

Cours Réseaux

Automne-2019

Pierre Leone

Pierre.leone@unige.ch

Adresse e-mail, évaluation

Enseignant: Pierre leone, pierre.leone@unige.ch

Evaluation:

Examen à la fin du semestre/ou contrôle continu

Exercices:

Pas notés mais sont inclus dans l'examen

Références

- Andrew Tanenbaum, *Réseaux Cours et exercices*, 3^{ème} édition, Prentice Hall (bibliothèque C.2.1. TAN)
- Guy Pujolle, *Les Réseaux*, édition 2003, Eyrolles, (bibliothèque C.2.1. PUJ)
- Craig hunt, *TCP/IP Administration de réseaux*, 3^{ème} édition, O'reilly, (bibliothèque C.2.5 HUN)

Généralités

Utilité des réseaux informatiques :

- **Partage des ressources**
 - imprimantes
 - puissance de calculs
- **Augmenter la fiabilité des systèmes**
 - duplication des données
 - duplication des ressources de calculs (banques, réservation,...)
- **Réduction des coûts**
 - ordinateurs personnels vs superordinateur (central)
- **Augmentation de performances (scalability)**
 - ajout de ressources (processeurs) si nécessaire
- **Disposer d'un média de communication (accès à l'information)**
 - messagerie
 - vidéoconférence
 - librairie en ligne

Définition générale

Un réseau est constitué d'un assemblage de composants informatiques, matériels et logiciels ainsi que d'un ou des supports de communications (câbles, faisceaux hertziens,...) permettant à plusieurs ordinateurs de communiquer entre eux.

Modèle ISO

Pour communiquer les équipements informatiques (les hôtes) doivent se mettre d'accord sur un ensemble de règles: **Les protocoles.**

Le modèle ISO (interconnexion des systèmes ouverts) propose un modèle en **couches.**

On commence par s'intéresser à des protocoles qui utilisent les couches physique et liaison de données.

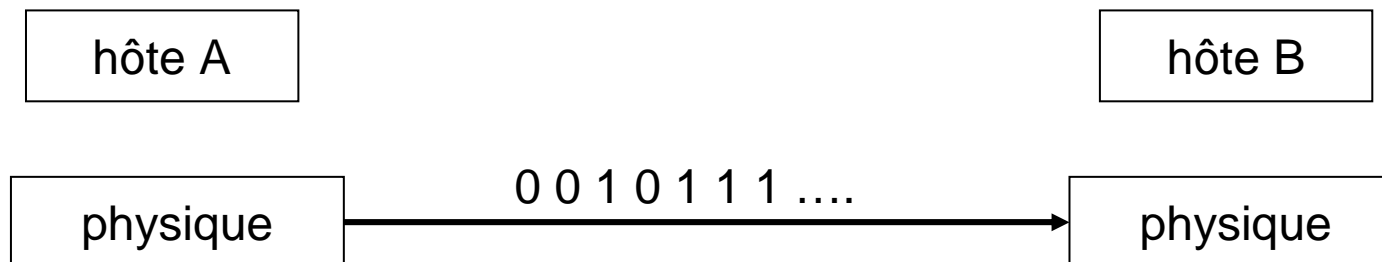
Les différentes couches (ISO)

- **La couche physique:**

est responsable de transmettre 'physiquement' l'information d'une station à une autre. La nature des signaux et du média de communication doivent être définis.

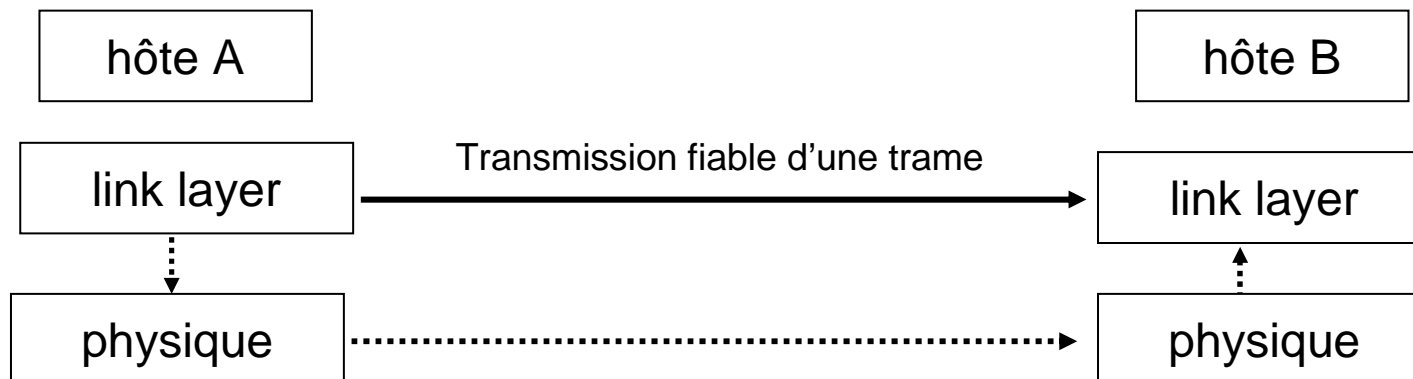
- Signaux électriques, tensions appliquées
- Signaux optiques, longueur d'ondes

Cette couche est responsable de la transmission d'une suite de 0 et de 1, sans format, sous la forme d'un *flot de données*.



Les différentes couches (ISO)

- **La couche liaison de données – link layer (LL):**
fiabilise les transmissions de la couche physique.
 - gère les erreurs de transmission
 - Structure le flot de données en **trame de données** (de 100 à 1000 octets).
 - Introduit différents types de trames – contrôles et données – en particulier les **trames d'acquittement** qui permettent de vérifier que les données ont bien été reçues.
 - Corrige les erreurs – trames perdues ou transmission.



Protocoles ARQ

Modèles markoviens (discrets homogènes)

Protocoles ARQ

Les protocoles ARQ (Automatic-Repeat_Request) sont destinés à corriger les erreurs de transmissions en répétant les messages non acquités. Ces protocoles sont implémentés dans les couches liaisons de données (et transport - TCP).

Comme ce sont des protocoles de la couche liaison de données, ils transmettent des trames. Elles ont un début et une fin, les protocoles peuvent identifier les trames de contrôles et les trames de données et aussi vérifier l'intégrité des données (détection d'erreurs).

Les trames de contrôles sont transmises à l'application qui utilise le réseau, les trames de contrôle sont utilisées par les protocoles.

Trames

La structure des trames est de la forme



L'en-tête contient les informations qui définissent le type de trames (données/contrôle) et des données de contrôle (par exemple le numéro de la trame).

Parmi les trames de contrôle il y a les trames d'acquittement qui sont utilisées par un récepteur pour indiquer à l'émetteur que les données ont été correctement reçues (acquittement positif) ou reçues avec erreur (acquittement négatif).

Protocoles ARQ

On va considérer trois protocoles ARQ différents

1. Stop-and-Wait ARQ
2. Go-Back-N ARQ
3. Selective-Repeat ARQ

Pour comparer ces différents protocoles on va développer des modèles simples et évaluer leurs performances.

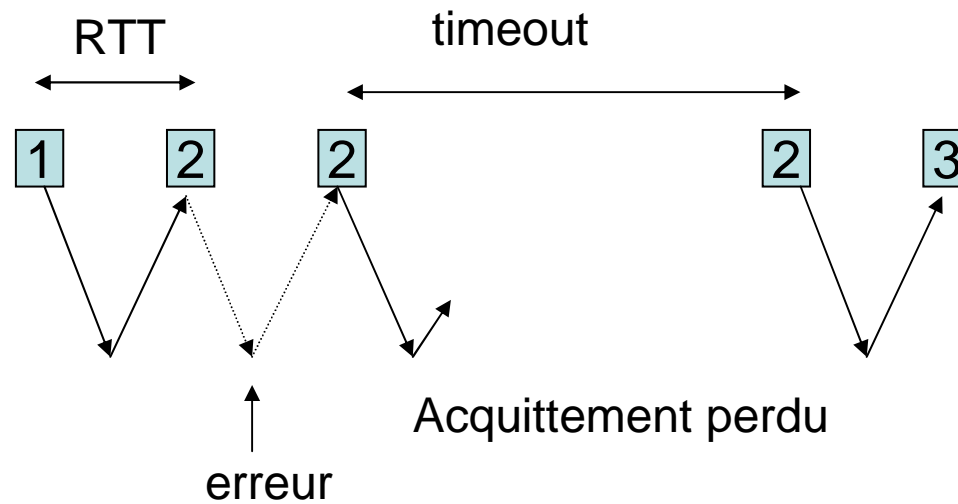
Stop-and-Wait

1. Stop-and-Wait ARQ

L'émetteur procède selon les étapes suivantes:

1. transmet une trame
2. déclenche un temporisateur et attend un acquittement
3. si un acquittement positif est reçu, transmet la trame suivante
4. si un acquittement négatif ou que le temporisateur expire, retransmet la trame.
5. goto 1.

Stop-and-Wait



Go-Back-N

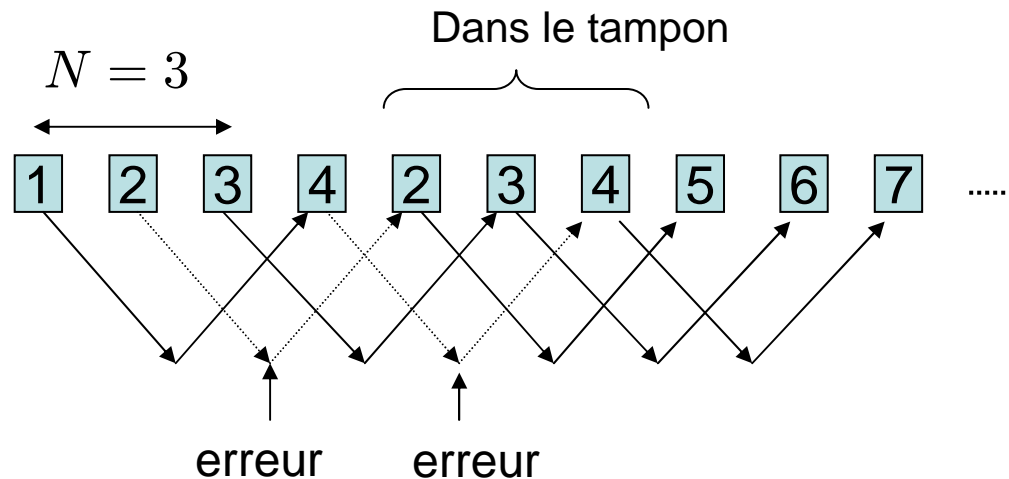
2. Go-Back-N

Avec le protocole Go-Back-N, l'émetteur transmet les trames et mémorise les trames localement jusqu'à réception de l'acquittement. Le nombre de trames dans le tampons d'émission s'appelle **la fenêtre de transmission**, noté N. N est le nombre de trames transmises pendant un intervalle RTT.

Le récepteur vérifie que les trames reçues sont correctes. Si oui, il transmet un acquittement positif (ACK) pour chaque trame séparément. Lorsque l'émetteur reçoit l'acquittement il supprime la trame correspondante de son tampon d'émission.

Go-Back-N

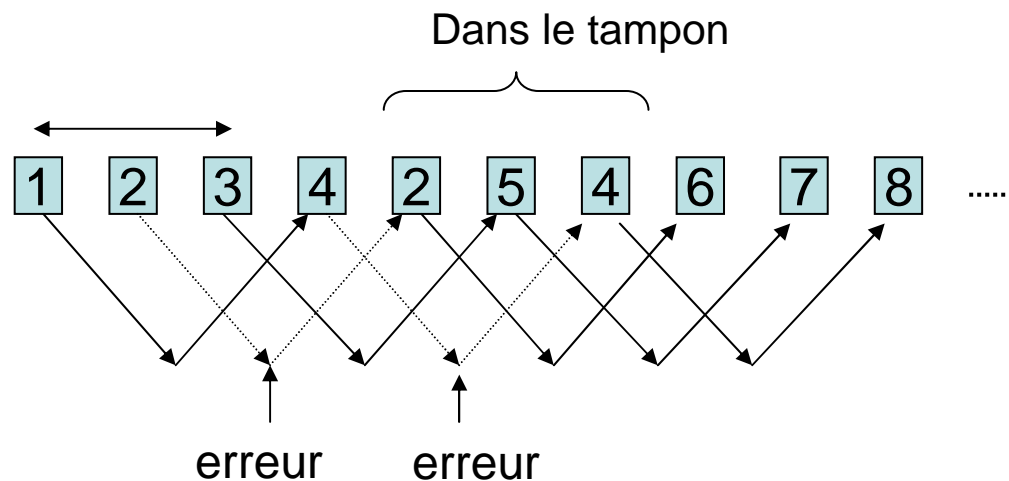
Si une erreur est détectée par le récepteur, il transmet un acquittement négatif (NAK) pour cette trame particulière. Lorsque le récepteur reçoit l'acquiescement négatif, il retransmet toutes les trames contenues dans son tampon d'émission (en commençant par la trame correspondante).



Selective Repeat ARQ

3. Selective Repeat ARQ (SR ARQ)

Le protocole SR ARQ améliore le protocole Go-Back-N. Avec ce protocole, l'émetteur ne retransmet que les trames négativement acquitées.



Remarques

Les protocoles Go-Back-N et Selective Repeat essaye d'utiliser au maximum l'intervalle de temps RTT pour transmettre des trames.

Le protocole Selective Repeat est implémenté dans la couche TCP du protocole TCP/IP.

La différence entre les protocoles Go-Back-N et Selective Repeat est que avec SR ARQ le récepteur doit être capable de mémoriser les trames reçues correctement pour les transmettre dans l'ordre au protocole de la couche réseau.

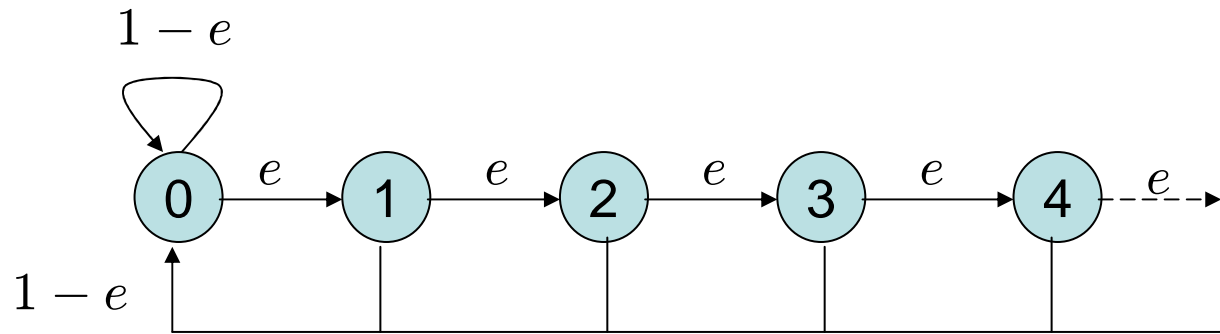
Stop-and-Wait - modèle

On cherche à estimer les performances de ce protocole. Supposons,
n est le nombre moyen de bits par trame transmise
 ϵ est la probabilité qu'un bit soit modifié lors de la transmission
on suppose que les acquittements sont toujours reçus
l'émetteur retransmet une trame jusqu'à ce qu'il reçoive un
acquittement positif

On modélise l'émetteur par un diagramme d'état, les différents états
possible sont 0, 1, 2, 3, ...

