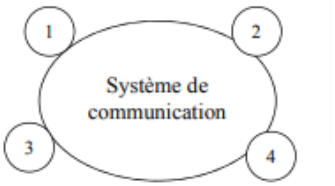
Chapitre 1 : Introduction aux Systèmes Répartis

1/Systèmes centralisés vs systèmes répartis :

- Les systèmes centralisés exécutent tout sur une seule machine.
- Les systèmes répartis utilisent plusieurs ordinateurs connectés en réseau, apparaissant comme une seule entité pour l'utilisateur.

2/Caractéristiques des systèmes répartis :



- Communication via des messages.
- Pas de mémoire ou horloge commune entre les machines.
- Variabilité des délais de transmission.
- Machines indépendantes prenant des décisions locales.
- La panne d'une machine n'empêche pas le déroulement de l'algorithme.

3/Intérêts des systèmes répartis :

- Partage des ressources distantes.
- Optimisation des ressources.
- Redondance pour améliorer la disponibilité.
- Exécution parallèle pour accélérer les traitements.

4/Critères de conception des systèmes répartis :

1. **Transparence :**
   - **Transparence à l'emplacement (à la localisation) :** L'utilisateur ne doit pas savoir où se trouvent les ressources.
   - **Transparence à l'accès :** L'accès aux ressources locales ou distantes doit se faire de la même manière.
   - **Transparence à la concurrence :** Les utilisateurs doivent penser qu'ils sont seuls à utiliser les ressources, même en cas d'accès simultané.
   - **Transparence à la migration :** Les ressources peuvent être déplacées sans changer leur nom ou leur chemin d'accès.
   - **Transparence à la duplication :** L'utilisateur n'est pas informé des copies multiples de données.
   - **Transparence au parallélisme :** Les processus peuvent s'exécuter en parallèle sans que l'utilisateur le sache.
   - **Transparence à l'extension :** L'ajout de ressources matérielles ou logicielles doit être invisible pour l'utilisateur.

2. **Synchronisation des activités :**

   Il faut synchroniser les activités sur différents sites, même sans connaître l'état global du système en temps réel.

3. **Gestion de l'information répartie,cohérence :**

   La gestion des copies multiples d'une même information doit rester cohérente pour garantir sa disponibilité.

4. **Fiabilité,disponibilité et tolérance aux fautes :**

   Le système doit être plus fiable et disponible qu'un système monoprocesseur, capable de fonctionner malgré des pannes.

5. **Dimensionnement et facteur d'échelle :**

   Le système doit s'adapter automatiquement aux changements de taille (nombre de sites ou d'applications).

5/Fiabilité :

De nombreux points de pannes peuvent survenir et à plusieurs niveaux :

Réseau :

- Une partie du réseau peut devenir inaccessible, rendant certaines ressources inaccessibles.
- Les temps de communication peuvent varier en fonction de la charge sur le réseau.
- Des données peuvent être perdues pendant leur transmission.

Machine :

- Une ou plusieurs machines peuvent tomber en panne, provoquant une paralysie partielle ou totale du système.

Serveur :

- Si le serveur plante, plus rien ne fonctionne.
- Le serveur peut devenir un goulot d'étranglement si le volume d'informations à traiter est trop important.

6/Sécurité :

La sécurité est plus vulnérable qu'en système centralisé à cause de plusieurs risques :

- Les communications réseau peuvent être interceptées.
- Il est difficile de bien connaître l'identité de l'entité distante avec laquelle on communique.
- Les messages échangés peuvent être interceptés et modifiés en cours de route.

7/Exemples de systèmes repartis :

1/Serveur de fichiers :

- **Composition** : Plusieurs clients et un serveur de fichiers où les fichiers sont stockés.
- **Avantage** : Les fichiers sont accessibles depuis n'importe quelle machine cliente.
- **Inconvénient** : Si le réseau ou le serveur tombe en panne, l'accès aux fichiers est impossible.

2/Calculs scientifiques :

- **Composition** : Un réseau de machines identiques ou hétérogènes, soit localement soit via Internet.
- **Principe** : Un serveur distribue des calculs aux clients, qui exécutent les tâches et renvoient les résultats au serveur.
- **Avantage** : Maximisation de l'utilisation des ressources de calcul.
- **Inconvénient** : Si le réseau ou le serveur plante, l'ensemble du système s'arrête.

Chapitre 2 : Synchronisation et Temps Logique

1/Introduction :

La synchronisation entre les tâches est réalisée grâce à l'échange de messages entre les différents sites d'un système réparti.Un événement est défini comme un changement d'état local ou l'envoi/réception d'un message par un processus.

2/Ordonnancement des événements et précédence causale :

L'ordonnancement dans un système réparti repose sur une relation de précédence causale entre les événements, notée « a → b ». Cette relation s'applique lorsque :

- Les événements (a) et (b) se produisent sur le même site, et (a) précède (b) chronologiquement.
- (a) correspond à l'envoi d'un message et (b) à la réception de ce message.
- Il existe une chaîne d'événements reliant (a) à (b) via plusieurs événements intermédiaires.

Remarque : Cette relation est un ordre partiel, c'est-à-dire qu'elle ne couvre pas tous les événements du système.

Définition de l'historique d'un événement : L'ensemble des événements précédant causalement un événement, y compris lui-même.

Définition de l'indépendance des événements : Deux événements sont dits indépendants s'ils ne se précèdent pas causalement.Cette relation est notée"||".

Propriétés : Soient **a** et **b** deux événements :

1) a → b ⇔ a ∈ historique(b)
   a | b ⇔ (a ∉ historique(b) et b ∉ historique(a)
2) si, ni a → b
   ni b → a , a et b sont dits indépendants ou concurrents.

