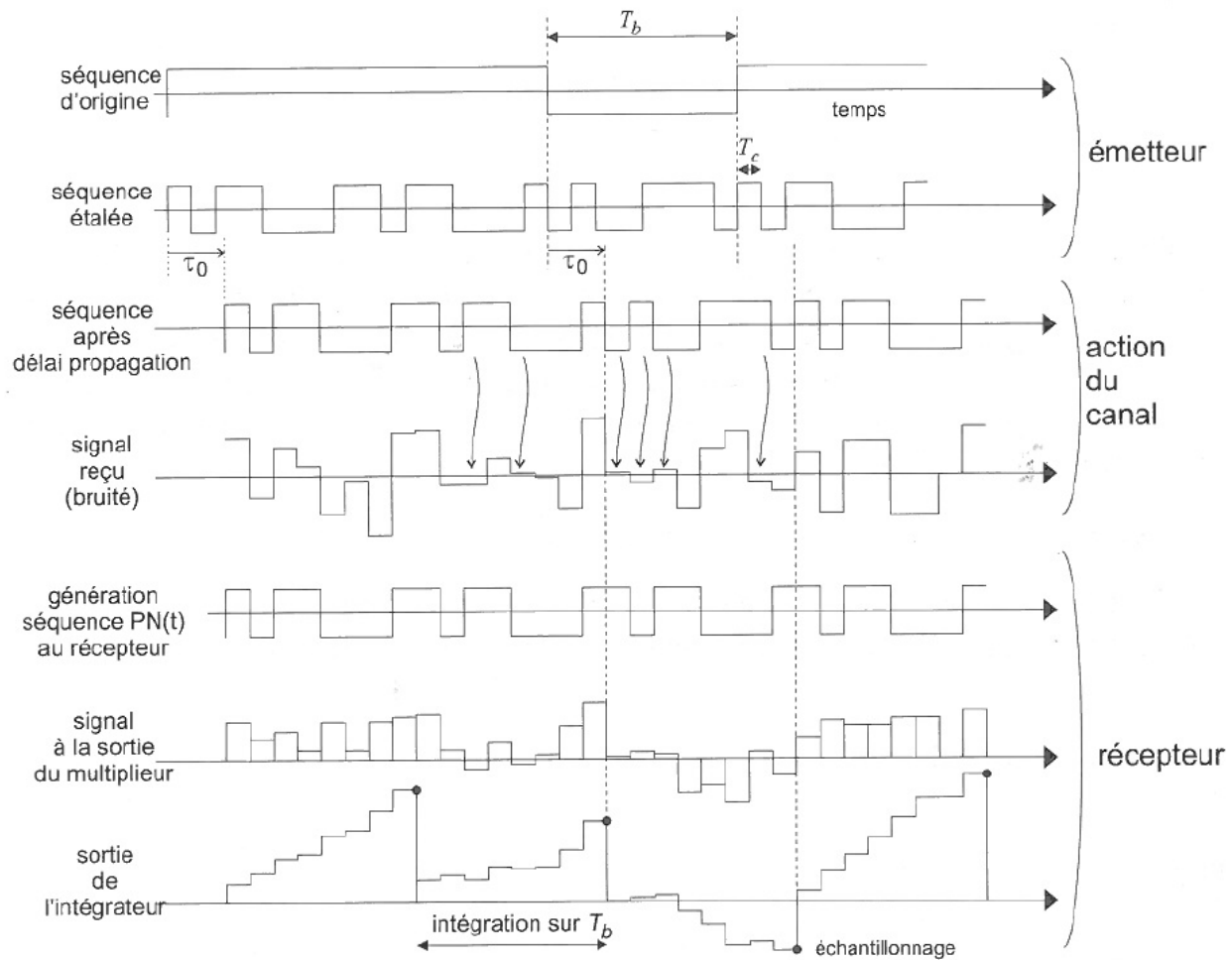


# Receiver

- To recover the original signal, we multiply the received signal by  $PN(t)$
- Calculating the integral of the signal over  $T_b$  (bit duration) allows detecting the transmitted bit
- The receiver needs to estimate the propagation delay ( $\tau_0$ ) to recover the original signal



# Exemple de réception



# CDMA (Code Division Multiple Access)

- Le multiplexage a pour principe de transmettre plusieurs signaux en parallèle sans que l'un n'interfère avec l'autre
- CDMA utilise l'étalement de spectre et suit le principe de multiplexage par des codes orthogonaux
- Un flux de données à un débit  $R_b$  est transmis par une séquence de chips à un débit supérieur  $R_c = n * R_b$
- Le multiplexage des différents flux s'effectue en choisissant des séquences ayant de bonnes propriétés d'orthogonalité

# Matrice d'Hadamard

- Les matrices d'Hadamard  $H_n$  sont construites récursivement à partir d'une matrice de dimension un  $H_1 = (+1)$  et de la relation récurrence

$$H_{2n} = \begin{pmatrix} +H_n & +H_n \\ +H_n & -H_n \end{pmatrix}$$

- Les dimensions des matrices d'Hadamard sont des puissances de 2

# Exemples des matrices d'Hadamard

- On calcule facilement  $H_2$

$$H_2 = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}$$

- Après 2 récurrences, on obtient  $H_8$

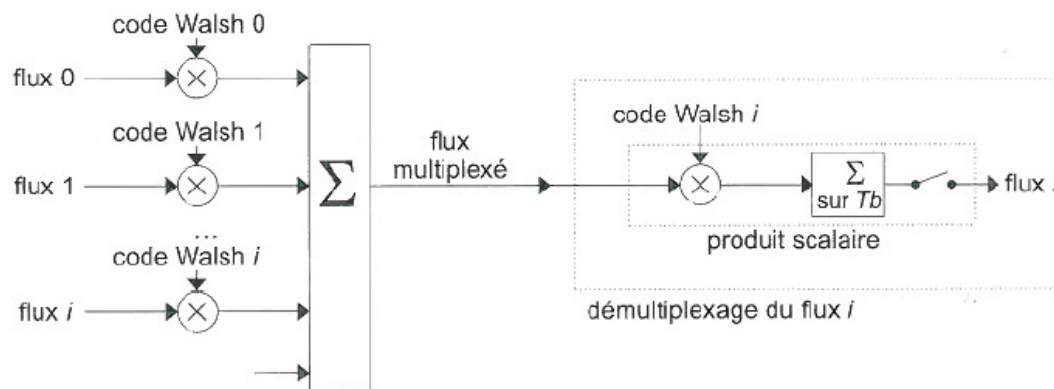
$$H_8 = \begin{pmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{pmatrix}$$

# Code de Walsh

- Une ligne de la matrice d'Hadamard est appelée un code de Walsh
- Les  $n$  codes de Walsh dans une matrice d'Hadamard  $H_n$  sont orthogonaux entre eux (c.à.d. tous les produits scalaires de 2 codes différents sont 0)
- Exemple avec deux dernière lignes (nommé  $w_6$  et  $w_7$ ) de  $H_8$ 
  - $\langle w_6, w_7 \rangle = (+1)*(+1) + (+1)*(-1) + (-1)*(-1) + (-1)*(+1) + (-1)*(-1) + (-1)*(+1) + (+1)*(+1) + (+1)*(-1) = 1 - 1 + 1 - 1 + 1 - 1 + 1 - 1 = 0$

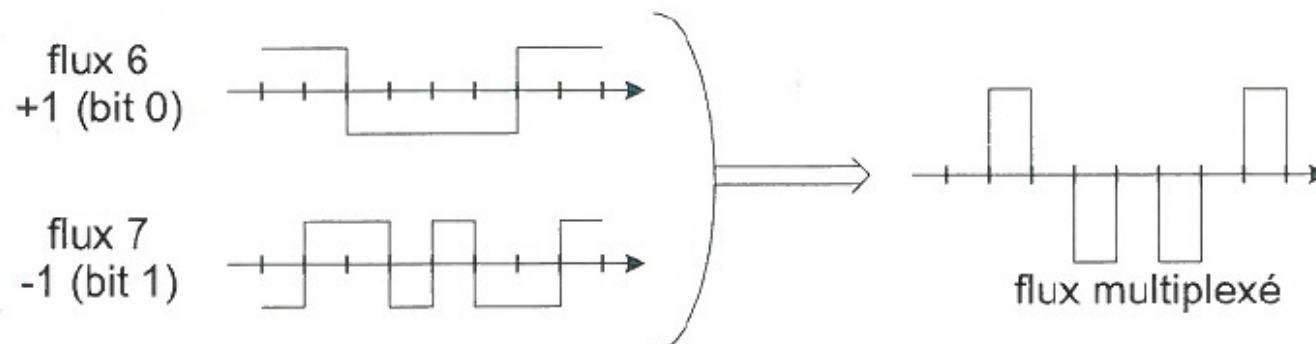
# Multiplexage et démultiplexage par les codes

- Une matrice d'Hadamard de dimension  $n$  possède  $n$  codes de Walsh
- Il est possible de transmettre  $n$  flux de  $n$  utilisateurs simultanément
- Chaque utilisateur se voit affecter un code  $w_i$  qui est une ligne de la matrice d'Hadamard
- A la réception, pour extraire la donnée transmise par l'utilisateur  $i$ , il suffit de faire le produit scalaire du signal reçu et de la séquence  $w_i$
- La valeur reçue est égale à  $n$  fois la valeur du signal émis  $\rightarrow$  on peut donc facilement récupérer le bit émis