L'émetteur est dans l'état i si la trame actuellement transmise a déjà
été transmise i fois sans succès.

Stop-and-Wait - modèle



e , la probabilité qu'une trame soit transmise avec erreur est donné par

$$e = 1 - (1 - \epsilon)^n \approx \epsilon n$$

Stop-and-Wait - modèle

On suppose qu'au temps $t=0$, l'émetteur doit transmettre la première trame. Le temps t est incrémenté chaque fois qu'une transmission a lieu, que ce soit une nouvelle trame ou une retransmission.

On remarque que l'état dans lequel l'émetteur se trouve au temps i est aléatoire. On décrit l'état de l'émetteur au temps t par un vecteur de probabilité

$$s(t) = (s_0(t) \quad s_1(t) \quad s_2(t) \quad \dots)$$

avec $s_i(t)$ est la probabilité que le vecteur se trouve dans l'état i au temps t .

Remarquez que $\sum_{i \geq 0} s_i(t) = 1$

Stop-and-Wait - modèle

Au temps $t = 0$, l'état de l'émetteur est déterminé

$$s(0) = (1 \quad 0 \quad 0 \quad \dots)$$

La première transmission échoue avec probabilité e et réussit avec probabilité $1-e$. On a donc

$$s(1) = (1 - e \quad e \quad 0 \quad \dots)$$

En effet, si on note $x(t)$ l'état dans lequel se trouve l'émetteur au temps t , on a

$$\text{Prob}(x(1) = i \mid x(0) = 0) = \begin{cases} 1 - e & \text{si } i = 0 \\ e & \text{si } i = 1 \end{cases}$$

Stop-and-Wait - modèle

Pour calculer le vecteur de probabilités au temps 2, on utilise la formule des probabilités totales. Rappel,

$$Prob(E) = \sum_k Prob(E \mid P_k) Prob(P_k)$$

Ou, E est un événement et avec $\bigcup_k P_k = \Omega$, $P_i \cap P_j = \emptyset$
 Ω est l'espace fondamental qui représente tous les événements possibles.

Dans notre cas, on veut calculer

$$Prob(x(2) = i)$$

Stop-and-Wait - modèle

L'ensemble des événements possibles est l'ensemble des suites d'états que l'émetteur peut atteindre. C'est-à-dire, l'ensemble des suites de la forme

$$x(0)x(1)x(2)x(3)x(4) \dots$$

On ne peut pas écrire n'importe quelle suite

Par contre, on peut classer ces suites en fonction l'état dans lequel se trouve l'émetteur au temps $t=1$. On définit donc

$$P_1 = \{x(1) = 0\}, \text{ et } P_2 = \{X(1) = 1\}$$

Stop-and-Wait - modèle

On applique la formule des probabilités totales pour obtenir

$$\begin{aligned} Prob(x(2) = i) = & Prob(x(2) = i \mid x(1) = 0) Prob(x(1) = 0) + \\ & Prob(x(2) = i \mid x(1) = 1) Prob(x(1) = 1) \end{aligned}$$

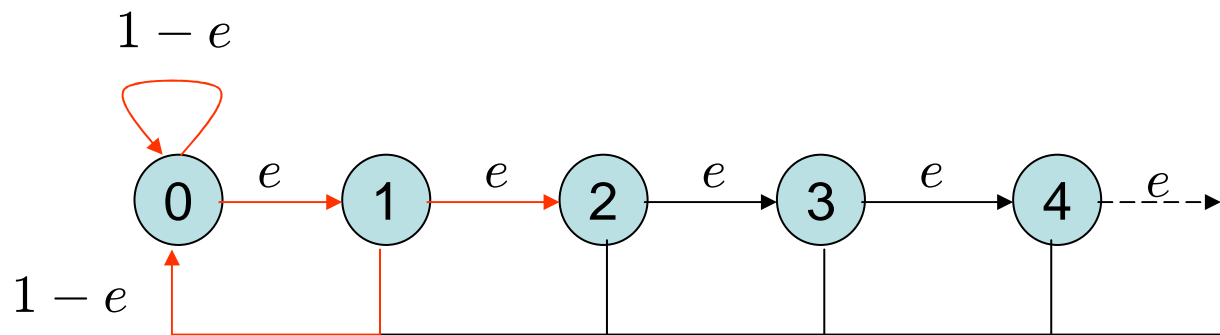
On a déjà calculé $Prob(x(1) = 0)$ et $prob(x(1) = 1)$ et on sait que

$$Prob(x(2) = i \mid x(1) = 0) = \begin{cases} e & \text{si } i = 1 \\ 1 - e & \text{si } i = 0 \\ 0 & \text{sinon} \end{cases}$$

Stop-and-Wait - modèle

De même

$$Prob(x(2) = i \mid x(1) = 1) = \begin{cases} e & \text{si } i = 2 \\ 1 - e & \text{si } i = 0 \\ 0 & \text{sinon} \end{cases}$$



Stop-and-Wait - modèle

On peut donc calculer le vecteur de probabilités au temps $t = 2$ et réappliquer la formule des probabilités totales pour calculer le vecteur aux temps $t = 3, 4, 5, \dots$

On peut simplifier les calculs en utilisant la notation matricielle. En fait, nous modélisons l'état de l'émetteur par une chaîne de Markov.

Une première remarque importante est que

$$Prob(x(t) = j \mid x(t-1) = i)$$

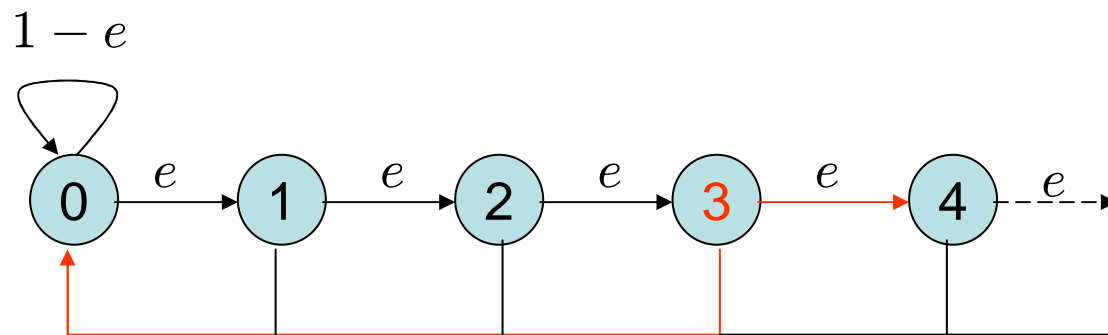
Ne dépend pas du temps t , on écrit donc

$$Prob(x(t) = j \mid x(t-1) = i) = p_{ij}$$

Stop-and-Wait - modèle

Les p_{ij} s'appellent les probabilités de transitions. Généralement, on a

$$p_{ij} = \begin{cases} e & \text{si } j = i + 1 \\ 1 - e & \text{si } j = 0 \\ 0 & \text{sinon} \end{cases}$$



Stop-and-Wait - modèle

On peut réécrire la formule des probabilités totales

$$Prob(x(t) = j) = \sum_{i=0}^{\infty} p_{ij} Prob(x(t-1) = i)$$

$s_j(t)$ ←

$$\sum_{i=0}^{\infty} s_i(t-1) p_{ij} \leftarrow \text{Produit vecteur-matrice}$$

$$s(t) = (s_0(t-1) \quad s_1(t-1) \quad s_2(t-1) \quad \dots) \underbrace{\begin{pmatrix} p_{00} & p_{01} & p_{02} & \dots \\ p_{10} & p_{11} & p_{12} & \dots \\ p_{20} & p_{21} & p_{22} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}}_{:=P}$$

Stop-and-Wait - modèle

Le vecteur de probabilité peut donc se calculer en effectuant les produits vecteur-matrice

$$s(t) = s(t-1)P = s(t-2)P^2 = \dots = s(0)P^t$$

On s'intéresse au comportement du protocole lorsque le temps devient grand. On veut donc calculer

$$\lim_{t \rightarrow \infty} P^t$$

Stop-and-Wait - modèle

La matrice P est une matrice stochastique, i.e. elle satisfait (pourquoi?)

$$\forall i, \quad \sum_{j \geq 0} p_{ij} = 1$$

Dans notre cas, elle s'écrit

$$P = \begin{pmatrix} 1-e & e & 0 & 0 & \dots \\ 1-e & 0 & e & 0 & \dots \\ 1-e & 0 & 0 & e & \dots \\ \vdots & \vdots & \vdots & 0 & \ddots \end{pmatrix}$$

Stop-and-Wait - modèle

Un résultat très général nous indique que pour *matrice stochastique* qui définit un processus de Markov tel que de *n'importe quel état on puisse accéder à n'importe quel autre état (et l'espérance du temps de retour est fini)*, la limite existe et s'écrit

$$\lim_{t \rightarrow \infty} P^t = \underbrace{\begin{pmatrix} p_1 & p_2 & p_3 & \dots \\ p_1 & p_2 & p_3 & \dots \\ p_1 & p_2 & p_3 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}}_{P^\infty}$$

Stop-and-Wait - modèle

Lorsque le temps devient grand le comportement de l'émetteur tend vers un comportement stationnaire (ne change pas en probabilité avec le temps). En effet, soit $s(0)$ n'importe quel vecteur de probabilité,

$$\lim_{t \rightarrow \infty} s(t) = \lim_{t \rightarrow \infty} s(0)P^t = s(0) \begin{pmatrix} p_1 & p_2 & p_3 & \dots \\ p_1 & p_2 & p_3 & \dots \\ p_1 & p_2 & p_3 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} = \underbrace{(p_1 \quad p_2 \quad p_3 \quad \dots)}_{p^\infty}$$

Stop-and-Wait - modèle

Le vecteur p^∞ est l'unique vecteur qui satisfait $p^\infty P = p^\infty$

En effet, $p^\infty = \lim_{t \rightarrow \infty} sP^{(t+1)} = \lim sP^t P = p^\infty P$

L'unicité est plus difficile à démontrer, mais on accepte le fait que l'équation $p^\infty P = p^\infty$ caractérise le vecteur invariant.

Dans notre cas, on obtient

$$P^\infty = (1 - e) \begin{pmatrix} 1 & e & e^2 & e^3 & \dots \end{pmatrix}$$

Stop-and-Wait - modèle

On peut maintenant évaluer les performances du protocole. Le **nombre moyen de retransmissions**. On définit

A_i est l'événement 'l'émetteur est dans l'état i et transmet avec succès', $Prob(A_i) = p_i^\infty(1 - e)$

$$N_t = \sum_{i \geq 0} i Prob(A_i) = (1 - e)^2 \sum_{i \geq 0} i e^i = e$$

L'efficacité du canal est l'inverse du nombre total de transmission par trame

$$\eta = \frac{1}{1 + N_t} = \frac{1}{1 + e}$$

Stop-and-Wait - modèle

La probabilité invariante p_i^∞ s'interprète comme le nombre de d'intervalles de temps que l'émetteur passe dans l'état i . Si la transmission d'une trame dure x secondes, le temps moyen est $x p_i^\infty$

Pour calculer le débit du protocole on calcule le temps moyen nécessaire à l'émetteur pour transmettre une trame avec succès.

$$\text{Throughput} = \sum_{i \geq 0} p_i^\infty (1 - e) = 1 - e \text{ trames/intervalle de temps}$$

ARQ remarques

Pour les protocoles ARQ on considère deux hôtes communiquant directement. Cette communication peut être possible soit via un câble partagé uniquement par les deux hôtes ou alors parce que les hôtes ont établi **une connexion**. Une connexion est gérée par le protocole de liaison de données. Un hôte initie la connexion qui est acceptée (ou refusée) par l'autre hôte par l'envoi de trames de contrôles.

Le temps de boucle RTT (Round-Trip-Time) est le temps qui s'écoule entre le moment où un paquet est émis jusqu'à la réception de l'acquittement.

Les protocoles ARQ sont généralement utiles dans la situation où le temps de boucle est faible par rapport au temps nécessaire à la transmission du message. Dans le cas contraire, le protocole passe trop de temps à attendre les acquittements.

Réseaux locaux – MAC

- Protocole à accès aléatoires (Aloha)

- Protocoles à accès avec détection de porteuse CSMA/CD (Ethernet)

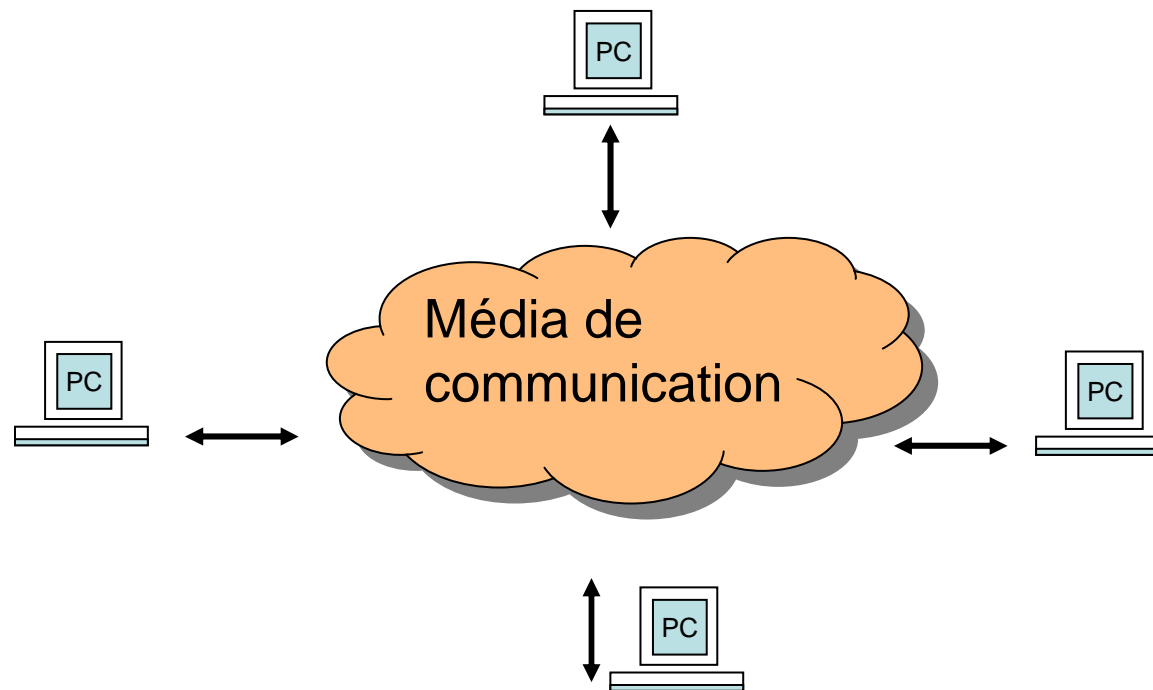
- Protocoles à partage de ressources (Token)

- Protocoles à accès avec détection de porteuse CSMA/CA (802.11)

Couche liaison et réseaux locaux

L'implémentation des couches physique et liaisons de données permet le développement de réseaux d'interconnexions d'hôtes.