3/Ordonnancement par estampilles et horloges logiques linéaires :

Dans l'ordonnancement par horloges logiques de Lamport :

- Chaque site SiS_iSi possède une horloge hih_ihi qui est incrémentée lors d'un événement local ou lorsqu'un message est émis. Le message est estampillé avec (hi,i)(h_i, i)(hi,i).
- Lorsqu'un site SjS_jSj reçoit un message avec l'estampille (hi,i)(h_i, i)(hi,i), son horloge hjh_jhj est mise à jour à max(hj,hi)+1\max(h_j, h_i) + 1max(hj,hi)+1.
- Cela permet d'établir un ordre partiel des événements et, sous certaines conditions, un ordre total en utilisant les valeurs et l'indice des sites.

Propriétés :

1. Si un événement **e** est daté par **h** alors **(h-1)** représente le nombre d'événements qui se sont produits avant **e** sur le chemin de causalité le plus long menant à **e**.
2. $\forall a, b$ deux événements:

$$a \rightarrow b \Rightarrow date(a) < date(b)$$

$$\text{Mais } date(a) < date(b) \not\Rightarrow a \rightarrow b$$

4/Horloges vectorielles (Mattern 1989) :

Les horloges vectorielles permettent de caractériser la causalité entre événements dans un système réparti. Chaque site SiS_iSi possède un vecteur d'entiers ViV_iVi, initialisé à zéro.

Mise à jour :

1. Lors d'un événement local, Vi[i]V_i[i]Vi[i] est incrémenté.
2. Lorsqu'un message est émis, Vi[i]V_i[i]Vi[i] est incrémenté, et le message est estampillé avec (Vi,i)(V_i, i)(Vi,i).
3. Lors de la réception d'un message, le site récepteur met à jour son vecteur en prenant le maximum des valeurs de chaque composante.

Propriétés **:** Soient Vh et Vk deux valeurs d'horloges vectorielles de longueur n.

1. $Vh \leq Vk \Leftrightarrow \forall x : Vh[x] \leq Vk[x] \quad 1 \leq x \leq n$
2. $Vh < Vk \Leftrightarrow Vh \leq Vk \text{ et } \exists x : Vh[x] < Vk[x] \quad 1 \leq x \leq n$
3. $Vh \,||\, Vk \Leftrightarrow \neg(Vh < Vk) \text{ et } \neg(Vk < Vh)$
4. Si on a deux événements a et b estampillés respectivement vh et vk alors :
   $a \rightarrow b \Leftrightarrow Vh < Vk$ autrement dit : $a \rightarrow b \Leftrightarrow date(a) < date(b)$
   $a \,||\, b \Leftrightarrow Vh \,||\, Vk$
5. Violation de l'ordre de délivrance causale dans la réception des messages.

5/Horloges et estampilles matricielles :

5.1/Délivrance causale : La délivrance d'un message consiste à le rendre accessible aux applications clientes, et peut être retardée pour garantir un ordre précis.

- **Propriété 1 (FIFO)** : Si deux messages sont envoyés successivement d'un site Si à un destinataire Sj, le premier message sera délivré avant le second.
- **Propriété 2 (Ordre de délivrance causale)** : Si l'envoi d'un message m1 précède causalement l'envoi d'un message m2, alors m1 sera délivré avant m2 sur le site destinataire.

5.2/Principe des horloges matricielles(Raynal M; Schiper A; Toueg S) :

Les horloges matricielles utilisent une matrice carrée Mi pour chaque site Si, où chaque élément représente le nombre d'événements sur iSi, ainsi que les messages reçus et connus par Si en provenance des autres sites.

Mise à jour :

1. Lors d'un événement local sur Si, Mi[i,i]est incrémenté.
2. Lorsqu'un message est envoyé, les compteurs Mi[i,i] et Mi[j,i]sont incrémentés et estampillés.
3. Lorsqu'un message est reçu, deux conditions doivent être vérifiées (ordre FIFO et réception des messages antérieurs) pour garantir la causalité

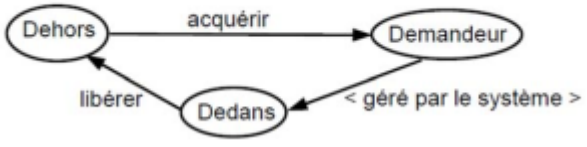
- **Condition 1 :** $EM_m[j,i] = M_i[j,i] + 1$ (ordre FIFO sur le canal (j,i))
- **Condition 2 :** **pour tout** $k \neq i, k \neq j$, $EM_m[k,i] \leq M_i[k,i]$ (tous les messages des sites différents de S_j ont été reçus).

➢ Si les deux conditions sont vérifiées, le message est délivrable et l'horloge du site S_i est mise à jour :
- $M_i[i,i]$++ (incrémentation);
- $\forall k \; et \; \forall l$, $M_i[k,l] = \max(M_i[k,l], EM_m[k,l])$

➢ **Sinon**, la délivrance du message est différée jusqu'à ce qu'elles deviennent vérifiées.

6/Exemple d'application des horloges:Exclusion mutuelle :

Diagramme d'états de l'accès en exclusion mutuelle :



Algorithme de Ricart et Agrawala pour l'exclusion mutuelle :

Pour entrer en section critique, un processus Pi envoie une requête datée à tous les autres. Lorsqu'un processus Pi reçoit une requête de Pj, il répond par un accord si :

1. Il n’est pas demandeur, ou
2. S’il est demandeur mais avec une date antérieure à celle de Pj; sinon, il diffère sa réponse.

Lorsque Pi sort de la section critique, il envoie un message de libération aux processus dont les requêtes étaient en attente. Un processus peut entrer en section critique une fois qu’il a reçu la permission de tous.