# Exemple de multiplexage en CDMA

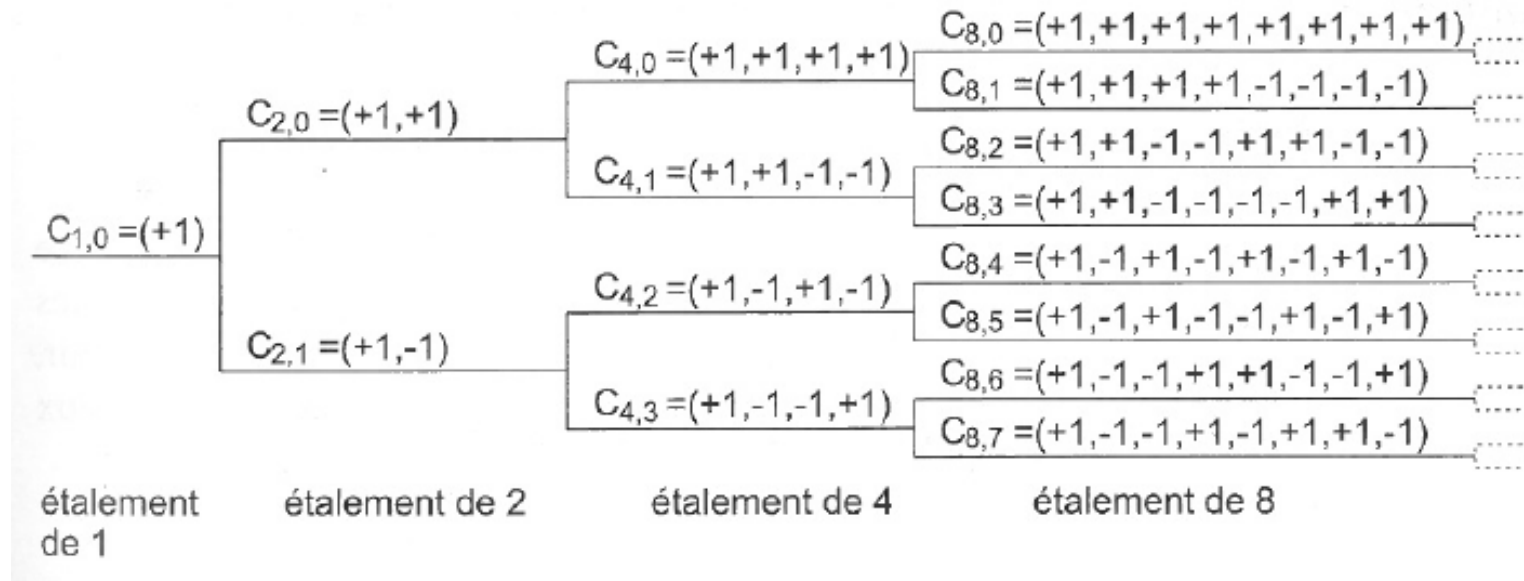
- On considère 2 flux (nommés flux 6 et flux 7) auxquels sont affectés les codes  $w_6$  et  $w_7$ 
  - Sur le flux 6, on transmet +1 (bit 0)
  - Sur le flux 7, on transmet -1 (bit 1)
- Le flux multiplexé est  $0 +2 \ 0 -2 \ 0 -2 \ 0 +2$
- $\langle \text{flux 6}, w_6 \rangle = 0 +2 -0 +2 -0 +2 +0 +2 = 8 \rightarrow$  le démultiplexage redonne un signal de valeur 1 (bit 0)
- $\langle \text{flux 7}, w_7 \rangle = +0 -2 -0 -2 -0 -2 +0 -2 = -8 \rightarrow$  le démultiplexage redonne un signal de valeur -1 (bit 1)





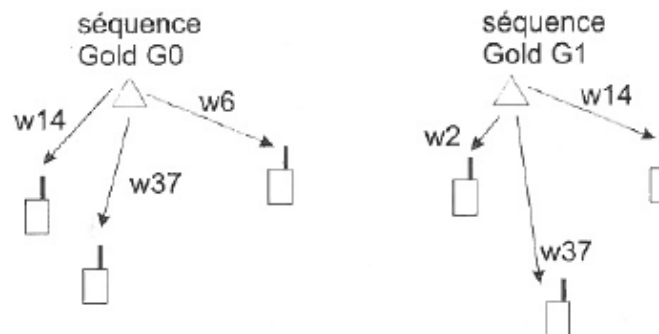
# Codes OVSF

- Orthogonal Variable Spreading Factor
- Il est possible d'obtenir les mêmes séquences que celles de Walsh mais dans un ordre différent



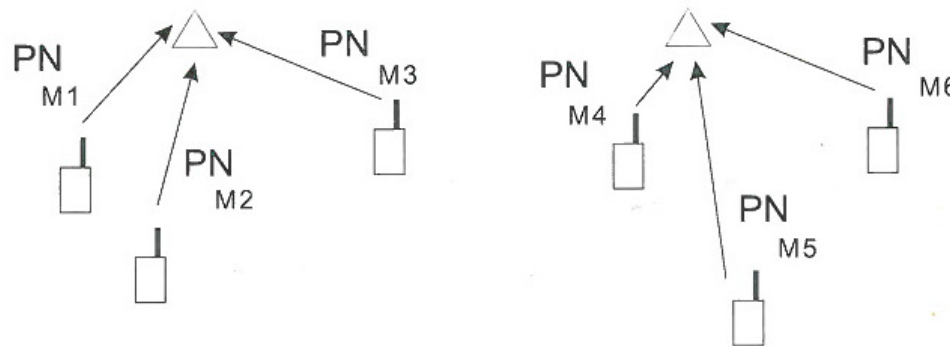
# Principe de transmission sur la voie descendante

- Les codes orthogonaux permettent un multiplexage de plusieurs transmissions
  - Une transmission vers un mobile = un canal = un code
- Les séquences PN (e.g. les séquence de Gold) permettent une réutilisation de la même fréquence sur différentes cellules
  - Une séquence de Gold appliquée à une station de base = un code d'embrouillage (scrambling code)
  - Les séquences de Gold ont de bonnes propriétés d'autocorrélation permettant à l'ensemble des interférences des stations de base voisines d'avoir une caractéristique proche d'un bruit Gaussien
- Le signal pilote permet au récepteur d'estimer les différents trajets de propagation



# Principe de transmission sur la voie montante

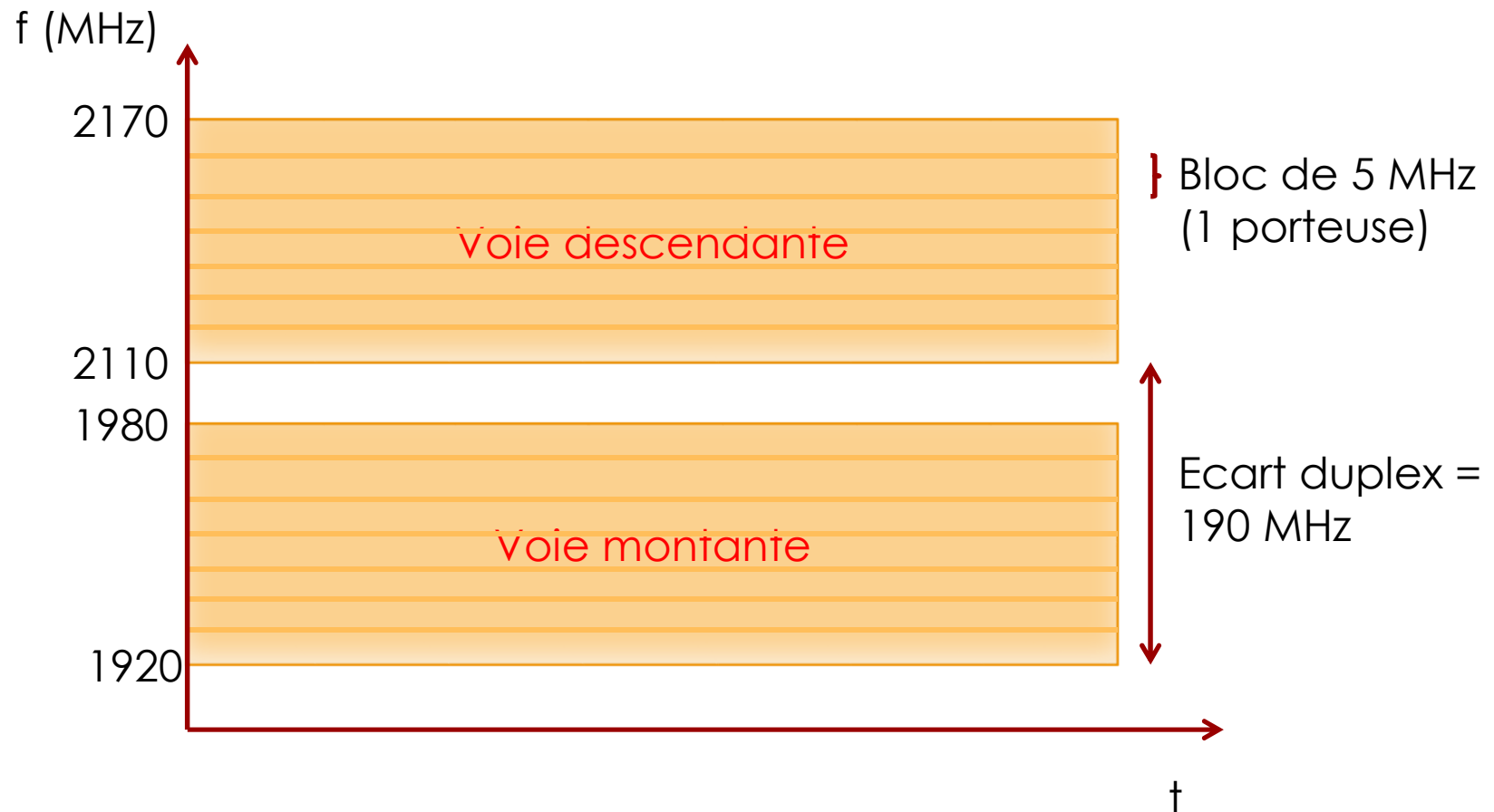
- Utiliser des séquences PN pour différencier les mobiles entre eux
- La séquence PN est propre au mobile et elle ne dépend pas de la station de base à laquelle il est rattaché → faciliter le soft-handover
- Pour que la station de base connaisse la séquence utilisée par le mobile, tout échange est précédé d'un accès aléatoire qui comprend la transmission de la séquence utilisée par le mobile à la station de base