Schématiquement



Réseaux locaux (LAN)

- généralement des réseaux privés
- 'faibles' distances de communications, délai de transmission borné (connu $\sim 10\mu\text{s}$)
- couvre des universités, centres de recherche (Initialement), des banques, entreprises localisées.

Réseaux locaux

Le média de communication peut-être

- Un câble partagé par les stations hôtes et transportant des signaux électriques
- Des ondes électromagnétiques, on parle de radio-fréquence (RF) pour les ondes radios comprises entre 3Khz et 300GHz utilisés pour les radiocommunications.

Dans les deux cas on parle de **liaison à diffusion** les stations hôtes se partagent le même canal de communication.

Le média de communication peut aussi être

- Un ensemble de câbles reliant chaque station hôte à toutes les autres. Dans ce cas, on a un ensemble de **liaisons point-à-point**.

Réseaux point-à-point

Pour les réseaux point-à-point (peer-to-peer) le protocole permet à une station hôte de communiquer avec une (seule) autre station hôte.

Ce type de protocoles sont généralement utilisés pour connecter deux stations hôtes via une ligne louée, ou un ordinateur à sont FAI (Fournisseur d'Accès Internet).

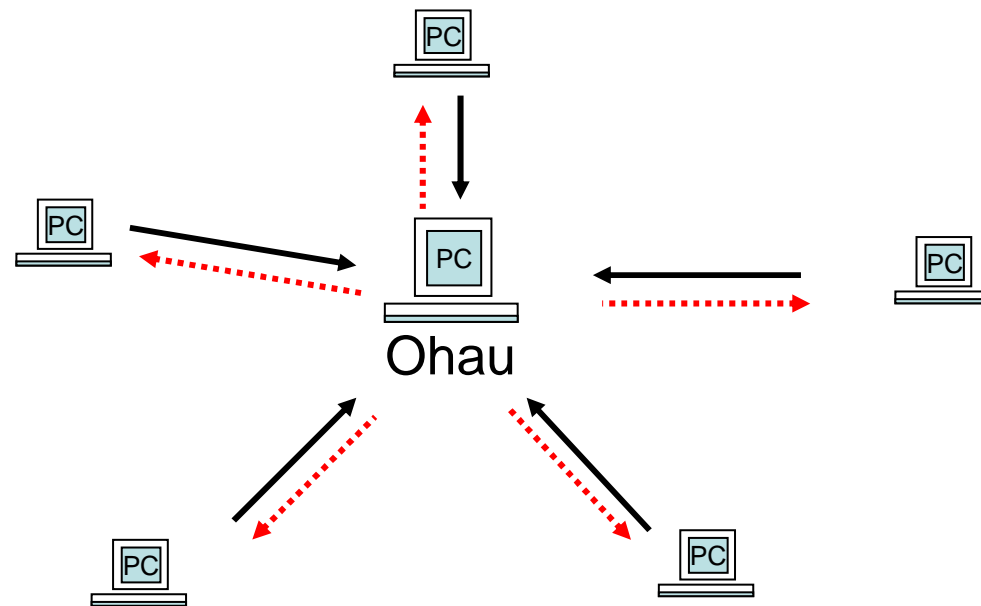
Par exemple les protocoles PPP (procole point-à-point), HDLC (High-level Data Link Control)

Aloha

Réseaux à diffusion - ALOHA

Une famille de protocoles à diffusion très répandue est la famille des **protocoles ALOHA**.

Le premier protocole ALOHA a été développé pour assurer des communications par RF entre l'île d'Oahu et d'autres îles environnante. La structure du réseau était un hub (en étoile).



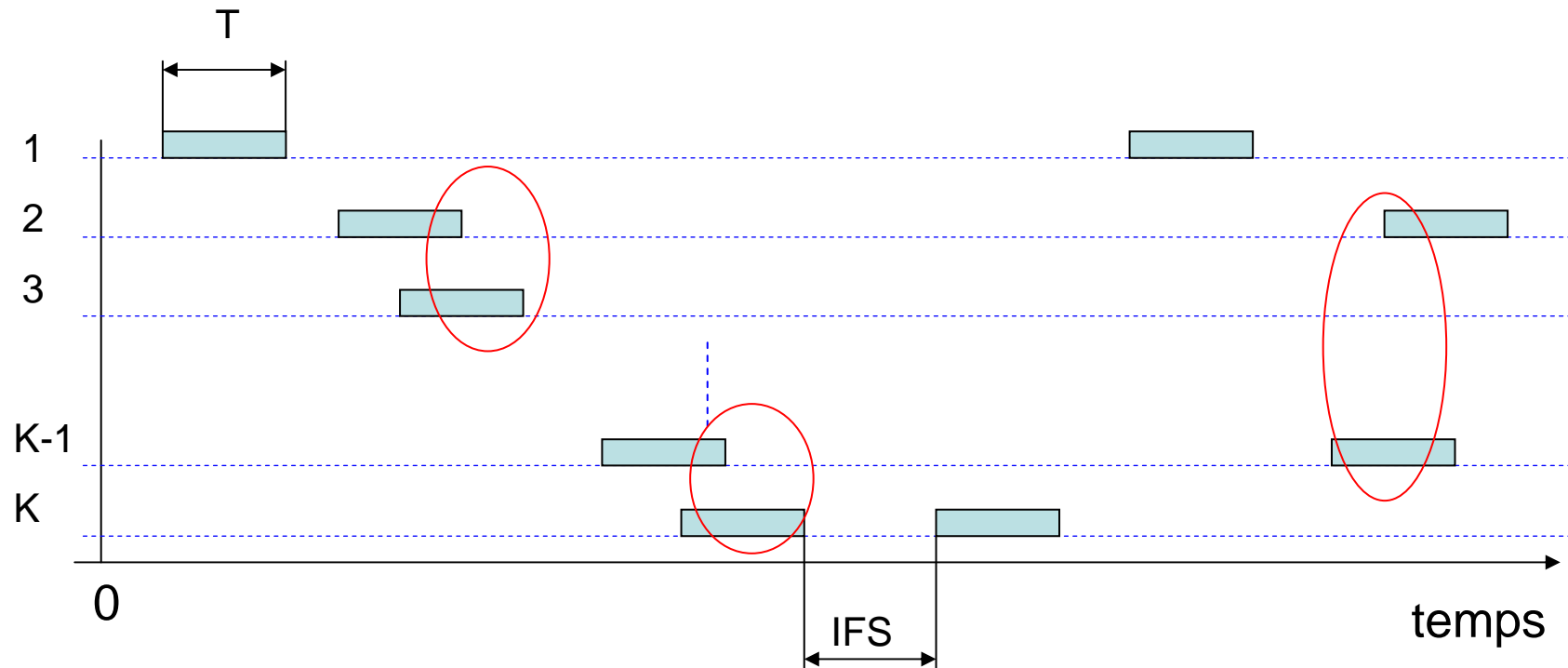
ALOHA

Les communications utilisent 2 fréquences distinctes pour les communications Ohau -> tout le monde et inversement.

Lorsque Ohau transmet, toutes les autres stations reçoivent les données. On suppose sans erreurs.

Lorsque plusieurs stations transmettent vers Ohau alors Ohau reçoit les données correctement seulement si la station est la seule à transmettre. Si plusieurs stations transmettent on parle de **collisions**.

ALOHA - collisions



IFS: Inter Frame Space, temps inter-frames

ALOHA - protocole

Lorsqu'une trame est soumise à la couche liaison de données d'une station,

1. La station attend un temps aléatoire avec une distribution exponentielle
2. La station transmet la trame
3. La station attend un temps fixe. Si un acquittement positif est reçu d'Ohau pendant l'attente la transmission est finie, sinon retour au point 1 (retransmission).

Le protocole exécuté par Ohau est

1. Attendre une trame
2. Si la trame est reçue correctement transmettre un acquittement positif.
3. Retour en 1.

Loi exponentielle

Une variable aléatoire X est distribuée selon une loi exponentielle de paramètre λ (λ -exponentielle) si la fonction de densité de probabilité est de la forme

$$f_x(t) = P(X = x) = \lambda e^{-\lambda t}, \quad t \geq 0$$

Et la fonction de répartition

$$F_X(t) = P(X \leq t) = 1 - e^{-\lambda t}, \quad t \geq 0$$

Exercices

On divise l'intervalle $[0,t]$ en sous-intervalles de longueur $1/n$. Pour chaque intervalle, on définit une variable de Bernoulli de probabilité λ/n , c'est-à-dire que la variable est 1 avec probabilité λ/n et 0 sinon.

a. Montrez que la probabilité que toutes les variables de Bernoulli soient nulles dans l'intervalle $[0,t]$ tend vers $e^{-\lambda t}$.

b. En déduire que vous pouvez approximer une variable aléatoire λ -exponentielle en divisant le temps en intervalles de longueur $1/n$ et en simulant des variables de Bernoulli.

Exercices

On appelle un processus de Poisson de paramètre λ une suite de temps aléatoires $\{T_n\}_{n=0}^{\infty}$ telle que

$$T_{k+1} - T_k$$

est distribué selon une loi λ -exponentielle. Généralement on utilise la convention que $T_0 \leq 0 < T_1$.

a. Montrez que T_1 suit une distribution λ -exponentielle.

b. Montrez que si deux processus de Poisson de paramètres λ et β sont superposés, alors on obtient un processus de Poisson de paramètre $(\lambda + \beta)$.

ALOHA - modèle

On veut évaluer les performances du protocole ALOHA, en particulier on veut évaluer le débit (throughput = #trames/secondes) du réseaux.

Hypothèses:

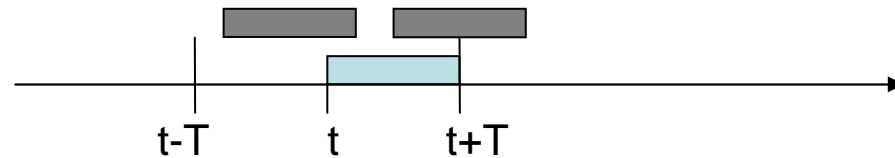
1. Les trames transmises par les stations ont toutes la même longueur **T**.
2. Les stations sont **saturées**, i.e. la couche liaison de données à toujours une trame à transmettre.
3. Les transmissions sont **indépendantes** les unes des autres.
4. Les **distributions exponentielles** utilisées par les stations pour transmettre ont toutes le même paramètre **λ** .

En particulier, pour une station

$$P(IFS > T) = \int_T^{\infty} \lambda e^{-\lambda x} dx = e^{-\lambda T}$$

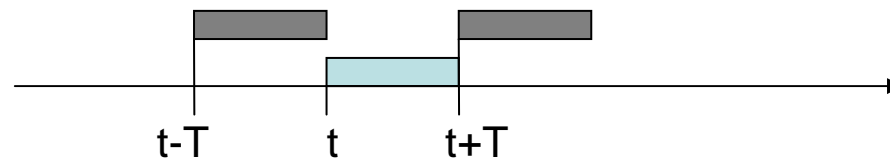
ALOHA

Pour qu'une trame transmise au temps t avec succès, il faut que les autres stations ne transmettent pas dans l'intervalle $[t-T, t+T]$ (la fenêtre de vulnérabilité).



Slotted ALOHA

On améliore le débit du réseaux si on a la possibilité de **synchroniser** les stations, c'est-à-dire s'assurer quelles ont accès à une horloge commune. Dans ce cas, on autorise les transmissions seulement aux temps $0, T, 2T, 3T, \dots$

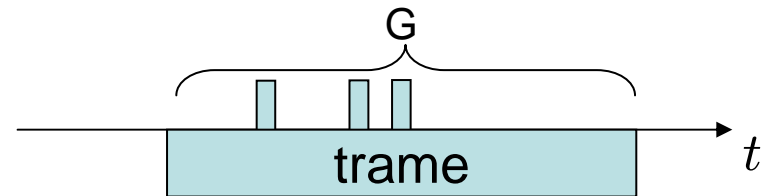


Dans cas, seules les trames générées pendant l'intervalle $[t-T, t]$ vont entrer en collisions. En synchronisant les transmissions la fenêtre de vulnérabilité est divisée par deux.

ALOHA - analyse

On trouve l'analyse suivante dans la littérature.