Chapitre 3 : Le modèle peer to peer

1. Introduction et contexte d'apparition des réseaux P2P

- Augmentation des transferts de fichiers a révélé les limites du modèle client/serveur.
- Les architectures P2P permettent à chaque nœud d'agir comme client et serveur simultanément.
- **Napster** (1999) est la première application P2P pour le partage de fichiers musicaux, suivie d'autres applications comme BitTorrent, eDonkey, et Skype.

2. Caractéristiques des réseaux P2P

- Chaque utilisateur partage des ressources directement avec d'autres sans passer par un serveur central.
- Propriétés principales :
  - **Auto-organisation** : S'adapte aux changements dans le réseau.
  - **Dynamisme** : Les nœuds peuvent rejoindre ou quitter librement.
  - **Passage à l'échelle** : Peut gérer un grand nombre d'utilisateurs.
  - **Forte disponibilité** : Les ressources restent accessibles malgré des pannes.

3. Applications des réseaux P2P

- **Partage de fichiers** : Napster, Gnutella, KaZaA, BitTorrent.
- **Communication** : Messagerie instantanée, vidéoconférence, téléphonie Internet.
- **Collaboration** : Travaux en temps réel sur des projets distribués (ex : Groove).
- **Traitement distribué** : Exploitation des capacités de calcul parallèles (ex : SETI@home, BOINC).

4. Avantages et contraintes

Avantages :

- Échanges rapides et directs entre clients.

- Optimisation de la bande passante et équilibrage de la charge réseau.
- Faibles coûts de maintenance.
- Résistance aux pannes grâce à la réplication des ressources.
- Passage à l'échelle par rapport au modèle client/serveur.

Contraintes :

- Sécurité faible lors des échanges.
- Risque de fichiers trompeurs ou de contenus non fiables.
- Problèmes de droits d'auteur et de copyright.

5. Taxonomie des systèmes P2P

- **Selon l'indexation :**
  - **Index centralisé** : Une machine conserve toutes les données (ex : Napster).
  - **Index local** : Chaque machine conserve ses propres données (ex : Gnutella initial).
  - **Index distribué** : Données réparties sur plusieurs machines (ex : systèmes DHT comme Chord).
- **Selon la décentralisation :**
  - **Modèle hybride** : Partiellement décentralisé.
  - **Modèle pur** : Totalement décentralisé.
- **Selon la structuration :**
  - **Non structuré** : Nœuds connectés librement (ex : Napster, Gnutella, BitTorrent).
  - **Structuré** : Connexion suivant des règles spécifiques basées sur l'identité (ex : Chord).

6. Modèle Hybride:

6.1 Napster

- **Fonctionnement :**
  1. Un **serveur central** met en relation les pairs connectés en gérant l'indexation des fichiers partagés.
  2. Les pairs annoncent leurs ressources au serveur lors de leur connexion.
  3. Recherche d'une ressource :
     - Le pair envoie une requête au serveur.
     - Le serveur fournit une liste des pairs possédant le fichier.
     - Le fichier est téléchargé directement entre pairs.

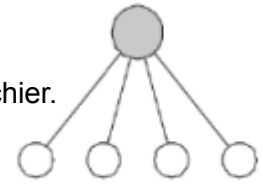
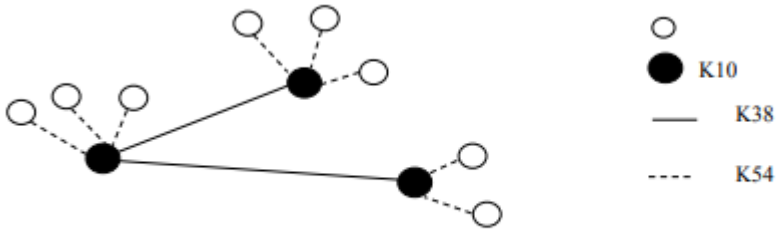


Figure 1 : Topologie du réseau Napster

6.2 Kazaa

- Fonctionne avec des **SuperPairs** (ou SuperPeers), des nœuds ayant une connexion rapide et un processeur puissant.
- Les SuperPairs forment une architecture décentralisée et gèrent les **clusters** (groupes de pairs connectés).
- Les serveurs principaux se limitent à la gestion des connexions et à fournir l'adresse d'un SuperPair aux nouveaux nœuds.



K24

6.3 Avantages et limites

Avantages :

- Administration facile grâce au serveur central.
- Mise en place de mécanismes d'authentification et de contrôle d'accès possible.
- Recherche rapide et peu coûteuse, sans routage.

Limites :

- Vulnérabilité accrue au **Déni de Service (DoS)** en raison de la dépendance au serveur central.
- Problème de **disponibilité** et de **passage à l'échelle** : le serveur est sollicité pour chaque opération (insertion, recherche, etc.).
- Absence de garanties concernant l'**anonymat** des utilisateurs.

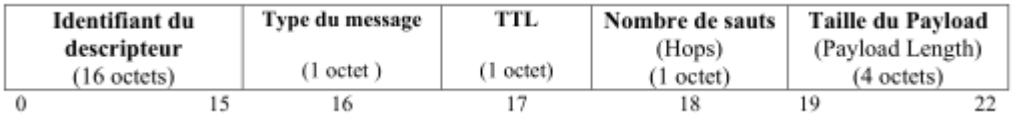
7. Modèle pur :

Caractéristiques générales :

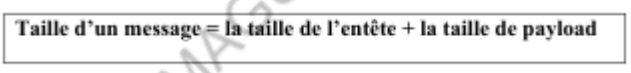
- Aucun serveur n'est utilisé.
- Chaque pair agit à la fois comme client et serveur (appelé **servent**).
- Exemple notable : **Gnutella v0.4**, développé en mars 2000 par Justin Frankel et Tom Pepper.