# UTRA-FDD

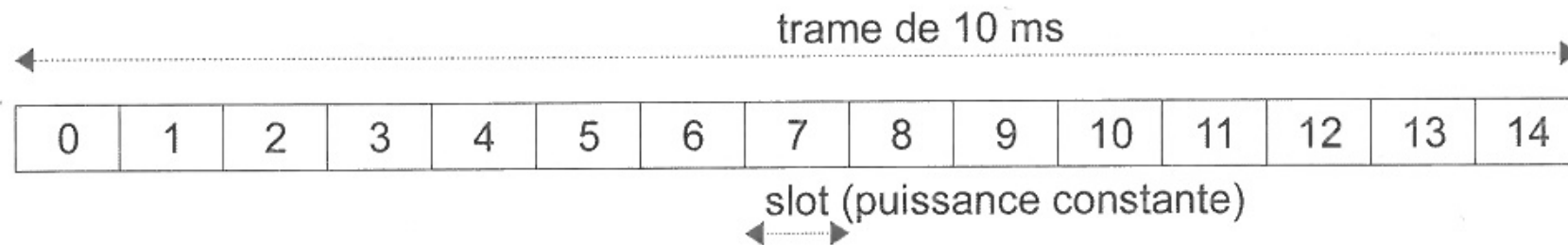
- CDMA à  $R_c = 3,84$  Mchip/s
- Utilise la modulation QPSK (Quaternary Phase Shift Keying)
  - 1 symbole = 2 bits
  - $SF = R_c/R_s$
- Voie descendante
  - Les canaux sont séparés par les codes OVSF orthogonaux
  - Un code d'embrouillage est appliqué pour différencier les stations de base
- Voie montante
  - Séparation se fait par un code pseudo aléatoire propre à chaque mobile

# Fréquences



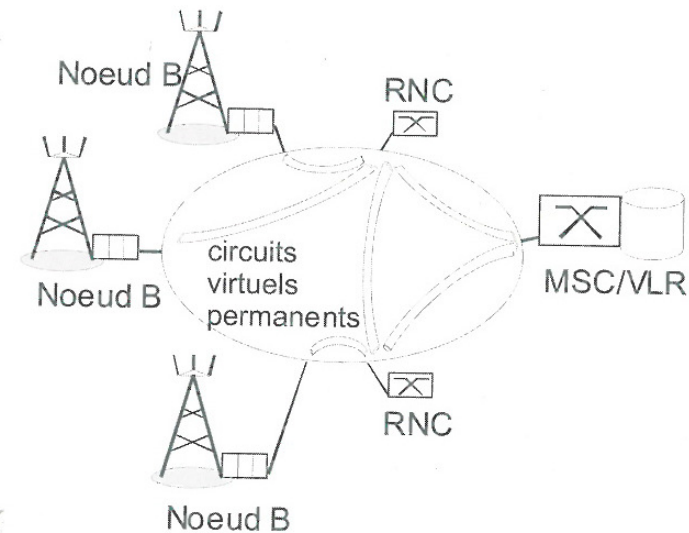
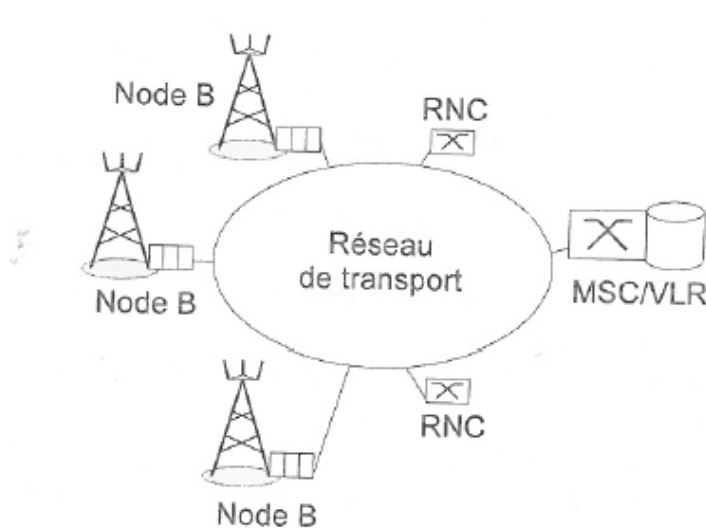
# Structure de la trame

- 10 ms = 15 slots de 666,67  $\mu$ s
- Le slot sert à définir la granularité du contrôle de puissance
- SFN (System Frame Number) : 0 - 4095



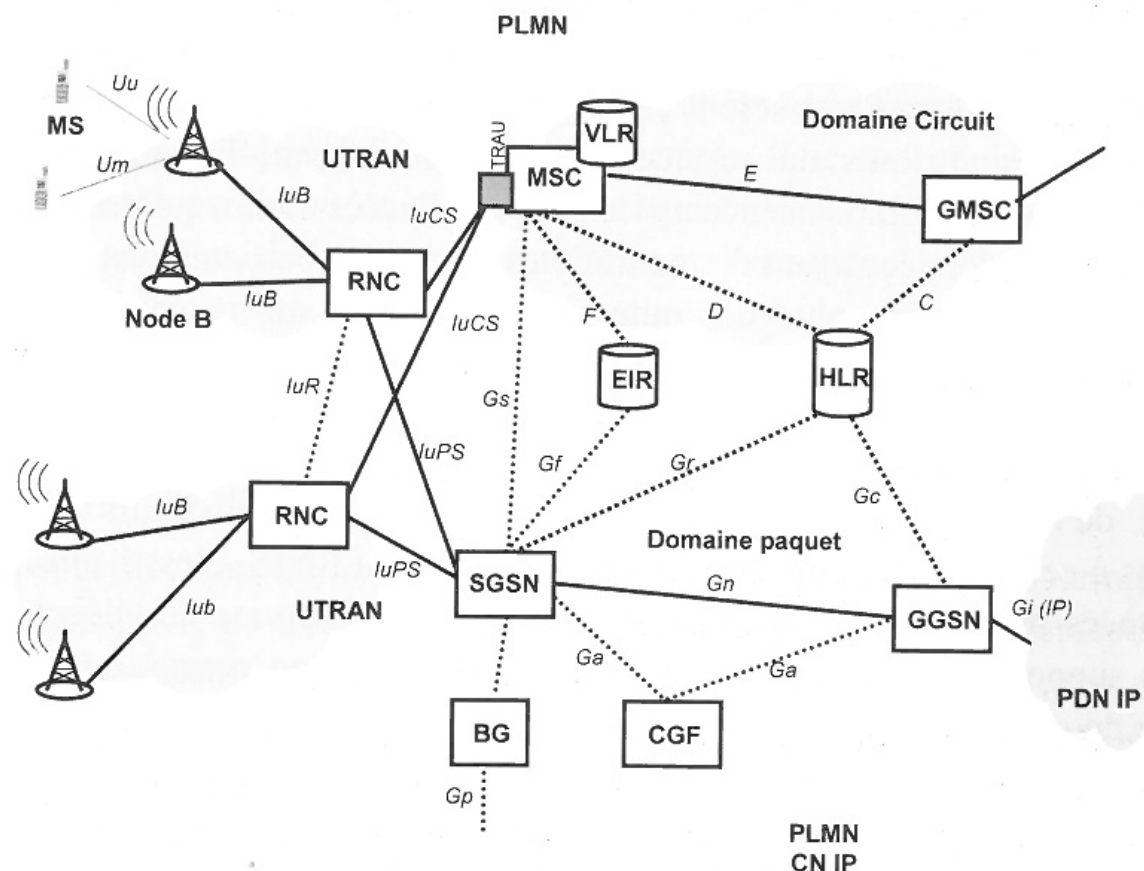
# Réseau d'accès UTRAN

- Nœud B (Node B) est un ensemble d'émetteurs-récepteurs (= BTS de GSM)
- RNC (Radio Network Controller) est le contrôleur de ressource radio (= BSC de GSM)
- Réseau de transport (qui remplace les liaisons spécialisées du GSM) de type ATM (R99) ou IP (R5)



# Réseau cœur UMTS (R99)

- Domaine circuit (CS domain)
  - Une évolution du cœur de réseau GSM
- Domaine paquet (PS domain)
  - Une évolution du cœur de réseau GPRS