On note **G** le nombre moyen de trames transmises par unité de temps de trame (nouvelles trames et retransmissions) et **S** le **débit effectif** (les trames transmises avec succès)



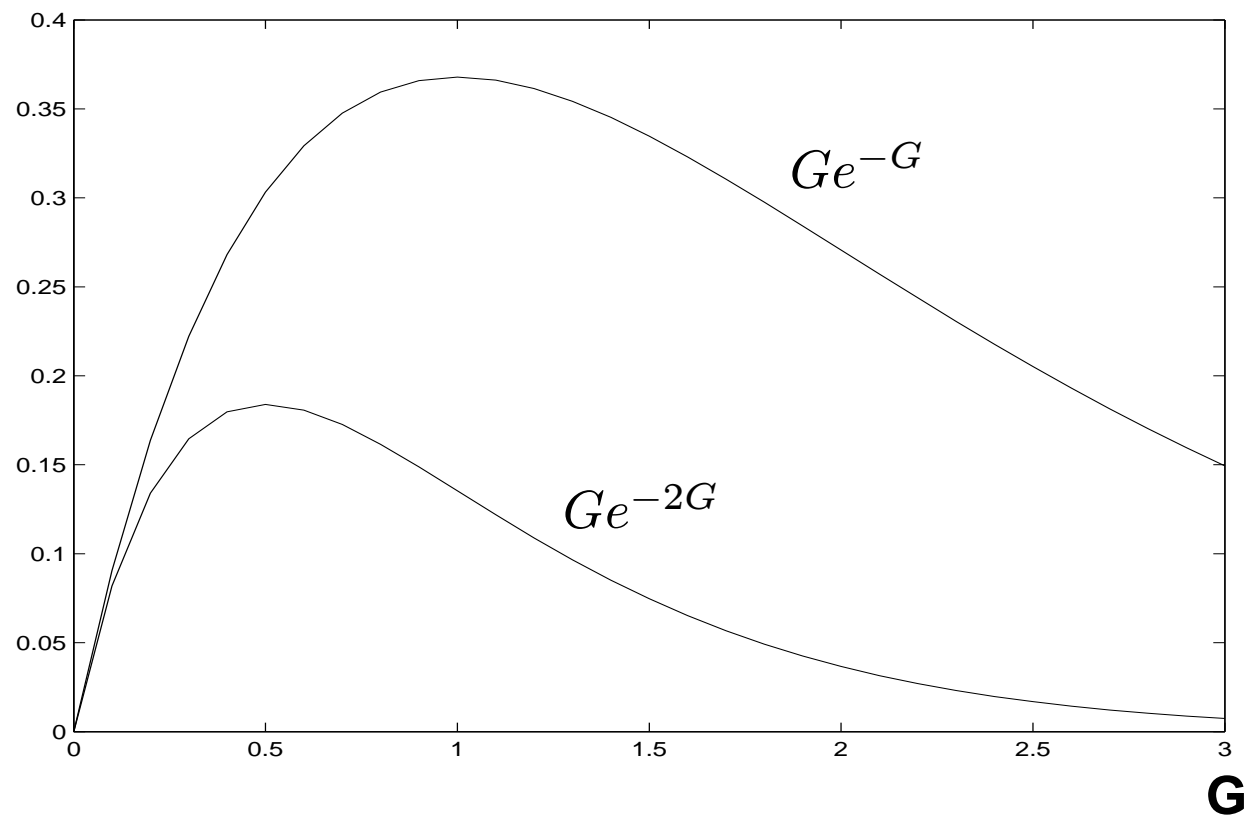
L'hypothèse est que les transmissions se font selon un processus de poisson d'intensité **G**.

Pour Aloha, pour qu'une trame soit transmise correctement il faut qu'il n'y ait pas de transmission initiée pendant la période de vulnérabilité.

$$\text{Pour Aloha pur: } S = Ge^{-2G}$$

$$\text{Pour slotted Aloha: } S = Ge^{-G}$$

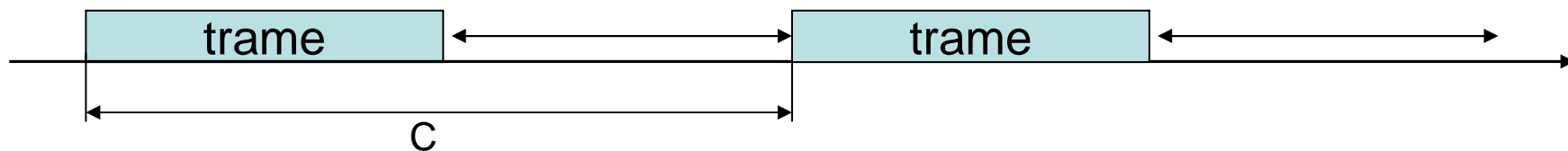
Comparaison



Exercices

On considère k stations qui exécutent le protocole Aloha, qui sont saturées, le temps de transmission d'une trame est T et les temps d'attentes sont λ -exponentiels.

Chaque station exécute répétitivement un cycle C qui comprend l'émission d'une trame et une attente aléatoire.



- Calculez la valeur moyenne d'un cycle. $[T+1/\lambda]$
- Calculez le nombre moyen de trames émise par une station par unité de temps (le débit). $[\lambda/(\lambda T+1)]$
- Quelle est la probabilité qu'une station transmette/ne transmette pas au temps t (quelconque)? $[\lambda T/(\lambda T+1) \mid 1/(\lambda T+1)]$

Exercices

d. Etant donnée qu'une station débute une transmission au temps t , quelle est la probabilité que cette trame soit transmise sans collisions?

$$[1/(\lambda T + 1)^{k-1} e^{-\lambda(k-1)T}]$$

e. Quel est le débit effectif? $[k\lambda/(\lambda T + 1)^k e^{-\lambda(k-1)T}]$

On veut évaluer le débit effectif du protocoles lorsque $k \rightarrow \infty$. Pour cela on suppose que le temps d'attente de chaque station est λ/k et faites $k \rightarrow \infty$ dans la formule obtenue en e (poser $T=1$). $[\lambda e^{-2\lambda}]$

En déduire que l'hypothèse que les transmissions se font selon un processus de Poisson est valide quand le nombres de stations est grand et que les stations transmette avec un probabilité 'petite'.

Exercices

Répétez les points précédents pour le protocole slotted Aloha. Montrez que le débit effectif est $\lambda e^{-\lambda}$ si le nombre de stations est très grand et avec les mêmes hypothèses que précédemment.

Remarquez que pour slotted Aloha les transmissions ne se produisent qu'aux temps 0, T, 2T, 3T, etc. Pour comparer les résultats il faut choisir les temps d'attentes tels que le temps de cycle soit le même que pour Aloha pur.

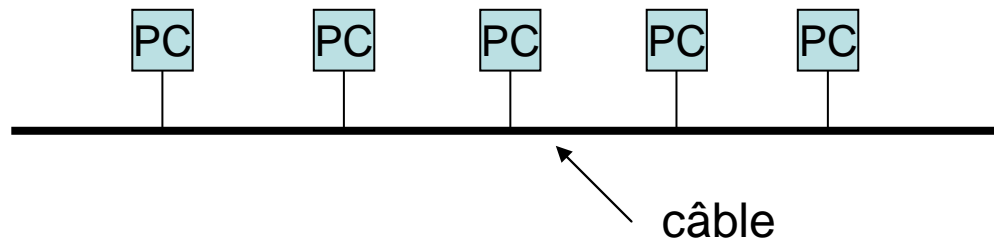
Remarque: On a posé T=1 pour comparer les résultats avec ceux obtenus quand on suppose que les transmissions forment un processus de Poisson de paramètre G.

Ethernet 802.3

Réseaux à diffusion - Ethernet

Un des premier réseaux à diffusion (broadcast) est le réseaux **Ethernet** (802.3).

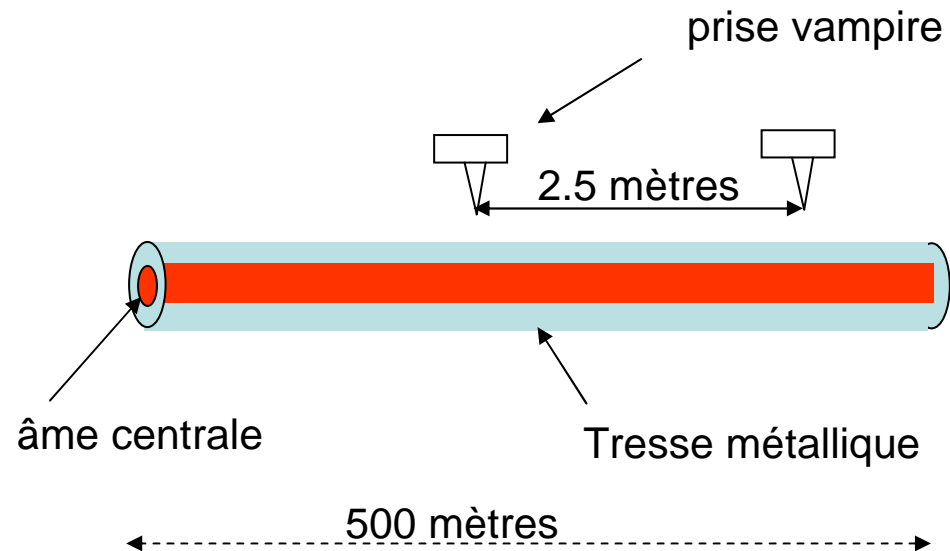
A l'origine le média de communication était un câble de type coaxial sur lequel les stations se 'branchaient' selon une topologie en bus:



10Base5

10Base5: (câble coaxial épais, thick Ethernet)

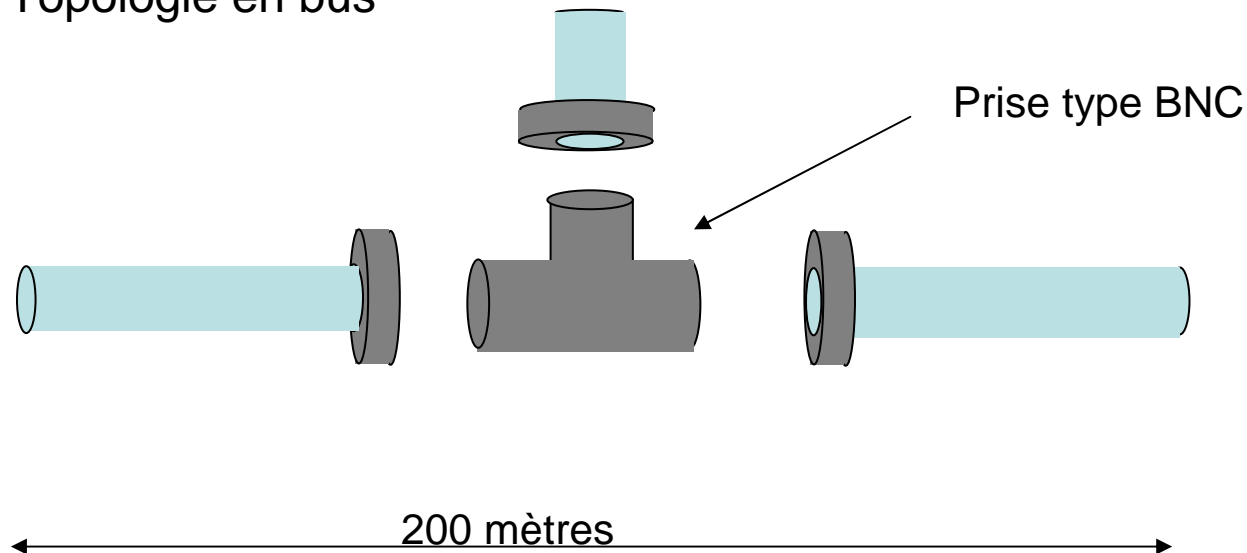
- 10 Mbits/sec
- Transmissions bande de base (NRZ, Manchester, AMI)
- Segments de maximum 500 mètres
- 100 stations maximum par segment



10Base2 – câblage simplifié

10Base2: (câble coaxial fin, thin Ethernet)

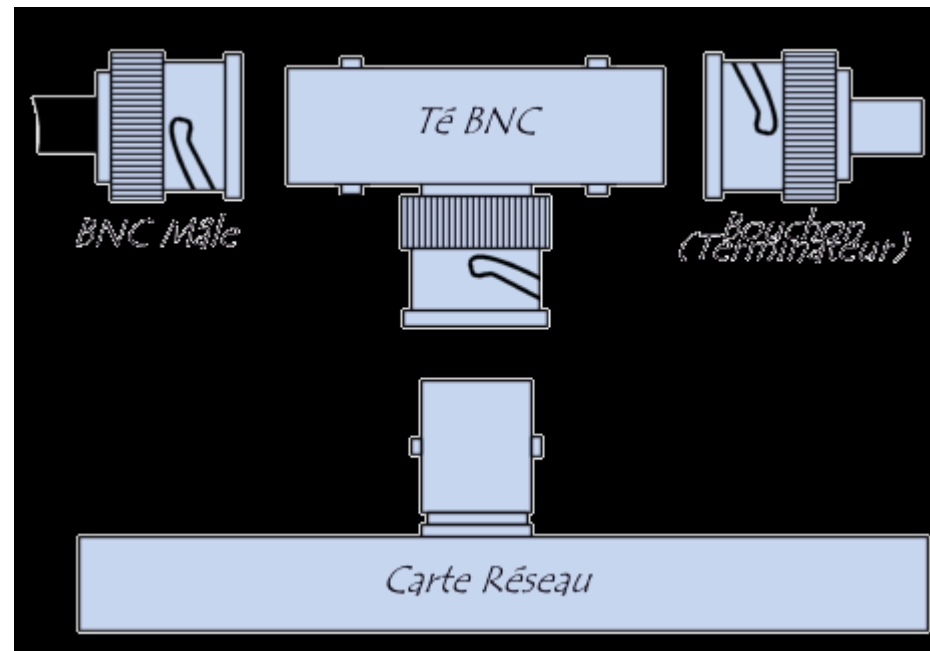
- 10 Mbits/sec
- Transmission en bande de base (NRZ, Manchester, AMI)
- Segments de 200 mètres
- 30 stations maximum par segments
- Topologie en bus



10Base2



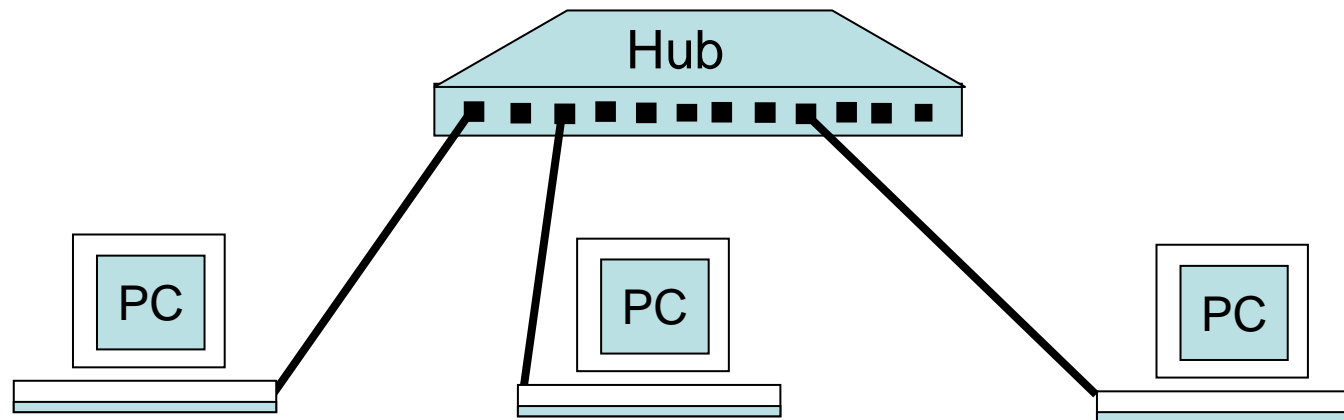
10Base2



10BaseT

10BaseT: (paires torsadées)

- 10 Mbits/sec
- Transmission en bande de base
- Paires torsadées connecteurs RJ45
- Segment de 100 mètres
- Topologie en étoile
- Connecteurs



10BaseT

10baseT:

- Chaque station est reliée au hub central par une liaison point-à-point par un câble à paires torsadées
- Le Hub reconstitue les signaux en provenance des stations (élément actif, composants électroniques).
- Le hub répète un signal en provenance d'une station sur tous ces ports, exceptés sur le port qui est en émission

**Hub = concentrateur de port = répéteur multiport =
transmetteur multiport**

Adaptateur - Tranceiver

Un tranceiver (**trans**mitter – **Rece**iver) permet de connecter l'interface Ethernet au media de communication (câble coaxial, paire torsadée).