7.1 Types et formats des messages

Gnutella utilise **5 types de messages** (appelés *descripteurs*), avec un entête commun de 5 champs (23 octets) :



1. **Types de messages** :
    - **Ping** : Identifier les pairs connectés (Type = `0×00`).
    - **Pong** : Réponse à un Ping avec les informations des pairs (Type = `0×01`).
    - **Query** : Requête pour rechercher des ressources (Type = `0×80`).
    - **QueryHit** : Réponse à une Query indiquant la disponibilité d'une ressource (Type = `0×81`).
    - **Push** : Demande pour initier un transfert de fichier (Type = `0×40`).
2. **Champs de l'entête** :
    - **Descriptor ID** : Identifiant unique pour chaque descripteur dans le réseau Gnutella.
    - **Type du message (Payload Descriptor)** : Définit la nature du message (Ping, Pong, etc.).
    - **TTL (Time To Live)** : Diminue à chaque passage par un nœud ; limite le nombre de sauts avant destruction.
    - **Hops (Nombre de sauts)** : Incrémente à chaque passage par un nœud.
    - **Payload Length** : Taille des données associées au message.

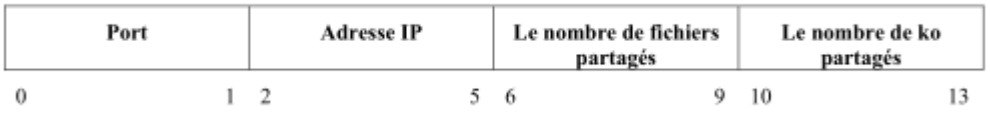


Ping (0×00) :

- Permet de découvrir les autres *servents* dans le réseau.
- Ne contient **aucun payload**, seulement un entête.

Pong (0×01) :

- Réponse à un message Ping.
- Payload de 14 octets structuré en 4 champs :
    - **Port** : Numéro de port où le servent peut accepter des connexions.
    - **Adresse IP** : Adresse IP du servent en hexadécimal (4 octets).
        - Exemple : IP `200.15.81.50` est représentée comme `0×C80F5132`.



Query (0×80) :

- Utilisé pour rechercher un fichier.
- Payload de taille variable avec deux informations :
    - **Vitesse minimum** : Seuil minimal de vitesse de transmission (en Ko/s). Les *servents* plus lents ne répondent pas.
    - **Critères de recherche** : Définis par l'utilisateur, se terminent par un caractère NULL.

Vitesse minimum	Critères de recherche \0
0 1	2 …

QueryHit (0×81) :

1. **Description générale** :
    - Ce message est une **réponse** à un message Query.
    - Contient des informations sur les résultats de la recherche.
2. **Champs principaux** :
    - **Nombre d'enregistrements (Number of hits)** : Nombre de résultats inclus dans le message.
    - **Port** : Numéro de port où le *servent* identifié accepte les connexions.
    - **Adresse IP** : Adresse IP du *servent* identifié.
    - **Vitesse de réponse (Speed)** : Vitesse du *servent* en ko/s.

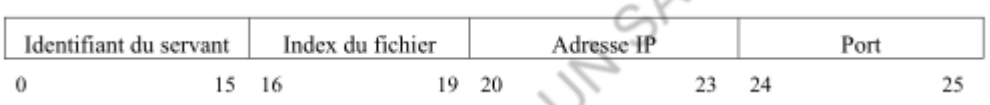
Nombre d'enregistrements	Port	Adresse IP	Vitesse de réponse (ko/s)	Réponse à QUERY	Identifiant du serveur
0	1 2	3 6	7 10	11…..	n n+16

3. **Réponse à Query (Result Set)** :
    - Ensemble de résultats correspondant à la requête.
    - Contient les informations suivantes pour chaque résultat :
        - **Index du fichier (File Index)** : Numéro unique attribué par le *servent* pour identifier le fichier.
        - **Taille du fichier (File Size)** : Taille du fichier en ko.
        - **Nom du fichier (File Name)** : Nom du fichier, se termine par `0x0000` et sa taille est limitée par le champ Payload de l'entête.
        - **Identifiant du *servent* (Servent identifier)** : Chaîne de 16 octets identifiant le *servent* répondant à la requête.

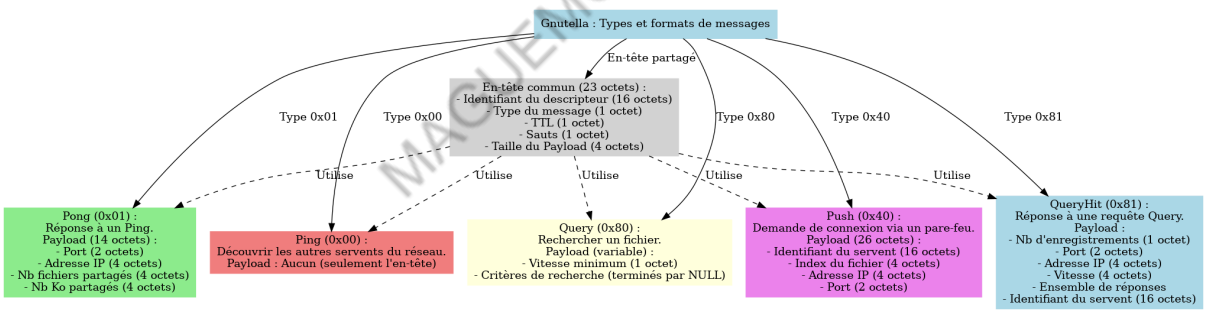
Index du fichier	Taille du fichier	Nom du fichier
0 3	4 7	8 …..