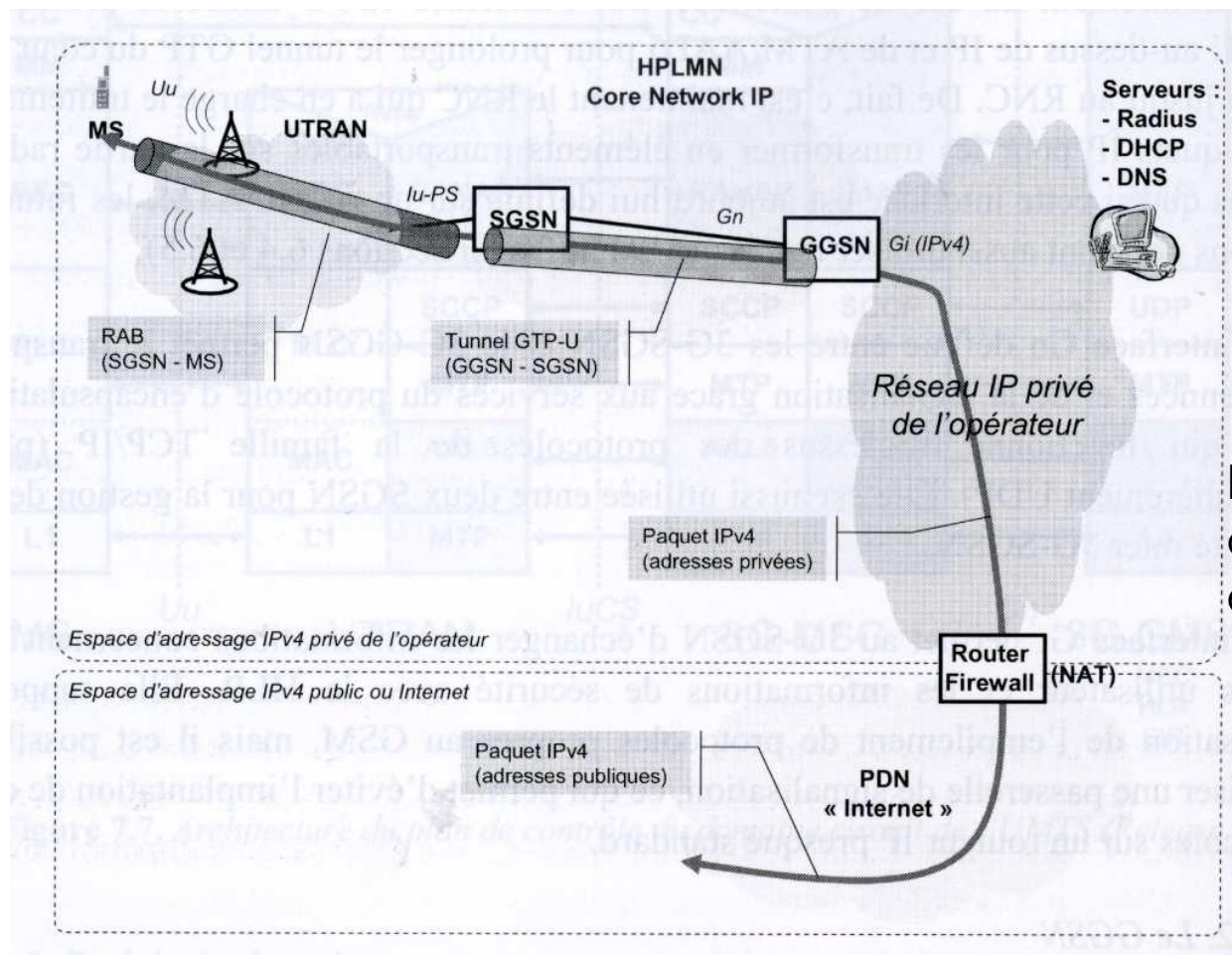
## 3G-GGSN

- 3G - Gateway GPRS Support Node
- La passerelle entre le domaine PS et un PDN (avoir une connexion directe avec un ou plusieurs PDN)
- Supporter les fonctions d'un routeur d'accès IP
  - Filtrage des communications (Access List)
  - Traduction d'adresse (NAT – Network Address Translation)
  - Relais des demandes DHCP vers le serveur DHCP
  - Relais des demandes d'authentification vers le serveur Radius
- Le mobile peut se déplacer de SGSN en SGSN mais les données passent toujours par le GGSN qui interconnecte le domaine PS et un réseau PDN

## 3G-SGSN

- 3G - Serving GPRS Support Node
- Utiliser le protocole GTP-U au-dessus de IP et de ATM/AAL5 pour prolonger le tunnel GTP du cœur de réseau jusqu'au RNC
- C'est le RNC qui a en charge le traitement des paquets IP pour les transformer en éléments transportables sur la partie radio
- Le protocole GTP est utilisé entre le 3G-SGSN et le 3G-GGSN pour le transport des données et de la signalisation (un tunnel GTP par un contexte PDP)
- Le tunnel suit le déplacement du mobile du SGSN en SGSN
- Echanger des données concernant le profil d'utilisateur avec le HLR pour vérifier l'abonnement aux services

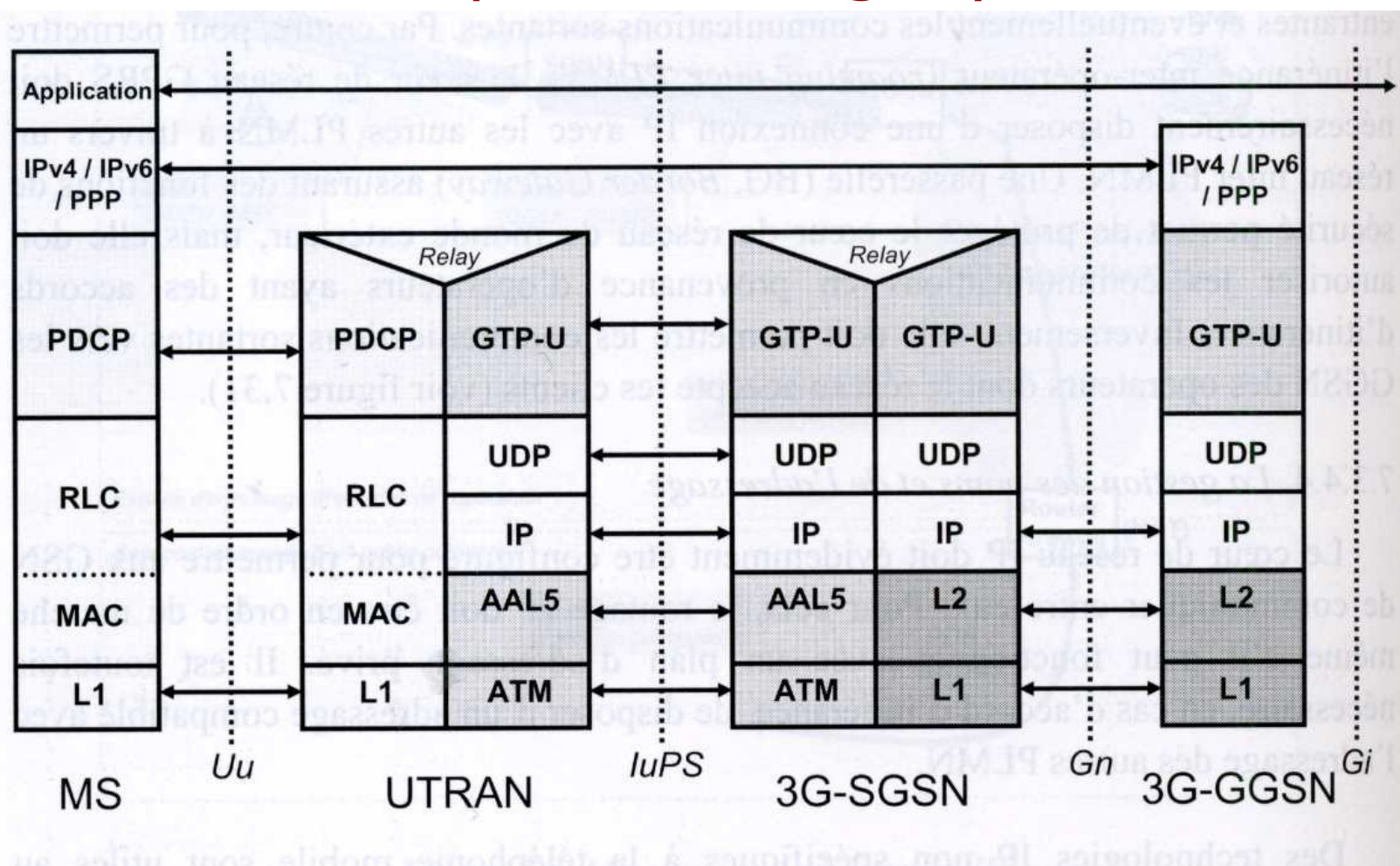
# Mise en œuvre du service IP



Le serveur DNS (Domain Name Server) permet de retrouver l'adresse IP du GGSN à partir de l'APN

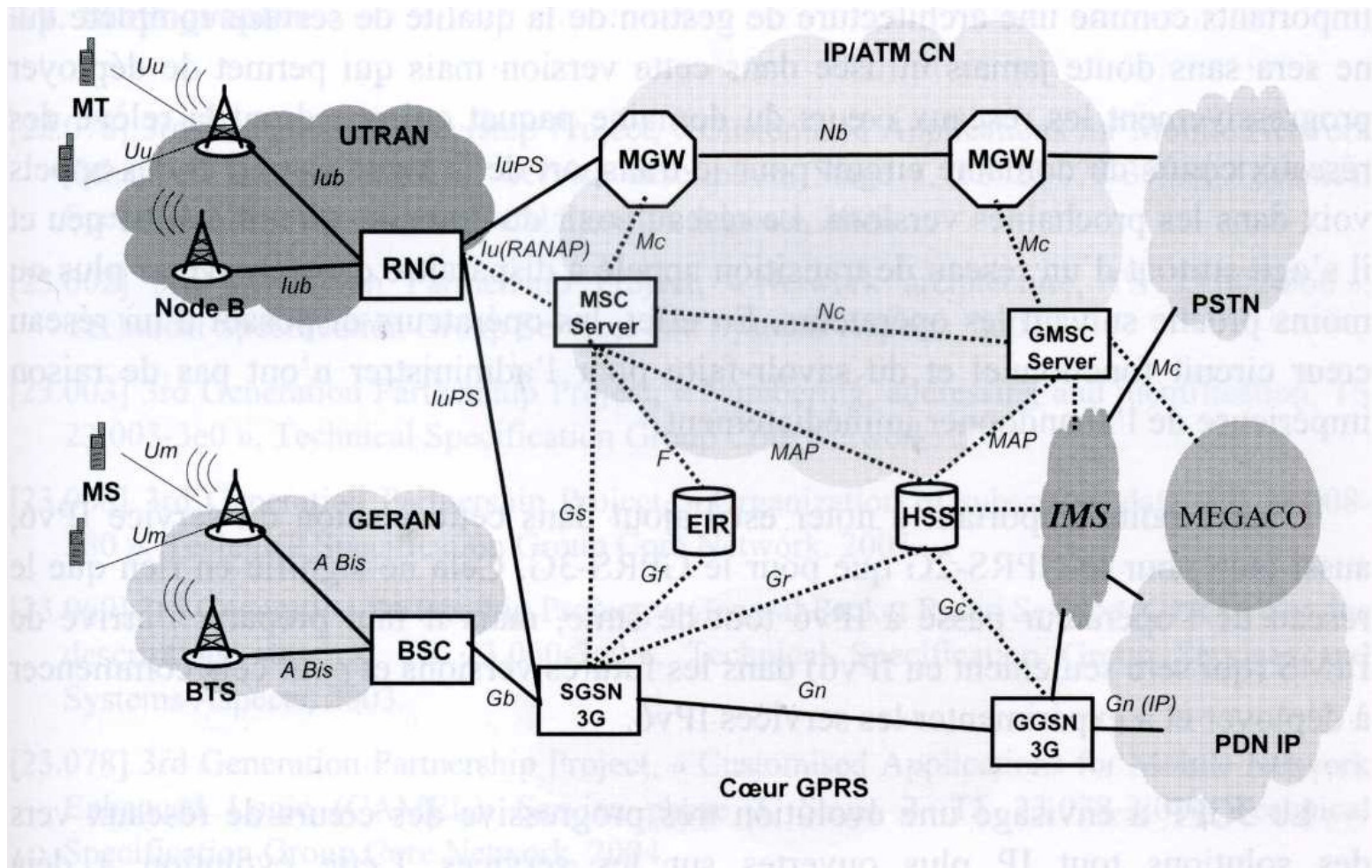
Le réseau IP privé permet aux GSNs de communiquer entre eux