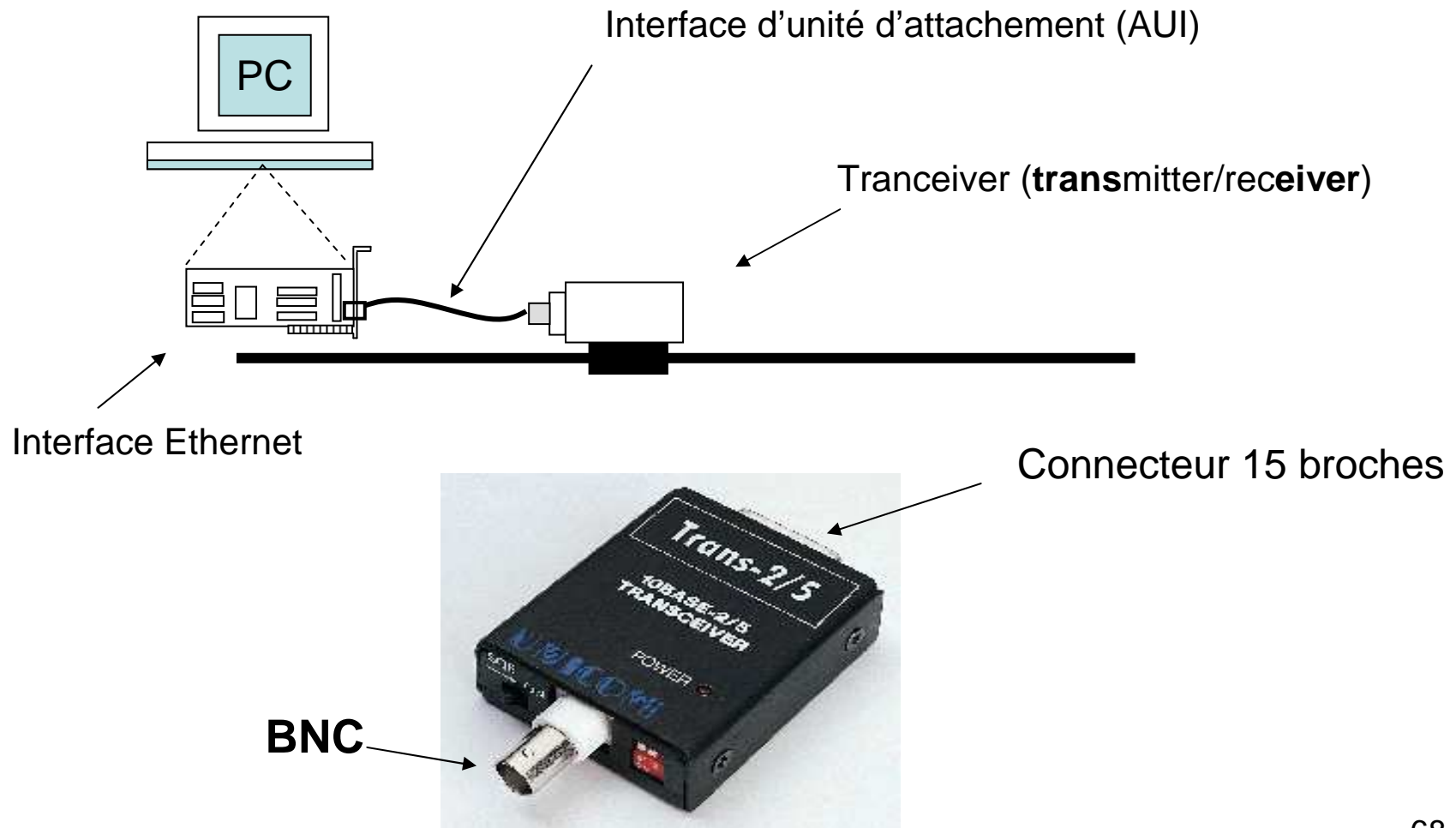
L'intérêt d'utiliser un tranceiver est que l'interface Ethernet est indépendant du média

Différent type de connecteur:

- BNC 10base2
- Connecteur 15 broches 10Base5
- RJ45, 10BaseT

Tranceiver = Unité d'Attachement au Média

Adaptateur - Tranceiver

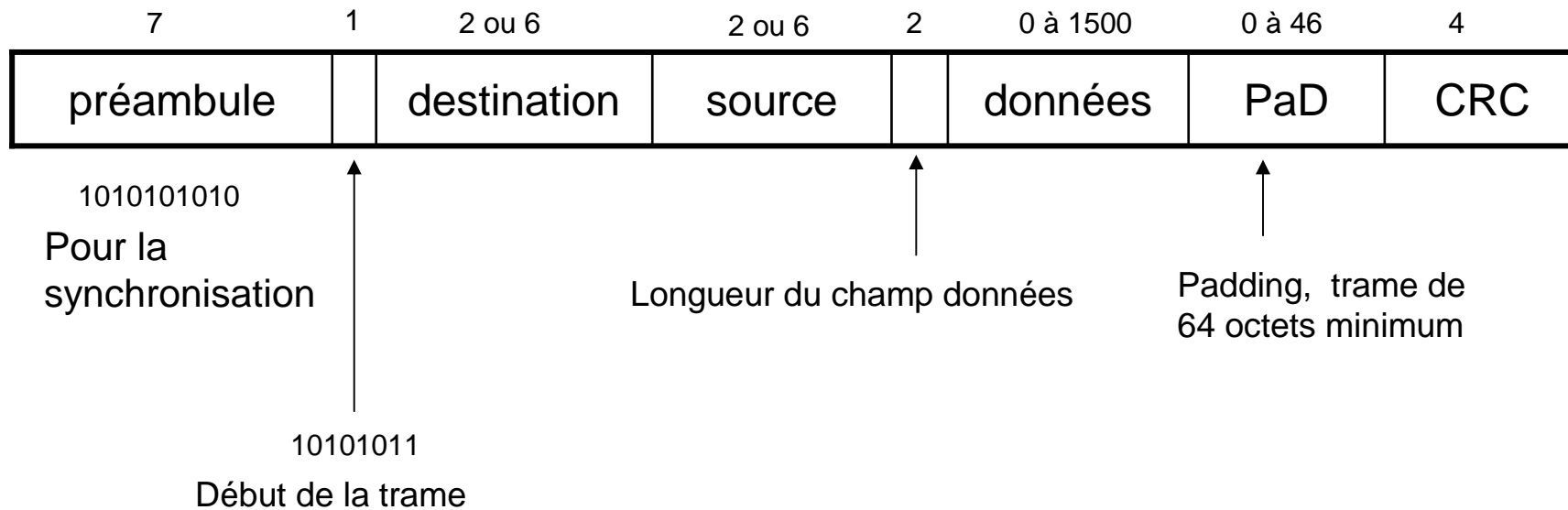


Adaptateur - Tranceiver

Fonctions du tranceiver

1. Transmission de données en bande de base
2. Réception de données
3. Détection de collisions
4. Surveillance fonctionnelle, inhibe la fonction d'émission
5. Possibilité d'interrompre les transmissions si collisions, ou si les données sont excessivement longues. Cette fonction permet d'isoler automatiquement une station défectueuse.

Trame 802.3



Adresse destination:

- tout à 1 est l'adresse de broadcast
- le bit de poids fort à 1 signifie une adresse de groupe (multicast)

Protocole

- À la réception d'une trame à transmettre la couche liaison de donnée teste si le canal est occupé.
- Si non, attend que le canal ne soit plus occupé.
- Si oui, transmet la trame.
- Pendant la transmission l'adaptateur observe qu'une autre station ne transmette pas en même temps (collision).
- Si une collision survient, l'émetteur cesse la transmission et recommence.
- Sinon, la trame est transmise.

Gestion des collisions

Après détection d'une collision:

- Les stations émettrices cessent d'émettre
- Génère une séquence de brouillage pour propager l'information
- Reprenne l'émission après attente d'un temps aléatoire
 - Les stations attendent 0 ou 1 slot
 - Si collisions 0,1,2,3 slots
 - Si collisions 0,1,2,3,4,5,6,7, slots
 - ...
 - Si collisions 0,1,... $2^i - 1$ slots

La phase d'attente s'appelle **phase d'attente exponentielle**
(exponential backoff phase)

Gestion des collisions

L'intervalle de temps qui correspond à un slot est suffisamment grand pour assurer que lorsqu'une station transmet pendant le slot i alors le signal sera détecté par toutes les stations avant le début du slot $i+1$.

La longueur maximale du câble, la vitesse de propagation du signal et l'intervalle de temps d'un slot assurent que une station qui commence à transmettre au slot i sera prioritaire par rapport à une autre qui prévoit de commencer pendant un slot $i+1$, $i+2$, ...

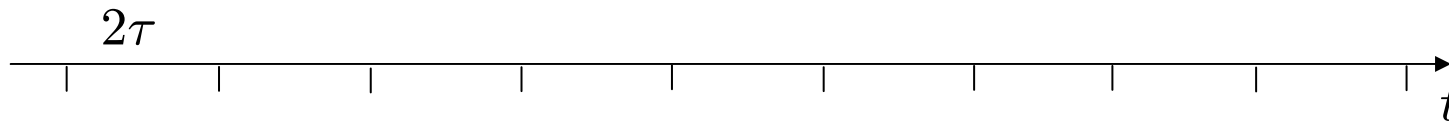
Acquittement

Lorsqu'une trame est envoyée sans qu'une collision soit détectée, toutes les stations laissent le slot suivant la réception libre pour que la station réceptrice puisse transmettre son acquittement (sans collisions)

Le protocole fait partie de la famille des protocoles CSMA/CD = Carrier Sense Multiple Access/Collision detection