Push (0×40) :

1. **Description générale** :
    - Ce message est utilisé pour **demander une connexion à travers un firewall**.
    - Un servent situé **à l'extérieur du firewall** envoie ce message lorsqu'il reçoit un QueryHit d'un servent situé **derrière un firewall**.
    - Objectif : Demander au servent cible de **démarrer une connexion HTTP**.
2. **Payload (26 octets, 4 champs)** :
    - **Identifiant du servent (Servent identifier)** :
        - Identifie le servent qui doit pousser le fichier demandé.
        - Correspond à l'identifiant du servent dans le QueryHit correspondant.
    - **Index du fichier (File Index)** :
        - Spécifie le fichier à "pousser".
        - Correspond à l'Index du fichier mentionné dans le QueryHit.
    - **Adresse IP** :
        - Adresse IP du servent qui envoie le message Push (celui qui reçoit le fichier).
    - **Port** :
        - Port du servent destinataire où le fichier doit être "poussé".



Résumé général:



7.2 Connexion au réseau :

1. **Connexion initiale** :
    - Un pair doit se connecter à un servent déjà présent sur le réseau.
    - Pour cela, il utilise les **GWebcaches**, qui sont des scripts web fournissant une liste de nœuds Gnutella potentiels.
2. **Commande d'établissement de connexion** :
    - Le pair utilise les commandes **Connect** et **OK** pour établir une connexion avec le premier servent.
3. **Découverte des servents** :
    - Une fois connecté, il envoie des messages **Ping** via un processus d'inondation pour découvrir d'autres nœuds.
    - Les servents répondent par des **Pong**, indiquant leur adresse IP, leur port, le nombre de fichiers, et la taille totale des données partagées.

7.3 Recherche d'un fichier :

Envoi de la requête :

- Un servent envoie une requête **Query** aux servents auxquels il est connecté.
- Ceux-ci retransmettent la requête par inondation à leurs voisins.

Réponse à la requête :

- Lorsqu'un servent possédant le fichier reçoit la requête, il répond par un **QueryHit**.
- Le QueryHit contient l'adresse IP, le numéro de port, et des informations sur le fichier demandé.

Transfert de fichier :

- Après réception du QueryHit, le servent demande le transfert du fichier.
- Le téléchargement se fait directement entre les pairs via le protocole **HTTP**.

8. Les systèmes pair à pair structurés (chord) :

- Chord repose sur les **tables de hachage distribuées (DHT)**.
- Utilise la fonction de hachage **SHA-1** (160 bits) pour attribuer des identifiants aux nœuds (via leurs adresses IP) et aux ressources (via leurs noms).

8.1. Placement des ressources :

Les nœuds et les ressources sont organisés dans un **anneau virtuel**.
Les ressources sont placées sur le nœud correspondant à la fonction **successeur(k)**, qui retourne le premier nœud réel dont l'identifiant est supérieur ou égal à la clé (k) sur l'anneau.
La figure illustre un anneau Chord pour **m = 6**, représentant les identifiants sur une plage de 0 à 63.

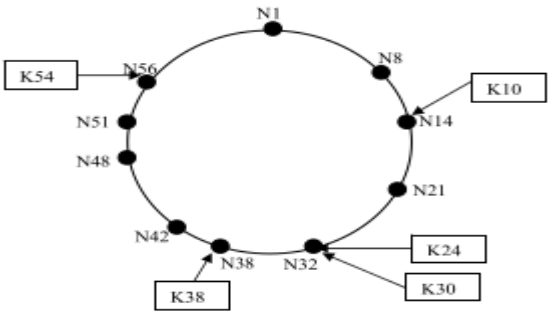


Figure 3 : Placement des ressources dans Chord

8.2. Recherche d'une ressources :

Principe général

1. **Étape de hachage** :
    - Le nœud demandeur hache le nom de la ressource (**nomR**) pour obtenir une clé **k** : k=H(nomR)k = H(nomR)k=H(nomR).
2. **Contacter le successeur(k)** :
    - Le nœud doit trouver l'adresse IP du successeur qui détient la ressource correspondant à la clé **k**.

Solution 1 : Recherche naïve

1. **Méthode** :
    - Chaque nœud connaît uniquement l'adresse IP de son successeur immédiat sur l'anneau.
    - Le nœud demandeur envoie une requête **recherche(k)** contenant son adresse IP au nœud courant.
2. **Processus** :
    - Si le nœud courant possède la clé **k**, il retourne les informations de la ressource au demandeur.
    - Sinon, il transmet la requête à son successeur immédiat dans l'anneau, jusqu'à atteindre le bon nœud.
3. **Inconvénient** :
    - La recherche est lente, car elle nécessite de propager la requête de nœud en nœud.

Solution 2 : Recherche améliorée avec table de repérage (Finger Table)

1. **Utilisation de la Finger Table** :
    - Chaque nœud maintient une table de routage appelée **Finger Table**, avec **m** entrées (où mmm est la taille de l'espace d'identifiants, par ex. 160 pour SHA-1).
2. **Structure de la table** :
    - La ie`mei^{ème}ie`me entrée contient le premier nœud réel sss situé au moins 2i−12^{i-1}2i−1 positions après le nœud courant nnn, défini par : s=successeur((n+2i−1)mod 2m)s = successeur((n + 2^{i-1}) \mod 2^m)s=successeur((n+2i−1)mod2m) avec 1≤i≤m1 \leq i \leq m1≤i≤m.
    - Chaque entrée stocke l'identifiant et l'adresse IP de ce nœud.
3. **Avantage** :
    - Réduit considérablement le temps de recherche en évitant un parcours linéaire de l'anneau.
    - La recherche utilise la Finger Table pour diriger rapidement la requête vers le nœud approprié.

Chapitre 4 : Les réseaux sans fils et les réseaux de mobiles

1. Introduction et contexte d'apparition des réseaux sans fils:

Les réseaux sans fil sont apparus grâce à l'évolution rapide des télécommunications pour répondre à des besoins tels que :

- L'utilisation accrue des terminaux portables en milieux publics.
- La mobilité et l'accès permanent aux systèmes d'information.
- La réalisation d'installations temporaires et rapides.
- La suppression du câblage, notamment pour les bâtiments historiques.