# Protocoles (plan usager)





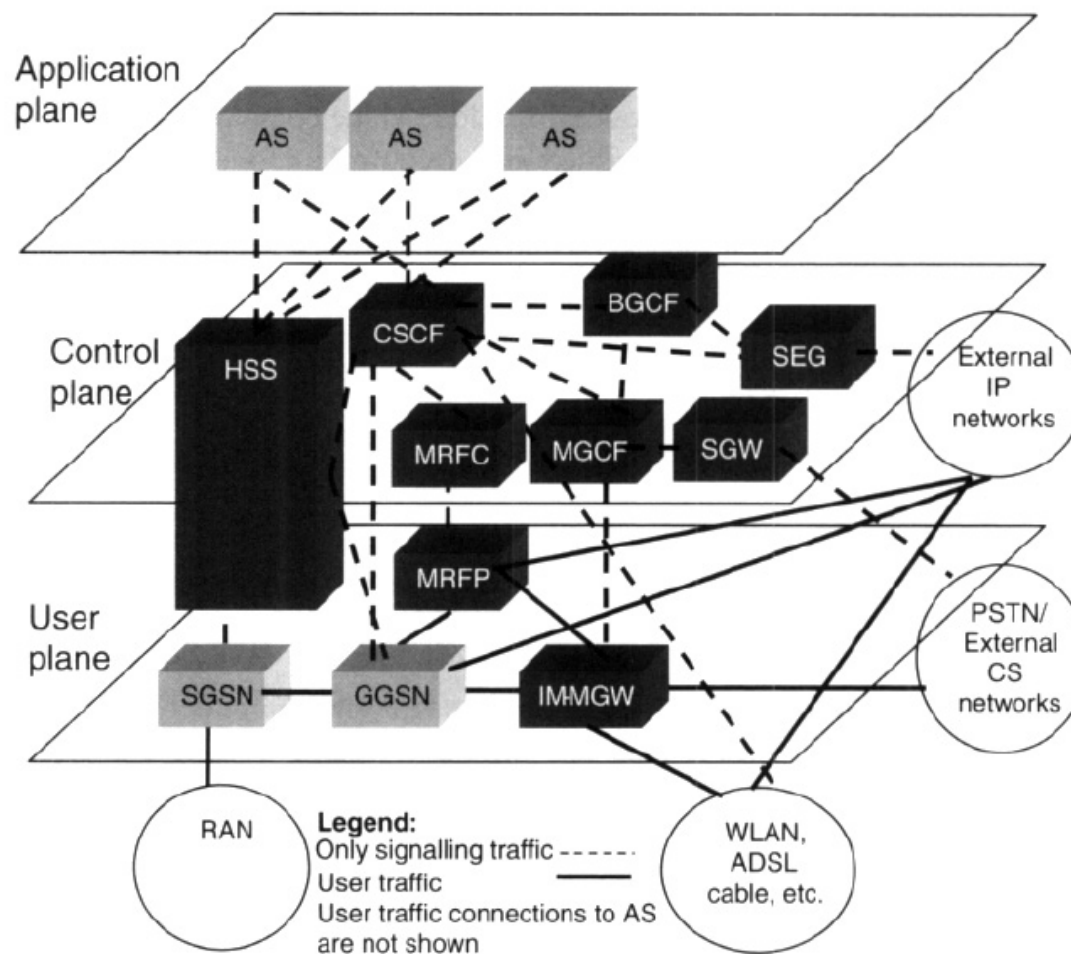
# Architecture UMTS (Release 5)



## Architecture UMTS (R-5)

- IMS (IP Multimedia Subsystem) est une nouveauté de la version 5 de l'UMTS
- Ce domaine est basé sur l'IP et utilise le protocole SIP (Session Initiation Protocol) pour la signalisation
- Le domaine circuit sera conservé pour un temps pour permettre aux anciens terminaux d'utiliser le réseau

# Architecture IMS



- Gestion des sessions + routage:  
Les CSCFs  
(Call Session Control Function)

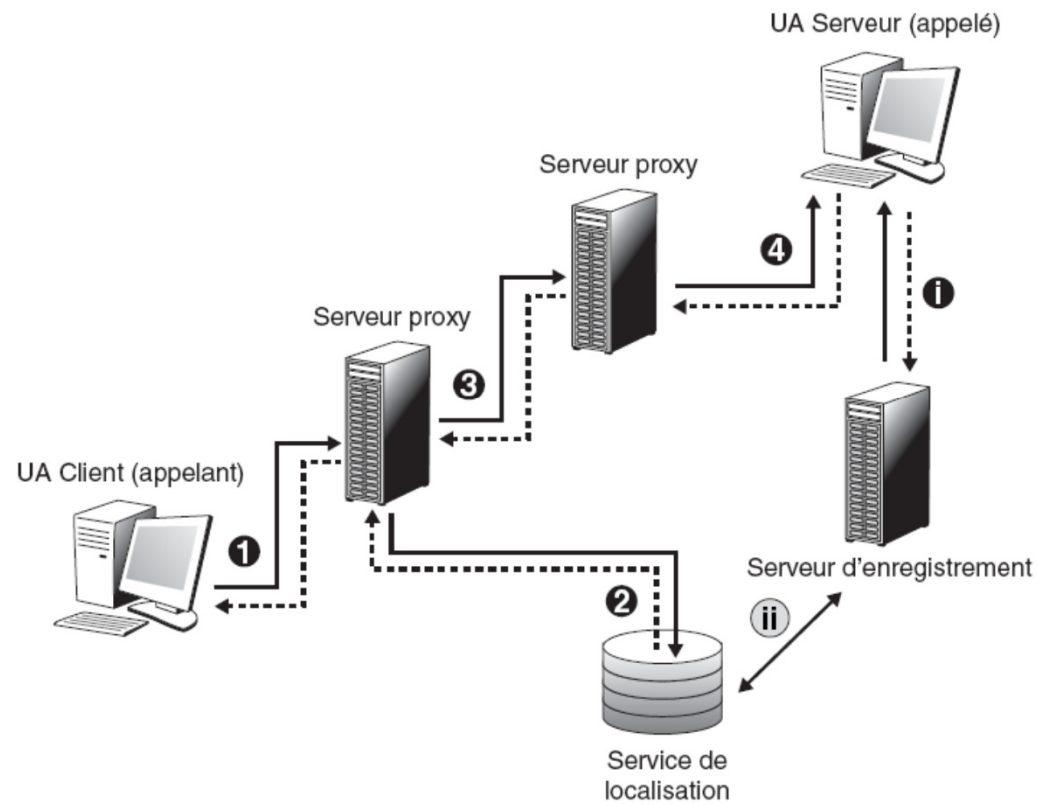
- Base des données:  
HSS  
(Home Subscriber Server)

# SIP

- Session Initiation Protocol, RFC 3261
- Standardisé en 1999 par l'IETF (Internet Engineering Task Force)
- Protocole de signalisation pour
  - Localiser l'appelé
  - Contacter l'appelé pour connaître sa volonté de recevoir l'appel
  - Échanger les informations de média permettant d'établir la session multimédia
  - Modifier les sessions multimédias existants
  - Terminer des sessions multimédias existants



# Architecture SIP (1)



# Architecture SIP (2)

- Entités utilisatrices - Agents utilisateurs (UA – User Agent)
  - Client agent utilisateur (UAC – User Agent Client)
    - L'entité qui envoie les requêtes SIP
    - Initialisation des sessions à la demande de l'utilisateur
  - Serveur agent utilisateur (UAS – User Agent Server)
    - L'entité qui répond aux requêtes SIP
    - Réception de toutes les sessions à destination de l'utilisateur
  - UA joue à la fois le rôle de l'UAC et le rôle de l'UAS
    - Un terminal SIP doit implémenter les deux côtés client SIP et serveur SIP car il doit jouer le rôle d'un client pour envoyer une requête SIP et jouer le rôle d'un serveur pour envoyer les réponses

# Architecture SIP (3)

## ■ Entités réseaux

### ■ Serveur proxy (Proxy Server)

- Avoir une fonctionnalité de relais

### ■ Serveur d'enregistrement (Registrar Server)

- Enregistrer l'association de l'identifiant d'un utilisateur et son adresse IP

### ■ Serveur de redirection (Redirect Server)

- Répondre à des requêtes en donnant l'adresse IP d'un utilisateur ou d'un serveur de localisation

### ■ Serveur de localisation (Location Server)

- Contenir la base de données de l'ensemble des abonnés
- Souvent associé avec le serveur d'enregistrement