Analyse des performances

Protocole discrétisé

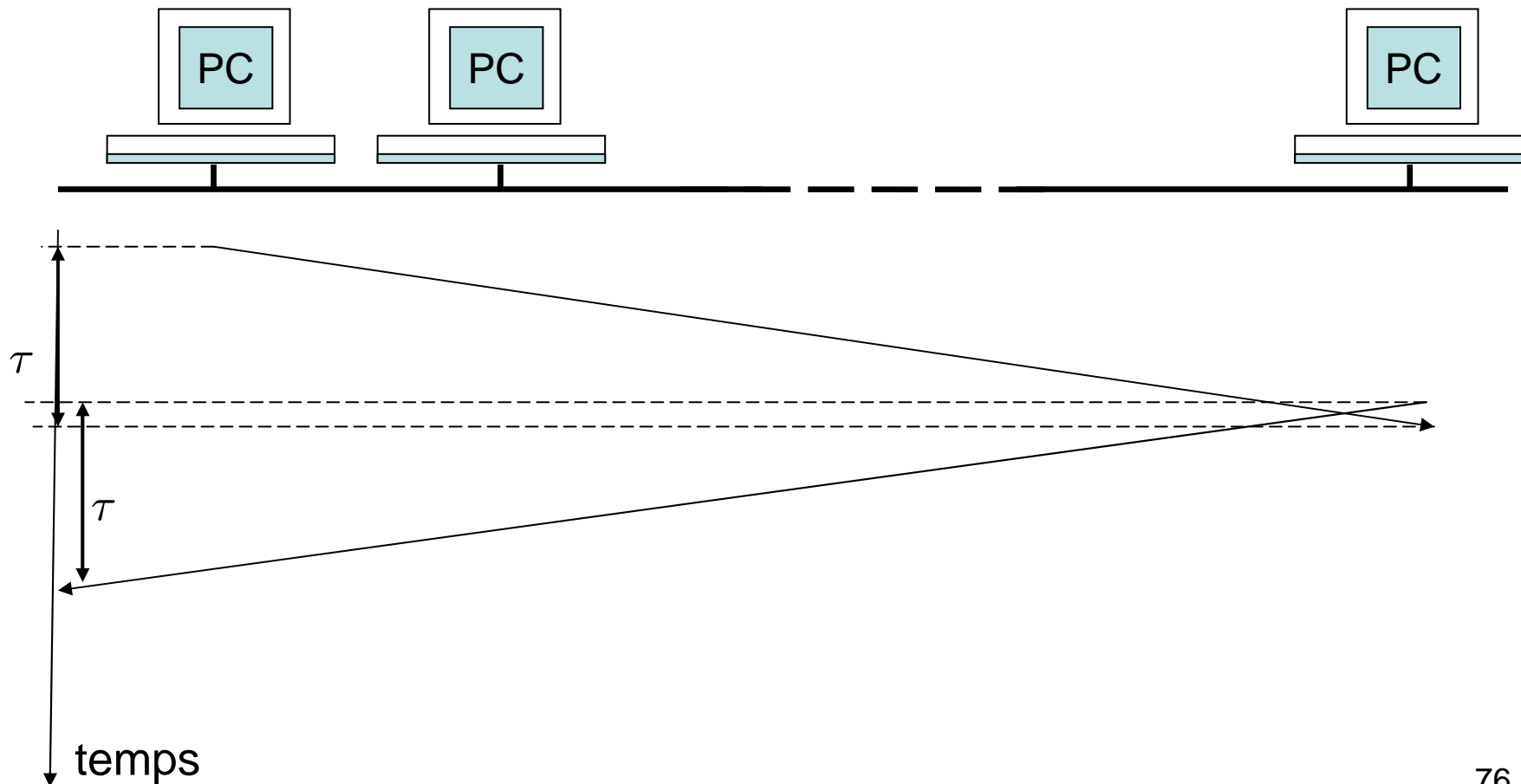


On a k stations qui émettent pendant un slot avec une probabilité p

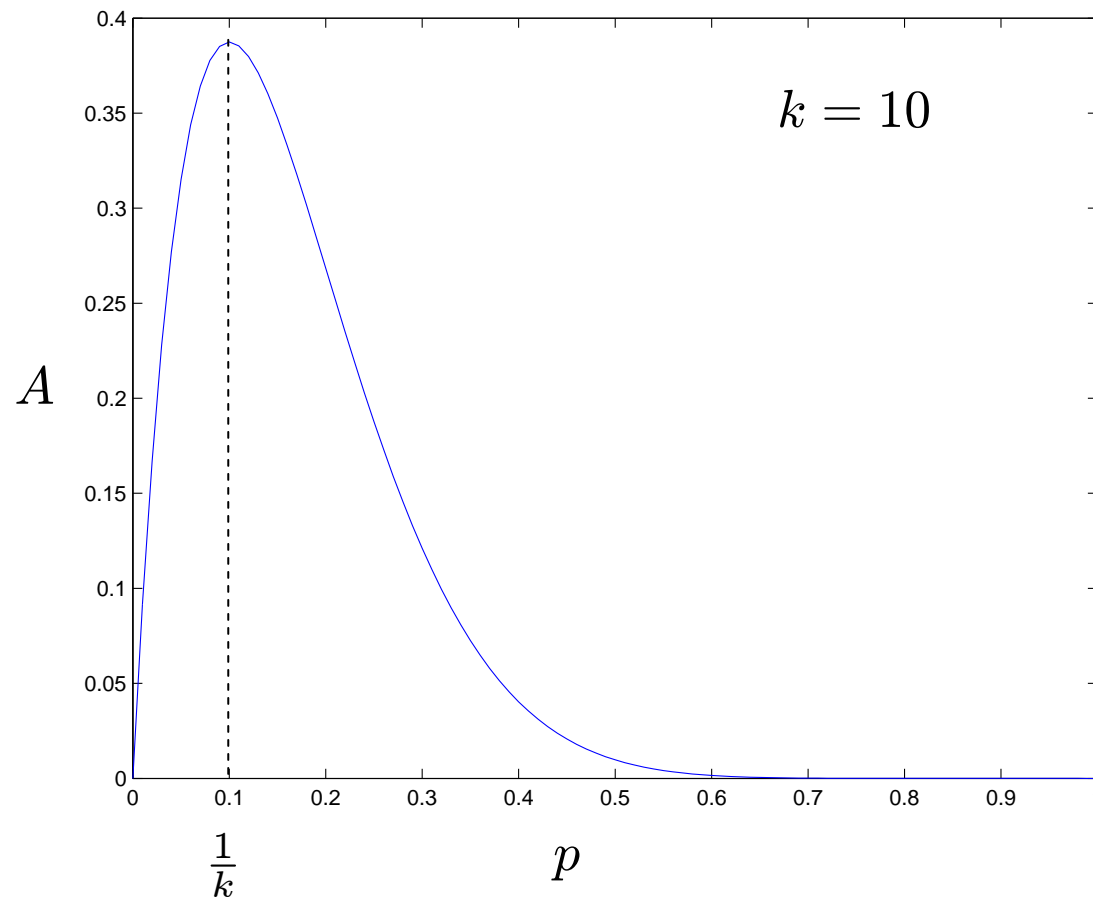
$A = kp(1 - p)^{k-1}$ est la probabilité qu'une station émette sans collision

τ est le temps maximum que met le signal pour être détecté par toutes les stations. La durée d'un slot est 2τ pour s'assurer qu'on détecte les collisions.

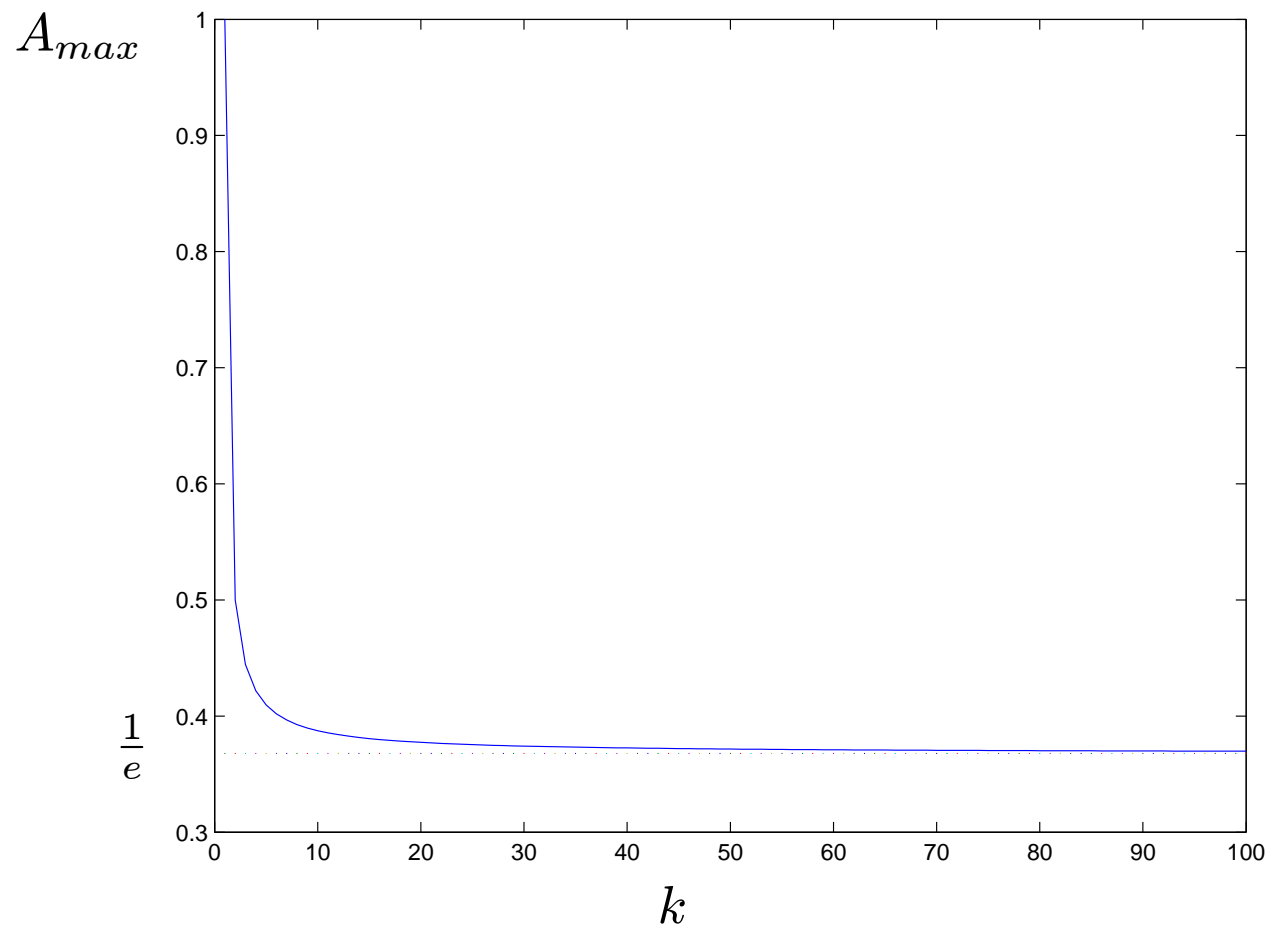
Analyse des performances



Analyse des performances



Analyse des performances



Analyse des performances

Probabilité que le nombre de slots de contention soit j , c'est-à-dire le nombre de slots total avant émission correcte d'une trame

$$A(1 - A)^{j-1}$$

Et le nombre moyen de slot de contention est

$$\sum_{j=0}^{\infty} j A (1 - A)^{j-1} = \frac{1}{A} < e$$

Et donc on a l'efficacité moyenne du canal, (P est la durée d'émission d'une trame

$$\frac{P}{P + 2\tau/A}$$

Analyse des performances

$$P = F/B$$

F la longueur des trames bits
 B le débit

$$\tau = L/c$$

L longueur du segment
 c vitesse de propagation du signal

$$A = \frac{1}{e}$$

optimal

Efficacité du canal

$$\frac{1}{1+2BLE/(cF)}$$

Exercices

On note λ le paramètre de la loi exponentielle pour le temps d'attente, T le temps de transmission d'un trame, 2τ la durée de la période vulnérabilité et B la durée de la séquence de brouillage si une collision est détectée.

Décrivez les cycles de transmissions et calculez le débit effectif. Comparez avec les résultats obtenus pour Aloha pur et slotted.

Remarque: Ignorez les phases d'attente exponentielles.

Protocole à jetons

Protocole de partage des ressources

Les protocoles d'accès au canal de types ALOHA ou Ethernet n'assurent pas un débit minimal aux stations.

En particulier ces réseaux ne permettent pas le développement d'applications temps-réels (= les temps de réponse sont bornés).

Les protocoles de partage de ressources cherchent à palier à cet inconvénient.

Protocoles à jeton

Une trame de contrôle particulière est définie et s'appelle le jeton.

Cette trame est transmise de station en station de manière cyclique.

Seule la trame qui dispose du jeton est autorisée à transmettre des données.

On suppose que chaque les stations dispose de l'adresse de la station successeur dans le cycle de m machines.

On veut assurer à chaque station une période de transmission de H_k secondes lorsqu'elle dispose du jeton

Protocoles à jeton

On note T_{TRT} le temps de rotation du jeton ce qui assure un débit de $\frac{H_k B}{T_{TRT}}$ [bits/secondes] si B [bits/secondes] est le débit du canal.

On a la condition

$$\sum_{k=1}^m H_k \leq T_{TRT}$$

Si les stations n'utilisent pas complètement la période H_k réservée alors le jeton va circuler en moins de temps que T_{TRT} .

L'idée est d'utiliser ce temps pour transmettre du trafic asynchrone (non urgent). Par exemple des trames de contrôles pour inviter d'autres stations à se joindre au cycle (l'anneau).

Protocoles à jeton

Les stations vont donc mesurer le temps que met le jeton pour effectuer un cycle, ce temps sera mesuré par la variable TRT_k . (L'indice k indique que c'est une variable de la station k.)

En fait, TRT_k est un timer, c'est-à-dire que la station k l'initialise avec une valeur T (en secondes) et démarre le timer. Le timer attend T secondes et lève une interruption.

Chaque station utilise un compteur LC_k et un timer THT_k pour le trafic asynchrone.

Protocoles à jeton

T_{TRT} : Temps de rotation du jeton optimal (Target Token Rotation Time)

TRT_k : Temps de rotation du jeton mesuré à la station (Token Rotation Time)

LC_k : compteur

THT_k : Temps d'émission de trafic asynchrone (Token Holding Time)

Algorithme

Initialisation : a) $THT_k = 0$, $LC_k = 0$, $TRT_k = T_{TRT}$, $k = 1, \dots, m$
b) **démarrer** TRT_k

Si $TRT_k = 0$ **alors** $TRT_k = T_{TRT}$, $LC_k = LC_k + 1$ (*interruption*)
redémarrer TRT_k

Si jeton arrivé **alors**

Si $LC_k = 0$ **alors** (jeton en avance, transfert trafic asynchrone possible)

$THT_k = TRT_k$, $TRT_k = T_{TRT}$

démarrer TRT_k

transfert synchrone pendant H_k **unités de temps**

démarrer THT_k

transfert asynchrone tant que $THT_k > 0$

passer le jeton à la station suivante

Si $LC_k = 1$ **alors** (jeton en retard)

transfert trafic synchrone pendant H_k **unités de temps**, $LC_k = 0$

passer le jeton à la station suivante

Si $LC_k > 1$ **alors** activer une procédure de récupération d'erreur

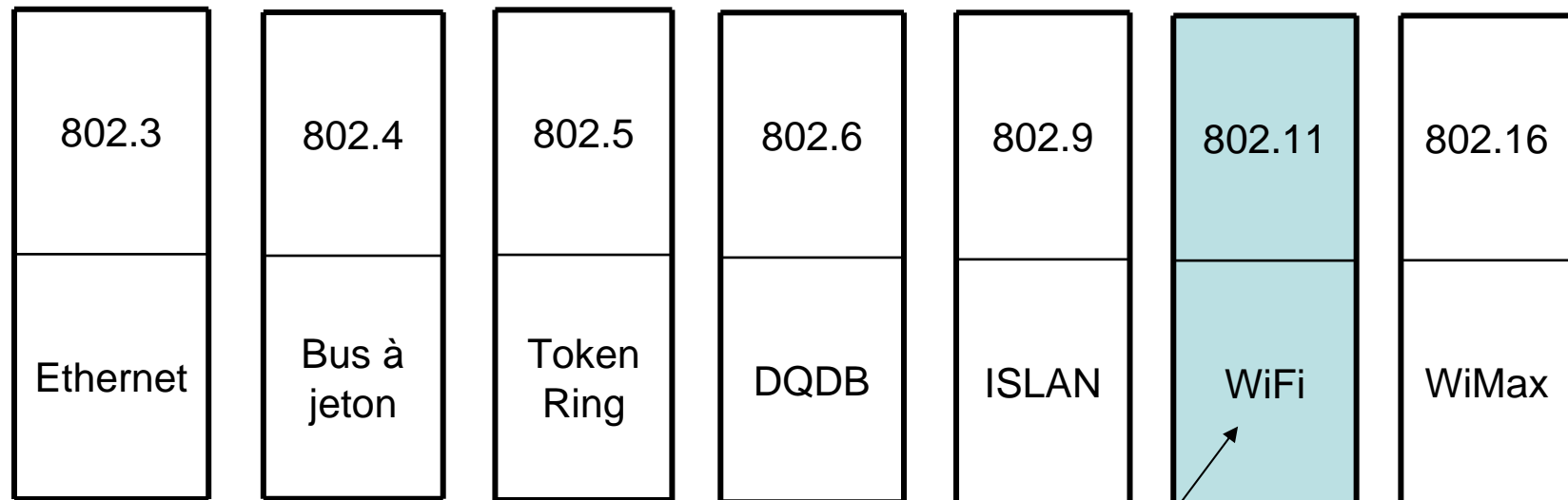
Réseaux radio 802.11

Contraintes caractéristiques

1. Une **atténuation** rapide, difficilement prédictible et dépendant de la position des stations des signaux transmis. Technique de codage et recouvrement des erreurs plus élaborées quand celles utilisées pour les réseaux câblés.
2. Limitation quand à la **puissance** des signaux émis (législation, technique)
3. L'**énergie** est souvent une ressource critique pour les utilisateurs des réseaux sans fil. En effet, ces réseaux sont souvent associés à des stations mobiles et alimentées par batterie.
4. Plus difficile d'assurer la **sécurité** des transmissions, le medium de transmission est facile à espionner.
5. La mobilité des stations implique que la topologie des réseaux de communications change avec le temps.