2.Les réseaux sans fil :

Un réseau sans fil utilise des moyens comme l'infrarouge ou les ondes radio pour communiquer sans câbles. Cependant, il présente des limites, notamment :

- Débit réduit.
- Atténuation rapide du signal.
- Erreurs, interférences et faible sécurité physique.
- Problèmes des nœuds cachés et exposés.

3. Classification des réseaux sans fil selon la zone de couverture :

Les réseaux sans fil se divisent en quatre catégories selon la portée.

3.1. Réseaux personnels sans-fil (WPAN:Wireless Personal Area Network) :

- Très faible portée (environ 10 mètres).
- Utilisations principales :
    - Communication entre équipements portés par une personne.
    - Liaison sans fil entre périphériques (exemple : imprimante et ordinateur).
    - Connexion entre machines très proches.
- Principales technologies : Bluetooth, infrarouge, ZigBee, HomeRF.

3.2. Réseaux locaux sans-fil (WLAN:Wireless Local Area Network) :

- Portée d'environ 100 mètres.
- Utilisés dans : entreprises, universités, particuliers.
- Normes principales : IEEE 802.11 (WiFi) avec variantes 802.11a, 802.11b, 802.11g.
- Autres technologies : hiperlan1, hiperlan2.

3.3. Réseaux métropolitains sans-fil (WMAN:Wireless Metropolitan Area Network) :

- Permettent la connexion à un opérateur (Internet, téléphonie, télévision) via ondes radio.
- Norme principale : IEEE 802.16.
- Exemple : WiMAX (Worldwide Interoperability for Microwave Access).

3.4. Réseaux étendus sans-fil (WWAN:Wirless Wide Area Network) :

- Comprend les réseaux téléphoniques et satellitaires.

Évolution de la téléphonie mobile :

- **2G (GSM)** : Débit de 9,6 Kbps.
- **2.5G (GPRS)** : Transfert de données par paquet, MMS, accès au web.
- **2.75G (EDGE)** : Débit plus élevé que GPRS.
- **3G (UMTS)** : Débit jusqu'à 2 Mbps, visioconférence, télévision, navigation web.
- **3G+ (HSDPA)** : Amélioration de la 3G.
- **4G (LTE)** : Débit de 100 Mb/s à 1 Gb/s, avec évolution LTE Advanced (>1 Gb/s).
- **5G** : Débits très élevés, connectivité rapide et fiable.

4. La norme IEEE 802.11(WIFI):

4.1. Architecture des réseaux IEEE 802.11 :

Le standard IEEE 802.11 propose deux modes de fonctionnement :

1. **Mode infrastructure :**
   - Les nœuds sans-fil se connectent à un point d'accès (AP), formant un **BSS** (Basic Service Set).
   - Plusieurs BSS peuvent être reliés via un **système de distribution (DS)** à travers leurs points d'accès.
   - Un ensemble de BSS interconnectés forme un **ESS** (Extended Service Set).
2. **Mode ad hoc :**
   - Les nœuds communiquent directement entre eux sans point d'accès, formant un **IBSS** (Independent Basic Service Set).

4.2. Modèle en couches de la norme IEEE 802.11 :

1. **Couches définies :**
   - **Couche physique** : Assure la transmission des données.
   - **Couche liaison de données** : Composée de deux sous-couches :
     - **LLC (Logical Link Control)** : Utilise les propriétés du standard 802.2.
     - **MAC (Medium Access Control)** : Spécifique à 802.11.
2. **Couche MAC (Medium Access Control) :**
   - Implémente deux méthodes d'accès au médium :
     - **DCF (Distributed Coordination Function)** : Basé sur CSMA/CA.
     - **PCF (Point Coordination Function)** : Accès coordonné par un point d'accès (AP).
   - **Mode infrastructure :** Utilise DCF et PCF.
   - **Mode ad hoc :** Utilise uniquement DCF.
3. **Méthodes d'accès au canal :**
   - **CSMA/CD (Ethernet)** : Détecte les collisions et retransmet après un délai aléatoire.
   - **CSMA/CA :** Évite les collisions grâce à l'utilisation de temporisations et de l'espacement intertrame (IFS) :

        - **SIFS (Short Inter-Frame Spacing)** : Pour séparer les transmissions dans un dialogue.
        - **DIFS (Distributed Inter Frame Space)** : Avant une nouvelle transmission.
4. **Mécanisme NAV (Network Allocation Vector) :**
    - Chaque nœud utilise un timer (NAV) pour savoir combien de temps le canal est occupé par un autre nœud, permettant d'éviter les collisions.

Principe de l'accusé de réception:

Le mécanisme d'accusé de réception (ACK) dans **CSMA/CA** fonctionne ainsi :

1. **Écoute du support** : Une station vérifie si le canal est libre.
2. **Canal occupé** : La transmission est différée.
3. **Canal libre (pendant DIFS)** : La station transmet après un délai aléatoire (backoff).
4. **Réception et vérification** : La station destination vérifie les données reçues et renvoie un accusé de réception (ACK) en cas de succès.
5. **Réception de l'ACK** : Indique à la station source que les données ont été reçues sans collision.
6. **Pas d'ACK** : La station retransmet les données en cas de non-réception de l'ACK.