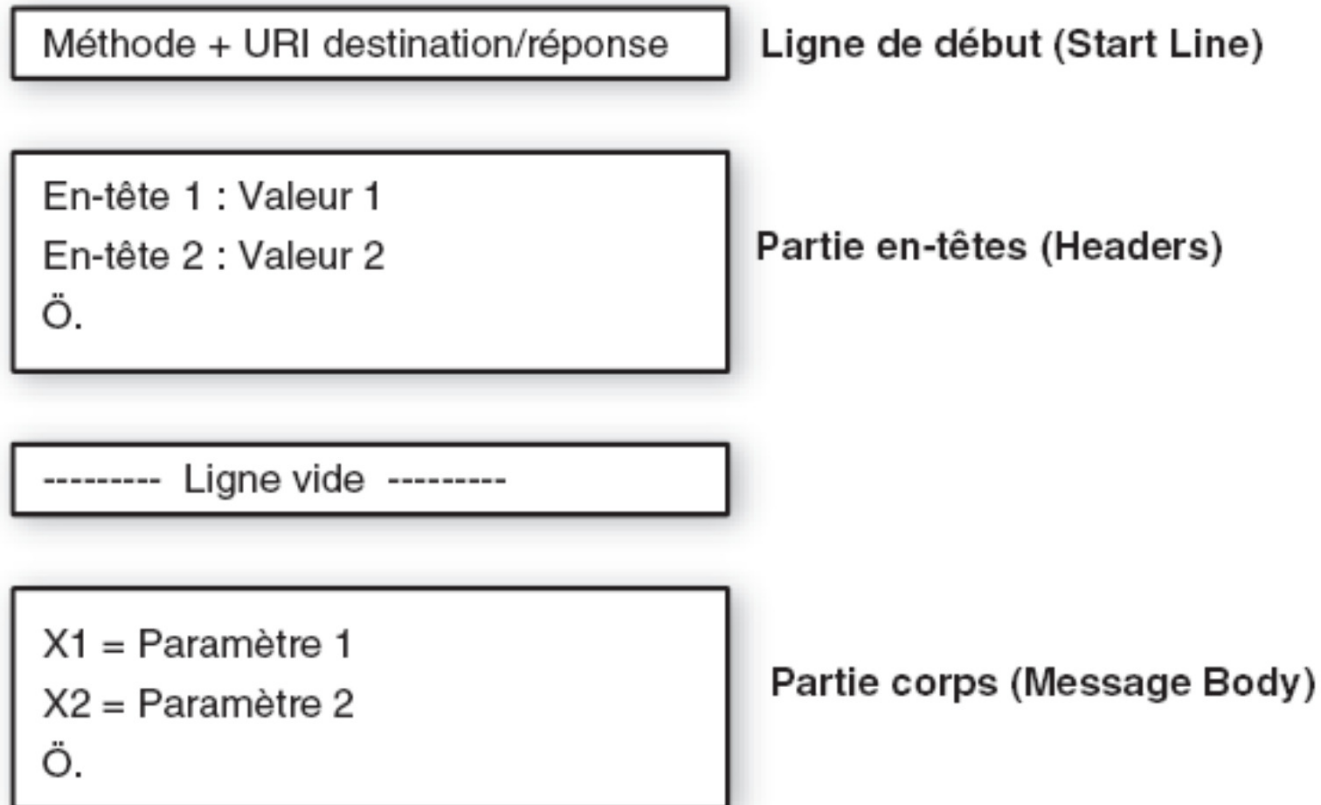
# Architecture SIP (4)

- Deux modes de communication
  - Mode direct
    - Les entités utilisatrices communiquent directement
    - SIP est un protocole de signalisation de bout en bout
    - Les serveurs ne sont pas nécessaires si les entités utilisatrices connaissent leurs adresses IPs
  - Mode indirect
    - Les entités réseaux relaient les messages échangés
    - A utiliser quand l'appelant ne connaît pas l'adresse IP de l'appelé

# Messages SIP

- Deux catégories de messages
  - Requête (i.e. méthode)
    - Les messages initialisés par les UAC (User Agent Client) à destination d'un ou de plusieurs UAS (User Agent Server)
  - Réponse
    - Les messages envoyés par les UAS pour répondre à une requête
- Encodés en langage textuel

# Structure du message SIP



# Méthodes (i.e. requêtes) SIP

- INVITE
  - Initialiser une communication en invitant un correspondant à y participer
- ACK
  - Acquitter et confirmer les paramètres de session
- BYE
  - Libérer une communication
- OPTIONS
  - Interroger un serveur SIP, y compris l'UAS, sur différentes informations (Codecs, présence)
- CANCEL
  - Annuler une requête donc la réponse n'est pas encore parvenue au demandeur
- REGISTER
  - Permettre à un utilisateur d'enregistrer son adresse IP auprès d'un serveur d'enregistrement

# Réponses SIP

- 1xx
  - Information
  - Exemple: 100 Trying
- 2xx
  - Succès
  - Exemple: 200 OK
- 3xx
  - Redirection
  - Exemple: 305 Use Proxy
- 4xx
  - Erreur côté client
  - Exemple: 400 Bad Request
- 5xx
  - Erreur côté serveur
  - Exemple: 500 Server Internal Error
- 6xx
  - Erreur globale
  - Exemple: 600 Busy Everywhere



# Adresse SIP

- Chaque utilisateur SIP a une identité SIP (i.e. une adresse SIP) sous la forme d'un URI (Uniform Resource Identifier)
- Exemple:
  - sip:alice@lip6.fr
- Une adresse SIP peut être associée à une ou plusieurs adresses IP à cause de la mobilité

# SDP (1)

- Session Description Protocol
- Une syntaxe de description de média
- Normalisé en 1988 par l'IETF
- RFC 4566
- Suit un codage textuel
- Utilisé avec SIP pour échanger les caractéristiques du média

## SDP (2)

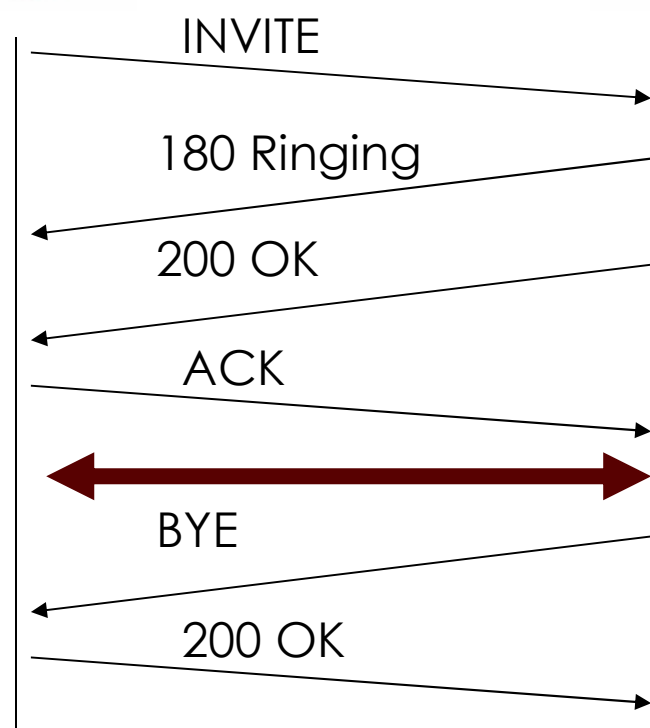
- Décrire les détails du média
  - v: version du protocole SDP
  - o: origine du message
  - s: sujet du message
  - c: connexion
  - t: temps
  - m: média
  - a: attributs

# Exemple 1 – Mode direct

tesla@high-voltage.org  
lab.high-voltage.org  
100.101.102.103



marconi@radio.org  
tower.radio.org  
200.201.202.203



# INVITE (1)

INVITE sip:marconi@radio.org SIP/2.0  
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b  
Max-Forwards: 70  
To: G. Marconi <sip:Marconi@radio.org>  
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341  
Call-ID: 123456789@lab.high-voltage.org  
CSeq: 1 INVITE  
Subject: About That Power Outage...  
Contact: <sip:n.tesla@lab.high-voltage.org>  
Content-Type: application/sdp  
Content-Length: 158

v=0  
o=Tesla 2890844526 2809844526 IN IP4 lab.high-voltage.org  
s=Phone Call  
c=IN IP4 100.101.102.103  
t=0 0  
m=audio 49170 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

# INVITE (2)

- Ligne de début
  - Nom de la méthode
    - INVITE
  - URI du destinataire
    - sip:marconi@radio.org
  - Version du protocole SIP
    - SIP/2.0
- Via:
  - Chaque élément SIP (terminal ou serveur) qui génère ou retransmet une requête SIP rajoute son adresse dans une en-tête Via
    - La version du protocole SIP et le protocole de transport utilisé (e.g. SIP/2.0/UDP)
    - Nom DNS ou adresse IP de la machine et le numéro de port (e.g. lab.high-voltage.org:5060) auxquels la réponse est envoyée
    - Transaction ID (i.e. branch) qui reste le même pour les réponses à cette requête

# INVITE (3)

- Max-Forwards:
  - Comme le champ TTL dans IP
  - Un nombre entier pour détecter les boucles
  - Diminué par chaque serveur SIP qu'il traverse
- To:
  - Adresse destinataire de la requête
- From:
  - Adresse expéditeur de la requête
  - Tag
    - Un nombre aléatoire généré par chaque participant dans un appel
    - Tag associé avec « From » est généré par l'appelant
    - Tag associé avec « To » est généré par l'appelé

# INVITE (4)

- Call-ID
  - ID de l'appel
  - Une chaîne de caractères générée par l'appelant suivi par le nom de sa machine
  - L'ensemble Call-ID + From Tag + To Tag identifie une session
- CSeq
  - Numéro de séquence de la requête
  - Incrémenté à chaque requête de même type envoyée
- Via, Max-forwards, To, From, Call-ID, CSeq sont les en-têtes obligatoires dans toutes les requêtes SIP



# INVITE (5)

- Contact
  - Obligatoire pour le message INVITE
  - SIP URI du terminal où se trouve l'UA de l'utilisateur
  - Permettre de contacter l'utilisateur directement une fois la session est établie
- Subject
  - Optionnel pour le message INVITE
  - Ne pas être utilisé par le protocole mais pouvoir être affiché sur l'écran de l'appelé pendant la sonnerie
- Content-Type
  - Spécifier le protocole utilisé pour la partie corps du message (e.g. SDP – Session Description Protocol)
- Content-Length
  - La taille de la partie corps du message (e.g. 158 octets)