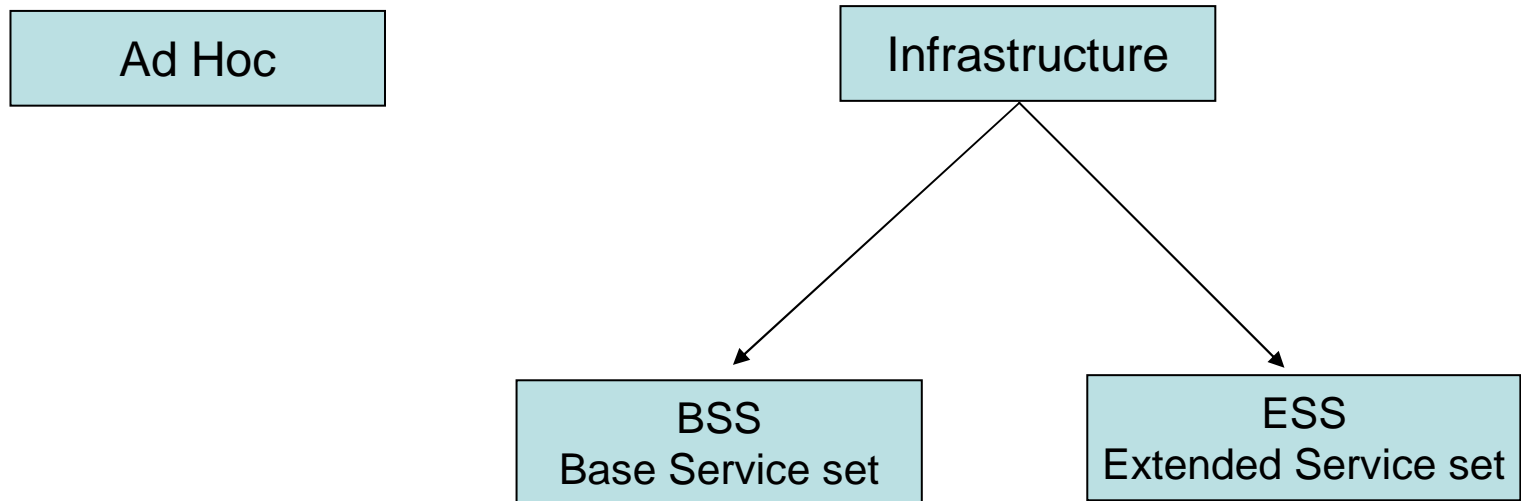
IEEE standards

802.2 couche liaison de données (Logical Link Layer LLC)
802.11 couche d'accès au canal (Medium Access Control MAC)



802.11a, 802.11b, 802.11c, ...

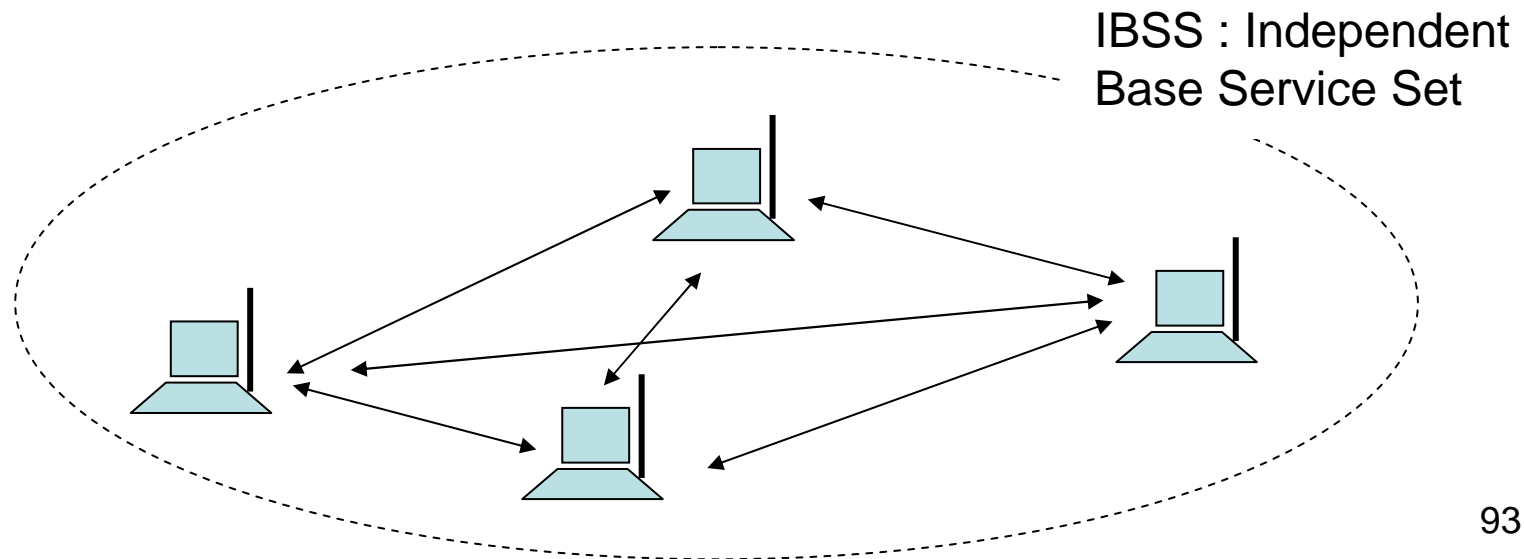
Topologies



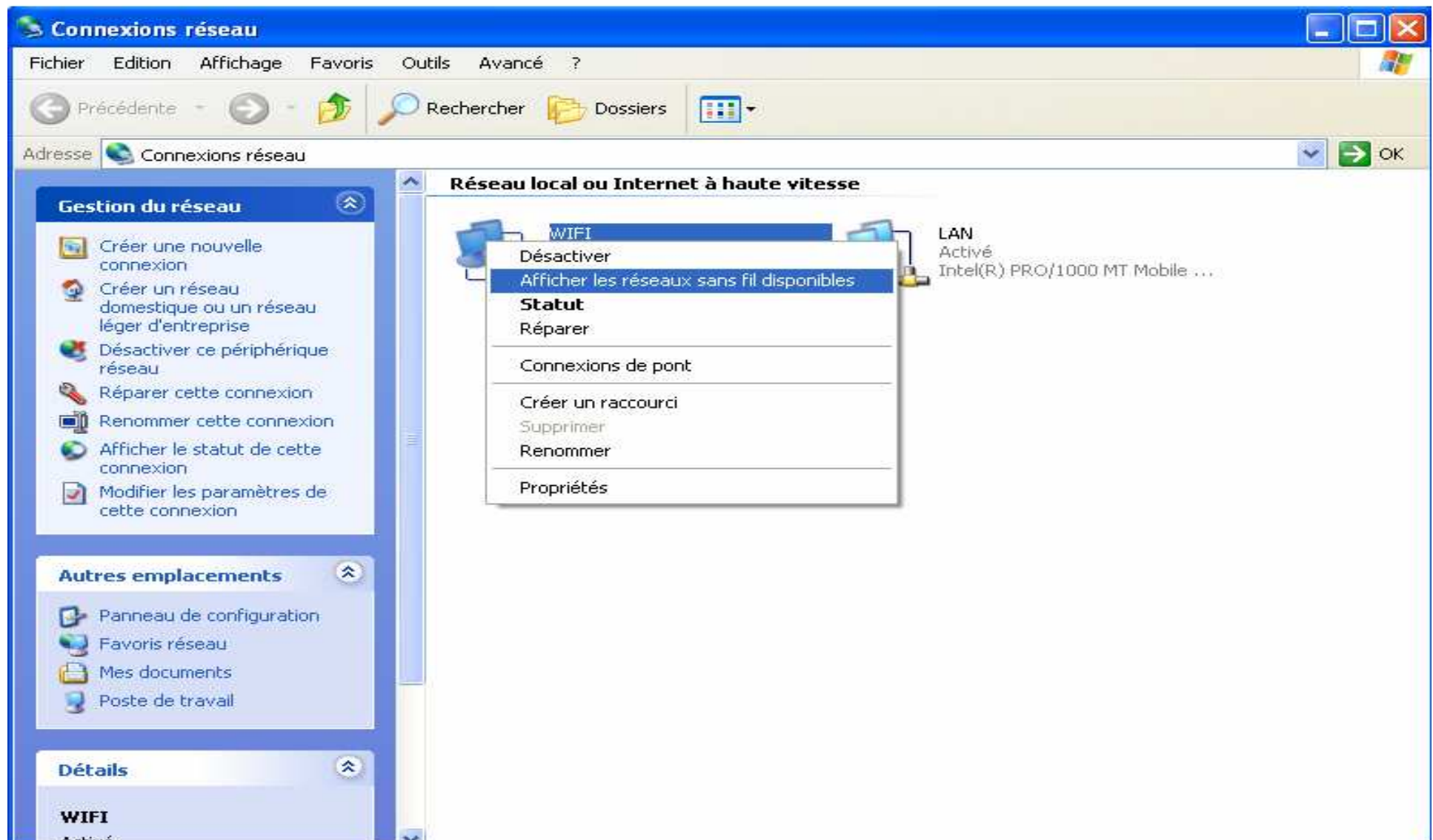
Ad Hoc

En Mode Ad Hoc (Independent/Ad Hoc Base Service Set) le réseau est composé de stations qui communiquent sans relais.

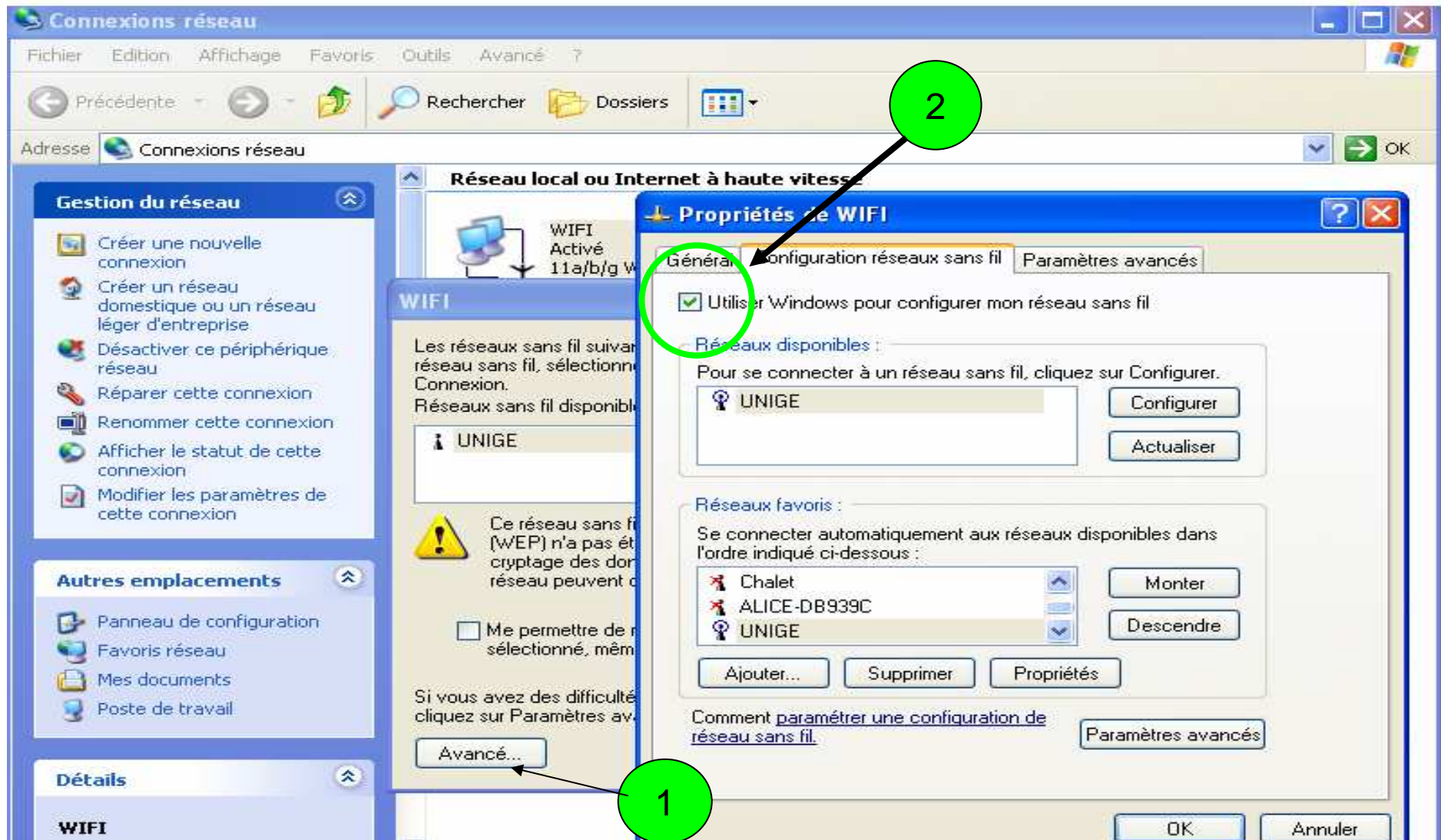
Les stations sont capables de se 'reconnaître' et d'établir des connexions point-à-point (peer-to-peer), c'est-à-dire que pour communiquer deux stations doivent se trouver dans le rayon de communication de l'autre.