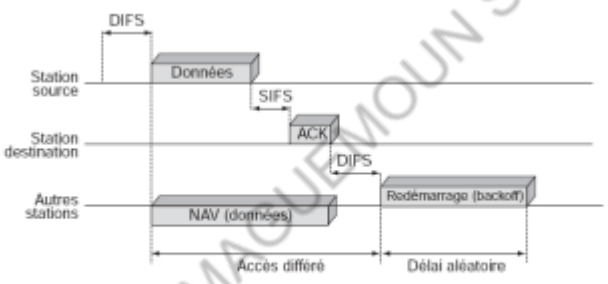


Fig. 1 : Principe de l'accusé de réception dans CSMA/CA (sans RTS/CTS) [G. Pujolle 2006]

Explication de la figure:

1. **Station source** :
    - Écoute le canal.
    - Si libre pendant un temps DIFS, elle transmet les données.
2. **Station destination** :
    - Reçoit les données, vérifie les erreurs.
    - Si tout est correct, envoie un accusé de réception (ACK) après un temps SIFS.
3. **Autres stations** :
    - Écoutent le canal et détectent une transmission en cours.
    - Utilisent un NAV (Network Allocation Vector) pour attendre jusqu'à la fin de la transmission.
4. **Accès différé (Backoff)** :
    - Si le canal est occupé, les stations utilisent un délai aléatoire (backoff) avant de réessayer la transmission.

Éléments clés :

- **DIFS** : Temps d'attente avant de transmettre.
- **SIFS** : Temps réservé aux réponses rapides comme l'ACK.
- **NAV** : Indique la durée pendant laquelle le canal est occupé.
- **Backoff** : Réduit les collisions en introduisant un délai aléatoire avant une nouvelle tentative.

Algorithme du backoff :

L'algorithme de **backoff** gère l'accès au canal pour éviter les collisions entre plusieurs stations voulant transmettre simultanément.

1. **Fonctionnement de l'algorithme :**
    - Si le canal est libre pendant un temps **DIFS**, chaque station génère un **backoff aléatoire** exprimé en nombre de time slots :
      Backoff_timer = Random[0, CW] × TimeSlot
      où **CW** est la fenêtre de contention.
    - Chaque station décrémente son backoff timer.
    - La station dont le backoff timer atteint 0 en premier émet ses données.
    - Les autres stations détectent l'occupation du canal, stoppent leur décrémentation, et entrent en période de **déferrement** (attente).
2. **Fenêtre de contention (CW) :**
    - Initialement, CW = **CWmin**.
    - En cas d'échec de transmission (collision), CW est doublée jusqu'à une limite maximale **CWmax** : **CW = (CWmin × 2^i) - 1** (où **i** est le nombre d'échecs consécutifs).
    - Après une transmission réussie, CW est réinitialisée à sa valeur minimale **CWmin**.

Ce mécanisme garantit un accès équitable au canal tout en minimisant les risques de collisions.

Mécanisme RTS/CTS et problème des nœuds cachés

Le mécanisme **RTS/CTS** dans la norme 802.11 utilise des paquets de contrôle pour éviter les collisions et résoudre le problème des **nœuds cachés** :

1. **Fonctionnement :**
    - Une station souhaitant émettre envoie un paquet **RTS (Request To Send)** avec les informations sur la source, la destination, et la durée de transmission.
    - La station destinataire répond par un **CTS (Clear To Send)** diffusé à son voisinage.
    - Les paquets **RTS** et **CTS** réservent le canal pour la durée de transmission.
2. **Réservations :**
    - Les stations qui reçoivent un **CTS** sans avoir envoyé de **RTS** comprennent que le canal est réservé et attendent leur tour.

    - La station ayant envoyé le **RTS** peut émettre une fois qu'elle reçoit le **CTS**.
3. **NAV (Network Allocation Vector) :**
    - Le NAV gère la réservation en indiquant aux stations la durée pendant laquelle le canal sera occupé, empêchant ainsi d'autres transmissions.

Ce mécanisme réduit les collisions et gère efficacement les conflits causés par les **nœuds cachés**.

4.2.2.Variantes de la couche physique dans la norme 802.11

La norme 802.11 propose plusieurs techniques pour la transmission au niveau de la couche physique :

1. **FHSS (Frequency Hopping Spread Spectrum)** :
    - La transmission saute rapidement entre différentes fréquences prédéfinies pour réduire les interférences et améliorer la sécurité.
2. **DSSS (Direct Sequence Spread Spectrum)** :
    - La donnée est étalée sur une large bande de fréquences à l'aide d'un code spécifique, rendant la transmission plus résistante aux interférences.
3. **IR (InfraRouge)** :
    - Utilise des signaux infrarouges pour la communication, adapté aux courtes distances.
4. **OFDM (Orthogonal Frequency Division Multiplexing)** :
    - Divise le signal en plusieurs sous-porteuses orthogonales pour transmettre les données en parallèle, améliorant l'efficacité et la vitesse.

Chaque technique est adaptée à des usages et environnements spécifiques pour optimiser la transmission sans fil.

5.Exemple d’un réseau cellulaire:le réseau GSM :

5.1.Architecture réseau GSM :

Le réseau GSM repose sur un système de **cellules** :

- Chaque cellule est couverte par une **station de base** (Base Transceiver Station - BTS), associée à des fréquences spécifiques.
- Un **contrôleur de station de base** (Base Station Controller - BSC) gère plusieurs BTS et leurs ressources radio.

Les composants principaux de l'architecture incluent :

1. **BSC (Base Station Controller)** : Gère les stations de base et les ressources radio associées.
2. **MSC (Mobile Switching Center)** : Centre de commutation pour la gestion des appels et des données mobiles.
3. **HLR (Home Location Register)** : Base de données contenant les informations des abonnés enregistrés dans un domaine.

4. **VLR (Visitor Location Register)** : Base de données temporaire pour les abonnés se trouvant dans une zone spécifique.
5. **AuC (Authentication Center)** : Centre d'authentification qui sécurise les communications.

Cette structure permet une gestion efficace des communications mobiles en répartissant les ressources et en suivant les abonnés dans le réseau.