# INVITE (6)

- Le corps du message contient les informations du média fournies par l'appelant
  - Adresse IP de la connexion (100.101.102.103)
  - Type du média (audio)
  - Numéro de port (49170)
  - Protocole de transport du média (RTP)
  - Codage utilisé (PCM loi  $\mu$ )
  - Fréquence d'échantillonnage (8000 Hz)

# 180 Ringing (1)

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b;received=100.101.102.103

To: G. Marconi <sip:marconi@radio.org>;tag=a53e42

From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 INVITE

Contact: <sip:marconi@tower.radio.org>

Content-Length: 0

# 180 Ringing (2)

- Réponse au message INVITE
  - Indiquer que le message INVITE est bien reçu par l'appelé et la sonnerie est en cours
- Ligne de début
  - Version du protocole SIP
    - SIP/2.0
  - Code de la réponse
    - 180
    - Réponse de type informationnel (1xx)
  - Phrase de description
    - Ringing (suggéré par le standard)
    - Tous les textes sont possibles (e.g. « Hold on, please »)

# 180 Ringing (3)

- Les en-têtes Via, To, From, Call-ID et CSeq sont copiées du message INVITE en ajoutant les informations du côté de l'appelé
  - Via
    - Ajouter le paramètre « received » qui est la même adresse de l'URI de cette en-tête Via (lab.high-voltage.org) mais sous la forme de l'adresse IP traduite par le DNS
    - Si l'URI contient déjà l'adresse IP, le paramètre « received » n'est pas nécessaire
  - To et From
    - Les valeurs des en-têtes To et From ne sont pas inversées dans le message de réponse car elles indiquent la direction de la requête
    - Un tag de l'appelé est rajouté
- Contact
  - Adresse SIP contenant l'adresse du terminal avec laquelle l'appelé peut être contacté directement une fois la session est établie

# 200 OK (1)

SIP/2.0 200 OK

Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b;received=100.101.102.103

To: G. Marconi <sip:marconi@radio.org>;tag=a53e42

From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 INVITE

Contact: <sip:marconi@tower.radio.org>

Content-Type: application/sdp

Content-Length: 155

v=0

o=Marconi 2890844528 2890844528 IN IP4 tower.radio.org

s=Phone Call

c=IN IP4 200.201.202.203

t=0 0

m=audio 6000 RTP/AVP 0

a=rtpmap:0 PCMU/8000

## 200 OK (2)

- La réponse OK est envoyée quand l'appelé décide d'accepter l'appel
- Les en-têtes Via, To, From, Call-ID et CSeq sont comme dans le message Ringing
- Le corps du message OK contient les informations de média de l'appelé

# ACK (1)

ACK sip:marconi@tower.radio.org SIP/2.0

Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bK321g

Max-Forwards: 70

To: G. Marconi <sip:marconi@radio.org>;tag=a53e42

From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341

Call-ID: 123456789@lab.high-voltage.org

CSeq 1 ACK

Content-Length: 0



# ACK (2)

- Acquittement
- Dernier message à échanger dans l'établissement d'une session
- Confirmer que l'appelant a bien reçu la réponse de l'appelé
- CSeq
  - Le même numéro de commande dans le message INVITE mais avec un autre nom de méthode
- Via
  - Le paramètre « branch » contient un autre numéro de transaction ID
  - Le message ACK est une requête et donc considéré comme une nouvelle transaction séparée de la transaction lancée par le message INVITE
  - Une transaction est considérée comme une requête suivie par les réponses à cette requête

# Terminer une session

- Un des participants accroche
  - Message BYE est envoyé comme une requête
  - Message 200 OK est envoyé comme une confirmation

# BYE (1)

BYE sip:n.tesla@lab.high-voltage.org SIP/2.0

Via: SIP/2.0/UDP tower.radio.org:5060;branch=z9hG4bK392kf

Max-Forwards: 70

To: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341

From: G. Marconi <sip:marconi@radio.org>;tag=a53e42

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 BYE

Content-Length: 0

## BYE (2)

- Remarquer que les identités dans les en-têtes To et From sont inversées
  - To: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
  - From: G. Marconi <sip:marconi@radio.org>;tag=a53e42
- C'est parce que la direction de la requête BYE est maintenant de Marconi à Tesla
- L'ensemble de {Call-ID, To-tag, From-tag} reste inchangé car il est encore toujours dans la même session
- Une nouvelle en-tête Via est générée

# 200 OK

SIP/2.0 200 OK

Via: SIP/2.0/UDP

tower.radio.org:5060;branch=z9hG4bK392kf;received=20  
0.201.202.203

To: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341

From: G. Marconi <sip:marconi@radio.org>;tag=a53e42

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 BYE

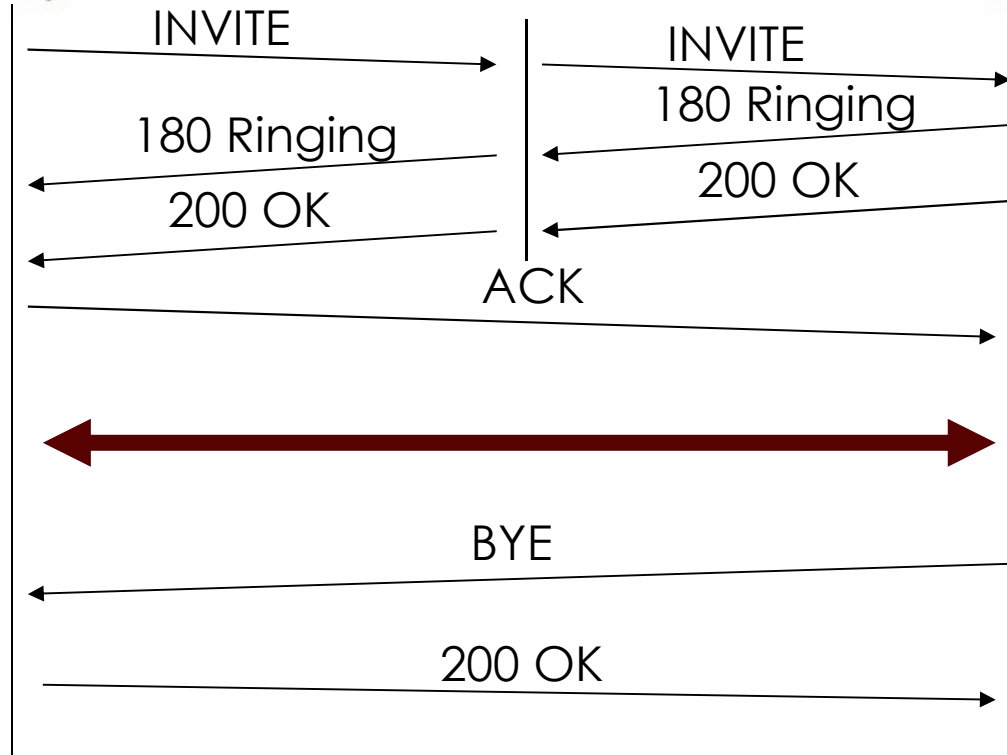
Content-Length: 0

## Exemple 2 – Mode indirect

Alice@lip6.fr  
nemo.lip6.fr  
100.101.102.103



Bob@lip6.fr  
dory.lip6.fr  
100.101.202.203



# Pourquoi un serveur proxy SIP?

- Adresse IP n'est pas fixe comme un numéro de téléphone
- Adresse IP est souvent attribuée dynamiquement à une machine via DHCP
- SIP UA de l'appelant ne connaît pas toujours l'adresse IP de l'appelé pour envoyer le message INVITE
- Serveur proxy SIP est une entité intermédiaire pour orienter les requêtes SIP vers le destinataire dans ce cas là
- Serveur proxy n'initie ni termine une session mais seulement retransmet les messages SIP
- Il est possible d'avoir plusieurs serveurs proxy dans un chemin de signalisation

# INVITE (Alice → Proxy)

INVITE sip:bob@lip6.fr SIP/2.0  
Via: SIP/2.0/UDP 100.101.102.103:5060;branch=z9hG4bKmp17a  
Max-Forwards: 70  
To: Bob <sip:bob@lip6.fr>  
From: Alice <sip:alice@lip6.fr>;tag=42  
Call-ID: 10@100.101.102.103  
CSeq: 1 INVITE  
Subject: Where are you?  
Contact: <sip:alice@nemo.lip6.fr>  
Content-Type: application/sdp  
Content-Length: 159

v=0  
o=alice 2890844526 2890844526 IN IP4 100.101.102.103  
s=Phone Call  
t=0 0  
c=IN IP4 100.101.102.103  
m=audio 49170 RTP/AVP 0  
a=rtpmap:0 PCMU/8000



# Traitement du message INVITE au serveur proxy

- Traduire adresse SIP du destinataire  
sip:bob@lip6.fr en adresse IP de Bob  
(100.101.202.203)
- Transmettre le message INVITE à Bob
  - Rajouter une en-tête Via dans le message
  - L'en-tête Via permet à la réponse d'être envoyée  
sur le même chemin de la requête
  - Diminuer la valeur de Max-Forwards

# INVITE (Proxy → Bob)

INVITE sip:bob@100.101.202.203 SIP/2.0  
Via: SIP/2.0/UDP proxy.lip6.fr:5060;branch=z9hG4bK83842.1  
Via: SIP/2.0/UDP 100.101.102.103:5060;branch=z9hG4bKmp17a  
Max-Forwards: 69  
To: Bob <sip:bob@lip6.fr>  
From: Alice <sip:alice@lip6.fr>;tag=42  
Call-ID: 10@100.101.102.103  
CSeq: 1 INVITE  
Subject: Where are you?  
Contact: <sip:alice@nemo.lip6.fr>  
Content-Type: application/sdp  
Content-Length: 159

v=0  
o=alice 2890844526 2890844526 IN IP4 100.101.102.103  
s=Phone Call  
t=0 0  
c=IN IP4 100.101.102.103  
m=audio 49170 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

# 180 Ringing (Bob → Proxy)

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP

proxy.lip6.fr:5060;branch=z9hG4bK83842.1;received=100.101.102.105

Via: SIP/2.0/UDP 100.101.102.103:5060;branch=z9hG4bKmp17a

To: Bob <sip:bob@lip6.fr>;tag=314159

From: Alice <sip:alice@lip6.fr>;tag=42

Call-ID: 10@100.101.102.103

CSeq: 1 INVITE

Contact: <sip:bob@100.101.202.203>

Content-Length: 0

# 180 Ringing (Proxy → Alice) (1)

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP  
100.101.102.103:5060;branch=z9hG4bKmp17a

To: Bob <sip:bob@lip6.fr>;tag=314159

From: Alice <sip:alice@lip6.fr>;tag=42

Call-ID: 10@100.101.102.103

CSeq: 1 INVITE

Contact: <sip:bob@100.101.202.203>

Content-Length: 0

## 180 Ringing (Proxy → Alice) (2)

- Le serveur proxy reconnaît son adresse dans la première en-tête Via
- Il utilise le transaction ID (i.e. le paramètre « branch ») pour identifier la transaction, enlever son en-tête Via, et retransmet le message à l'adresse IP indiquée dans l'en-tête Via suivante
- L'en-tête Via simplifie le routage des réponses SIP dans les serveur proxy car l'adresse IP du prochain nœud est déjà indiquée dans le message

# 200 OK (Bob → Proxy)

SIP/2.0 200 OK

Via: SIP/2.0/UDP proxy.lip6.fr:5060;branch=z9hG4bK83842.1;received=100.101.102.105

Via: SIP/2.0/UDP 100.101.102.103:5060;branch=z9hG4bKmp17a

To: Bob <sip:bob@lip6.fr>;tag=314159

From: Alice <sip:alice@lip6.fr>;tag=42

Call-ID: 10@100.101.102.103

CSeq: 1 INVITE

Contact: <sip:bob@100.101.202.203>

Content-Length: 159

v=0

o=bob 2890844526 2890844526 IN IP4 100.101.202.203

s=Phone Call

t=0 0

c=IN IP4 100.101.202.203

m=audio 49172 RTP/AVP 0

a=rtpmap:0 PCMU/8000

# 200 OK (Proxy → Alice)

SIP/2.0 200 OK

Via: SIP/2.0/UDP 100.101.102.103:5060;branch=z9hG4bKmp17a

To: Bob <sip:bob@lip6.fr>;tag=314159

From: Alice <sip:alice@lip6.fr>;tag=42

Call-ID: 10@100.101.102.103

CSeq: 1 INVITE

Contact: <sip:bob@100.101.202.203>

Content-Length: 159

v=0

o=bob 2890844526 2890844526 IN IP4 100.101.202.203

s=Phone Call

t=0 0

c=IN IP4 100.101.202.203

m=audio 49170 RTP/AVP 0

a=rtpmap:0 PCMU/8000

## ACK (Alice → Bob) (1)

ACK sip:bob@lip6.fr SIP/2.0

Via: SIP/2.0/UDP

100.101.102.103:5060;branch=z9hG4bKka42

Max-Forwards: 70

To: Bob <sip:bob@lip6.fr>;tag=314159

From: Alice <sip:alice@lip6.fr>;tag=42

Call-ID: 10@100.101.102.103

CSeq 1 ACK

Content-Length: 0

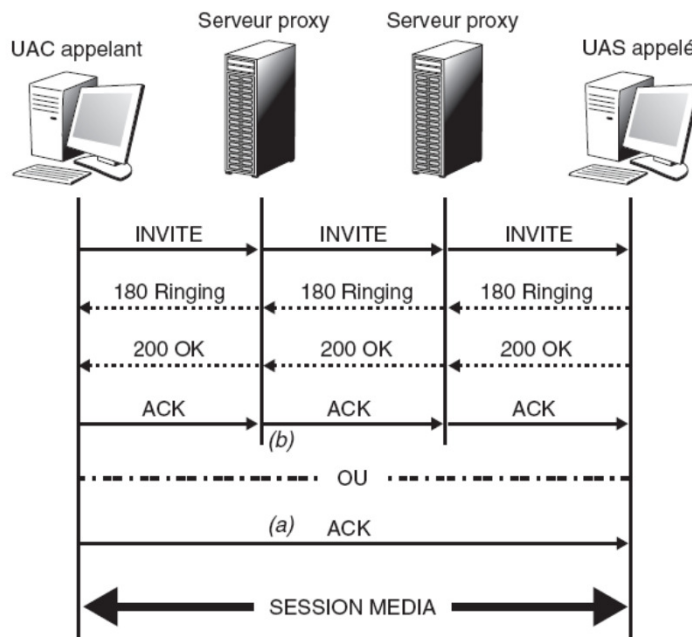


## ACK (Alice → Bob) (2)

- La requête ACK est envoyée directement à Bob cette fois ci, sans passer la le serveur proxy, car Alice connaît déjà l'adresse IP de Bob grâce à l'en-tête « Contact » dans la réponse de Bob

# ACK passé par le serveur

- Il est possible que le serveur proxy force tous les messages à passer par lui



- (a) Le message ACK est envoyé directement à l'appelé
- (b) Le message ACK est forcé à passer par le serveur proxy

## BYE (Bob → Alice)

BYE sip:alice@nemo.lip6.fr SIP/2.0

Via: SIP/2.0/UDP 100.101.202.203:5060;  
branch=z9hG4bK4332

Max-Forwards: 70

To: Alice <alice@lip6.fr>;tag=42

From: Bob <bob@lip6.fr>;tag=314159

Call-ID: 10@100.101.102.103

CSeq: 2000 BYE

Content-Length: 0

# OK (Alice → Bob)

SIP/2.0 200 OK

Via: SIP/2.0/UDP 100.101.202.203:5060;  
branch=z9hG4bK4332

To: Alice <alice@lip6.fr>;tag=42

From: Bob <bob@lip6.fr>;tag=314159

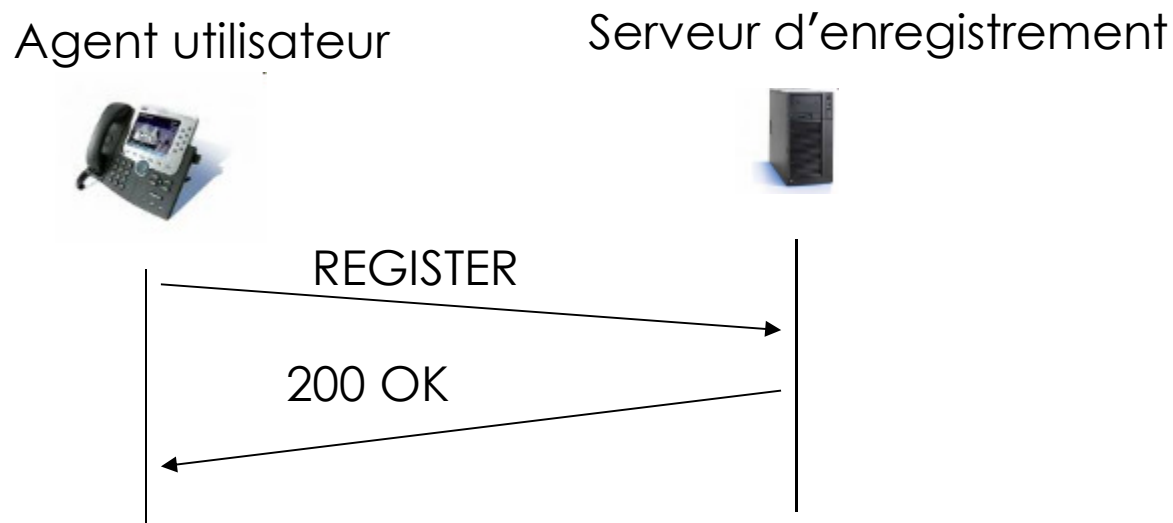
Call-ID: 10@100.101.102.103

CSeq: 2000 BYE

Content-Length: 0

# Procédure d'enregistrement

- L'UA enregistre la correspondance entre l'adresse IP et l'adresse SIP auprès d'un serveur d'enregistrement
- Le serveur d'enregistrement maintient une base de données qui sera consultée par les serveur proxy pour router les requêtes SIP vers le destinataire



# REGISTER (1)

REGISTER sip:registrar.lip6.fr SIP/2.0

Via: SIP/2.0/UDP

100.101.202.203:5060;branch=z9hG4bKus19

Max-Forwards: 70

To: Bob <sip:bob@lip6.fr>

From: Bob <sip:bob@lip6.fr>;tag=3431

Call-ID: 23@100.101.202.203

CSeq: 1 REGISTER

Contact: sip:bob@100.101.202.203

Content-Length: 0

## REGISTER (2)

- L'en-tête To contient l'adresse SIP de l'utilisateur
- L'en-tête Contact contient l'adresse IP où l'utilisateur peut être joint directement
- L'enregistrement est souvent lancé automatiquement dès le démarrage d'un terminal SIP

# 200 OK (1)

SIP/2.0 200 OK

Via: SIP/2.0/UDP

100.101.102.103:5060;branch=z9hG4bKus19

To: Bob <sip:bob@lip6.fr>;tag=8771

From: Bob <sip:bob@lip6.fr>;tag=3431

Call-ID: 23@100.101.202.203

CSeq: 1 REGISTER

Contact: <sip:bob@dory.lip6.fr>;expires=3600

Content-Length: 0



## 200 OK (2)

- Paramètre `expires` [seconde]
  - Ex: 3600 secondes (i.e. une heure)
- Bob doit renouveler l'enregistrement avant son expiration
- Plusieurs terminaux (i.e. plusieurs adresses IP) peuvent être enregistrés pour une même adresse SIP
- Le serveur proxy peut retransmettre une requête à un ou plusieurs terminaux enregistrées
- D'autres opérations permettent d'annuler un enregistrement ou de demander une liste des terminaux actuellement enregistrés