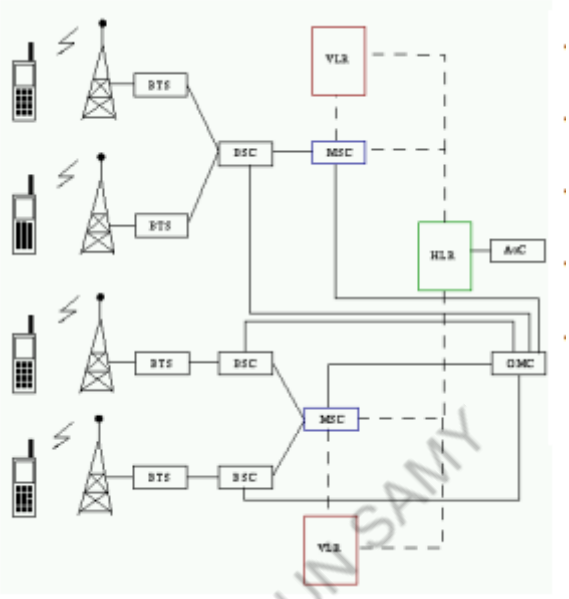


Fig. 2 : Architecture du réseau GSM

Explication de la figure:

La figure montre l'**architecture d'un réseau GSM**, composée de plusieurs éléments clés pour gérer les communications mobiles :

1. **Stations de base (BTS)** : Elles couvrent les cellules et assurent la communication entre les mobiles et le réseau via des fréquences radio.
2. **Contrôleur de stations de base (BSC)** :
   - Contrôle plusieurs BTS.
   - Gère les ressources radio et coordonne les handovers.
   - Relie les BTS au reste du réseau.
3. **Centre de commutation mobile (MSC)** :
   - Gère les appels, messages et transferts de données.
   - Assure la communication avec les réseaux externes (fixes ou Internet).
4. **Bases de données** :
   - **HLR (Home Location Register)** : Stocke les informations des abonnés.
   - **VLR (Visitor Location Register)** : Stocke temporairement les données des mobiles présents dans une zone.
   - **AuC (Authentication Center)** : Sécurise les communications en authentifiant les utilisateurs.

Fonctionnement :

- Les mobiles communiquent avec la BTS.
- La BTS transmet au BSC, qui envoie au MSC pour gérer les appels et les données.
- Les bases de données (HLR, VLR) fournissent les informations nécessaires.
- Le AuC garantit l'authentification et la sécurité.

Résumé : Cette architecture hiérarchique organise efficacement la gestion des communications mobiles, de la cellule locale jusqu'aux réseaux externes, tout en assurant la sécurité et la fluidité des échanges.

L'enregistreur de localisation nominale : HLR (Home Location Register) :

- Base de données principale mise à jour par le MSC.
- Contient des informations sur les abonnés (type d'abonnement, numéro) et des données dynamiques (position de l'abonné via le VLR, état du terminal : allumé, éteint, en communication).

L'enregistreur de localisation des visiteurs: VLR (Visitor Location Register) :

- Base de données temporaire liée à un MSC.
- Stocke uniquement des informations dynamiques transmises par le HLR lorsqu'un abonné entre dans la zone de couverture du MSC.
- Les données de l'abonné sont transférées vers un autre VLR lorsqu'il quitte la zone, permettant aux informations de suivre ses déplacements.

5.2.Gestion de la mobilité(MM:Mobility Management) :

La gestion de la mobilité permet de suivre la localisation d'un utilisateur dans un réseau via deux fonctions principales :

1. **Enregistrement** :
    - **Cas 1 : Déplacement dans le domaine d'abonnement**
        - Lorsqu'un utilisateur arrive dans un nouveau VLR, ce dernier contacte le HLR de l'utilisateur.
        - Le HLR met à jour la nouvelle localisation et annule l'attachement au précédent VLR.
    - **Cas 2 : Roaming (départ du domaine d'abonnement)**
        - Le HLR de la zone visitée contacte le HLR d'origine pour annuler l'attachement au VLR précédent et mettre à jour la localisation.
2. **Paging** :
    - Processus où le terminal mobile reste à l'écoute des informations de sa station de base.
    - Lors d'un appel ou de la réception de données, l'adresse de l'utilisateur est diffusée dans toutes les cellules concernées.
    - Une fois l'adresse reçue, le terminal envoie un accusé de réception à la station de base et la communication commence.

5.3.Gestion des ressources radio (RRM:radio ressource management) :

Méthodes d'accès radio dans le GSM :

1. **FDMA (Frequency Division Multiple Access) :**
    - Repose sur un multiplexage fréquentiel qui divise la bande de fréquences en plusieurs sous-bandes.
    - Chaque porteuse transporte le signal d'un seul utilisateur.
2. **Utilisation dans le GSM :**
    - **GSM 900 MHz :**
        - Deux bandes de 25 MHz :
            - Voie montante : 890 - 915 MHz.
            - Voie descendante : 935 - 960 MHz.
        - Écart duplex : 45 MHz.
        - Largeur de chaque canal : 200 kHz.
    - **GSM 1800 MHz (DCS-1800) :**
        - Bande de fréquences élargie :
            - Voie montante : 1710 - 1785 MHz.
            - Voie descendante : 1805 - 1880 MHz.
        - Écart duplex : 95 MHz.
3. **TDMA (Time Division Multiple Access) :**
    - Repose sur un multiplexage temporel en découpant le temps disponible entre plusieurs utilisateurs.
    - Chaque utilisateur occupe un **time slot (Tslot)** pendant une fraction de temps.
    - Dans le GSM :
        - Une porteuse de 200 kHz est divisée en **8 slots** (numérotés de 0 à 7).
        - Durée d'un slot : **577 µs (0,577 ms)**.
        - Une trame TDMA : **8 slots regroupés**, soit une durée de **4,615 ms**.