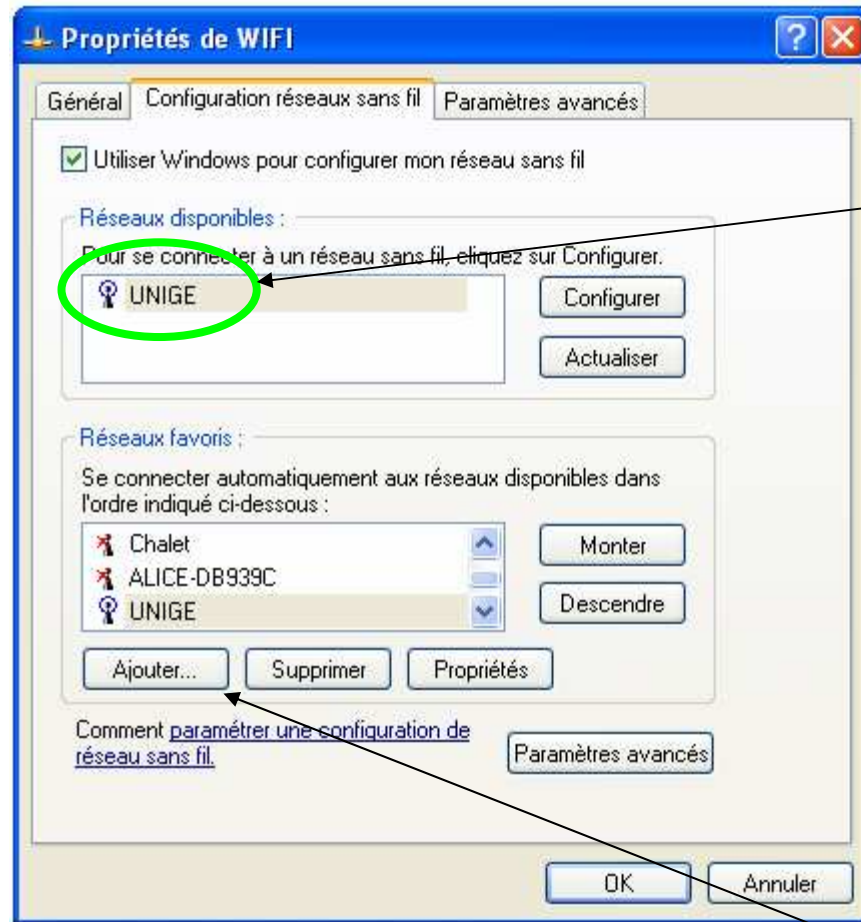
Ad Hoc – Windows XP



Ad Hoc – Windows XP

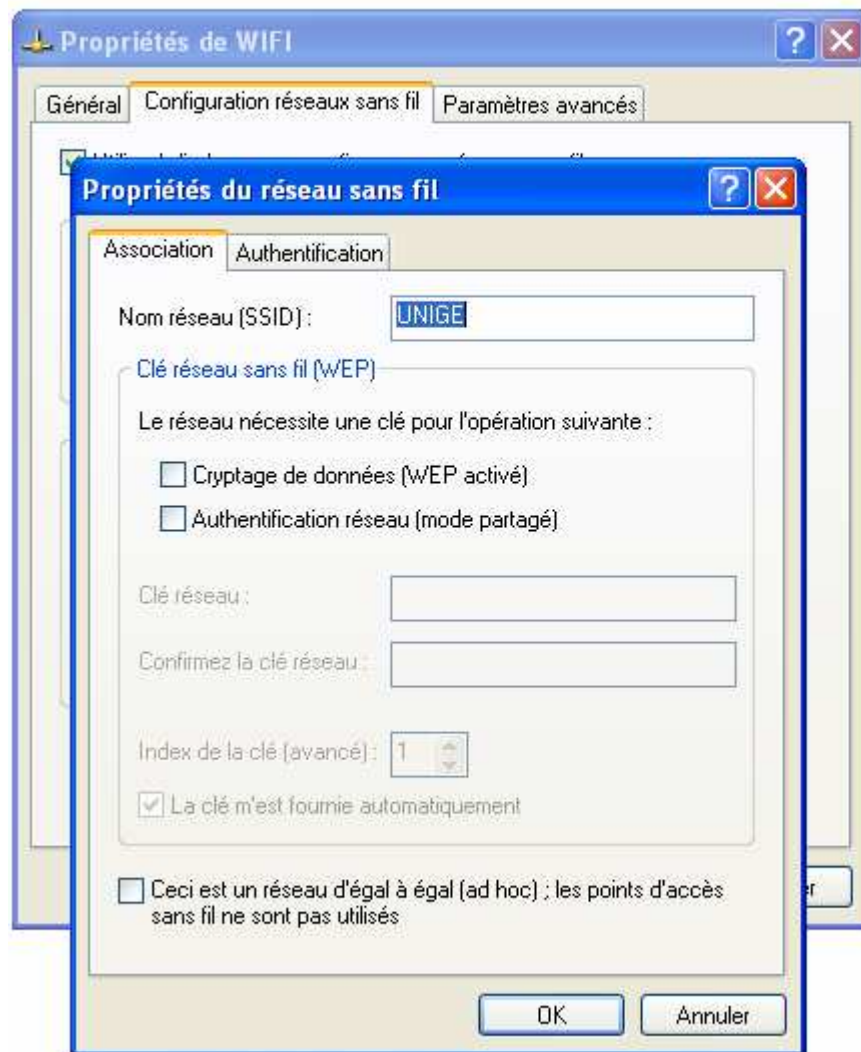


Propriétés



SSID (Server Set ID): première sécurité, seules les stations qui partagent le même SSID peuvent communiquer ensemble

Propriétés



Configurer un réseau ad-hoc

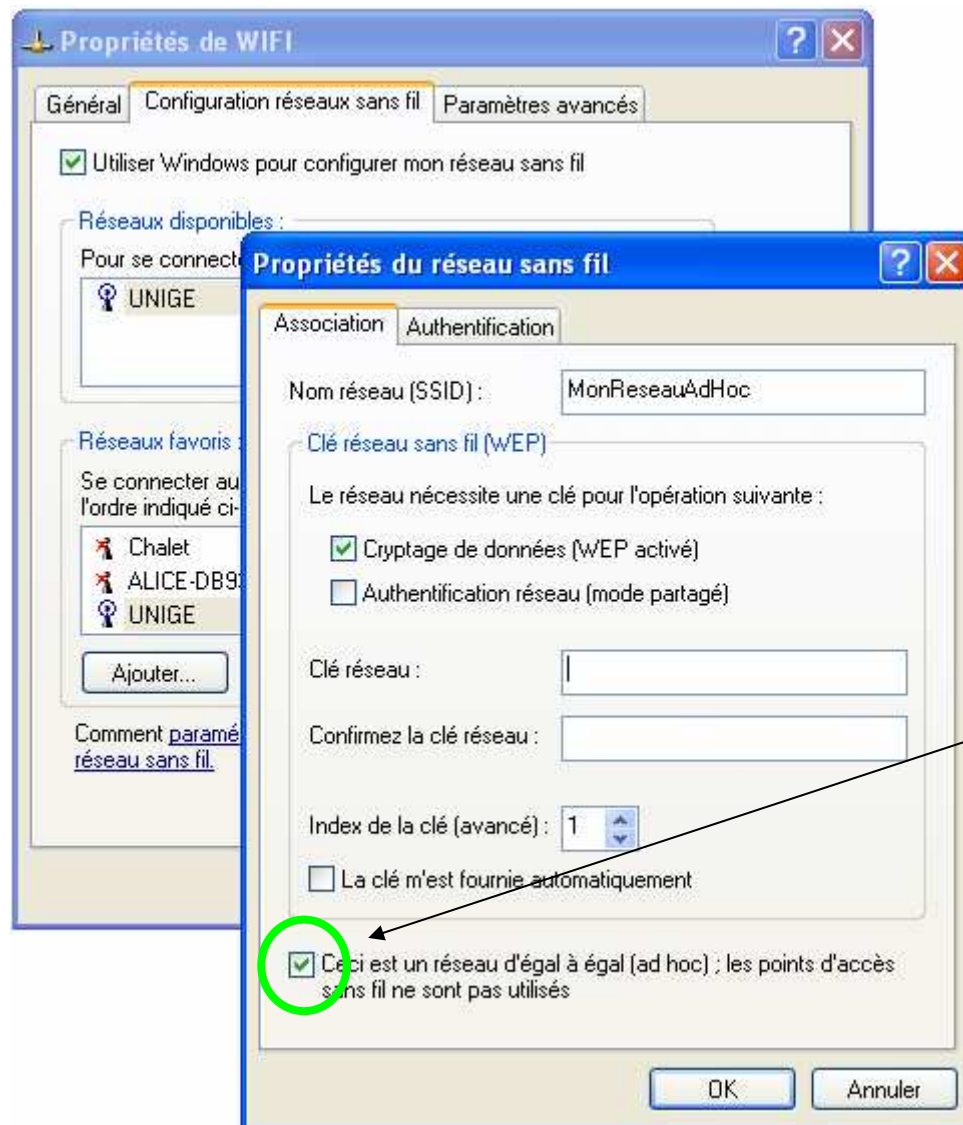
But: Paramétrer deux ordinateurs pour qu'il puissent communiquer ensemble (échange de fichiers,...) de manière ad-hoc.

Une fois les deux ordinateurs configurés ils peuvent établir des connexions point-à-point pour échanger des fichiers,... (utiliser les adresses IP qui appartiennent au même sous-réseau)

Si le nombre de machines connectées est plus important le réseau peut s'auto-configurer, par exemple en utilisant des algorithmes de routages, en être utilisé comme un réseau classique.

auto-configurer est important, aucune machine n'est privilégiée par rapport aux autres, si une machine doit être distinguée c'est le réseau des machines qui doit le faire collectivement (élection).

Configurer un réseau Ad-Hoc



Le réseau est ad hoc

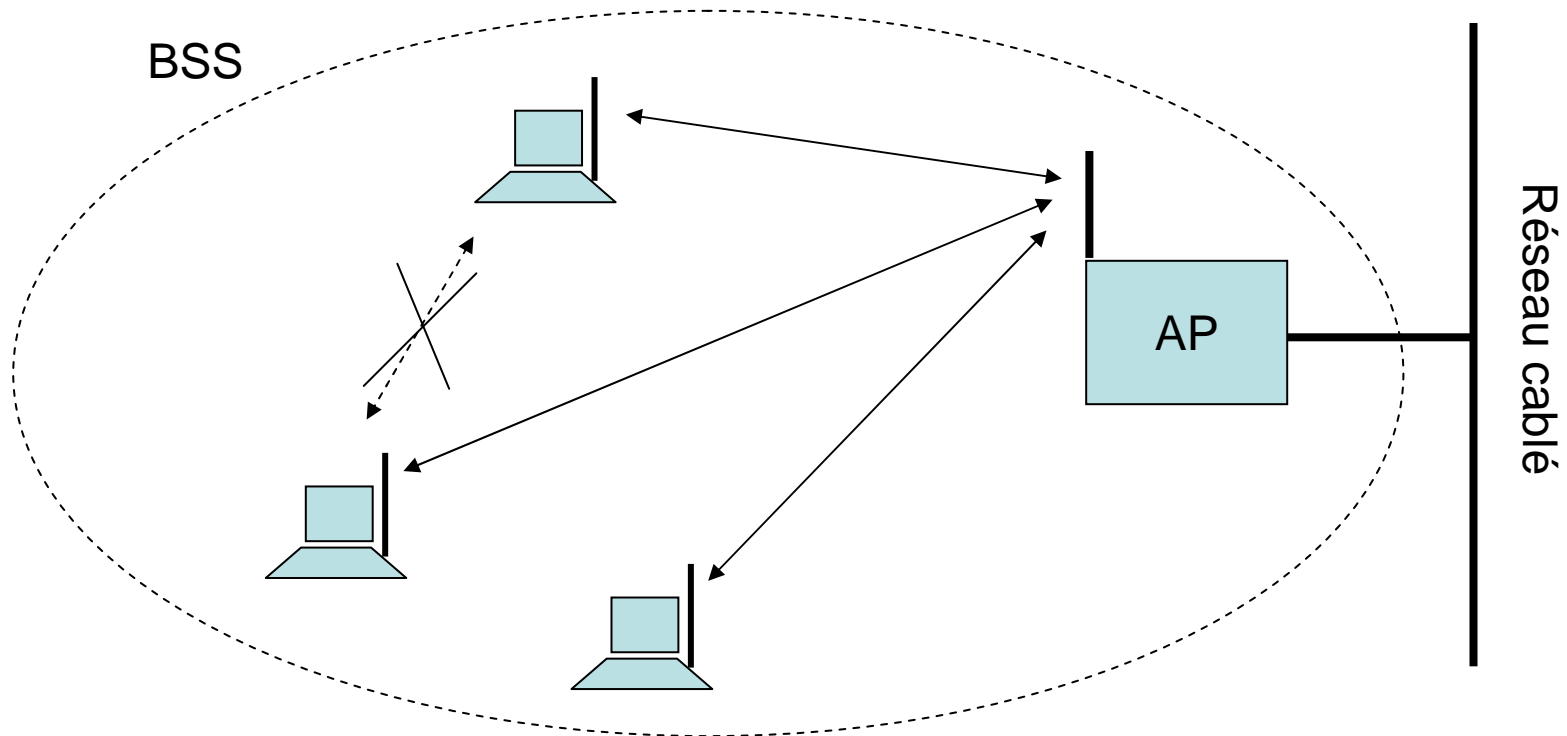
Configurer un réseau Ad Hoc



Avant utilisation il faut contrôler que les adresses IP des stations qui communiquent appartiennent au même sous-réseau (le masque de correspondre)

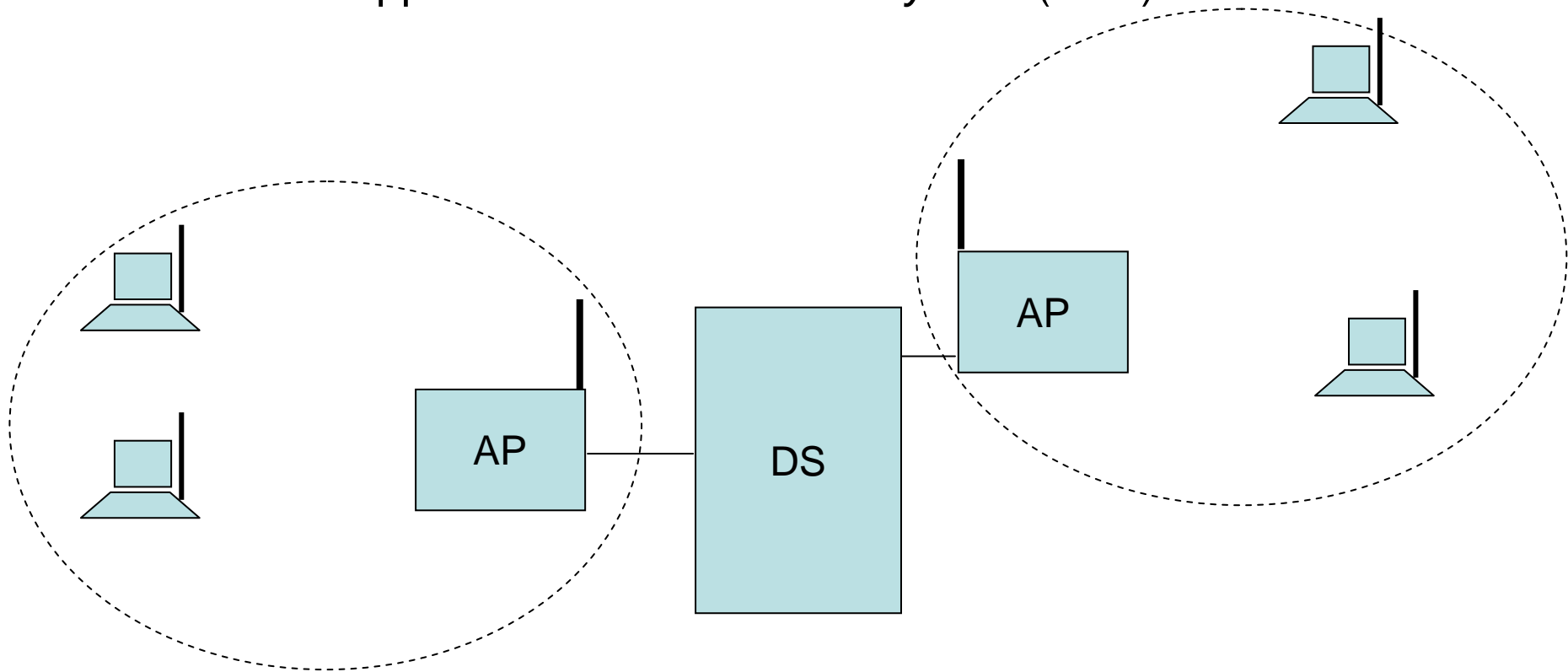
Mode Infrastructure

Le mode *Infrastructure* est le mode généralement mis en œuvre et permet l'interconnexion d'un réseau sans fil avec un réseau câblé conventionnel en utilisant un élément dédié appelé *Acces Point* (AP).



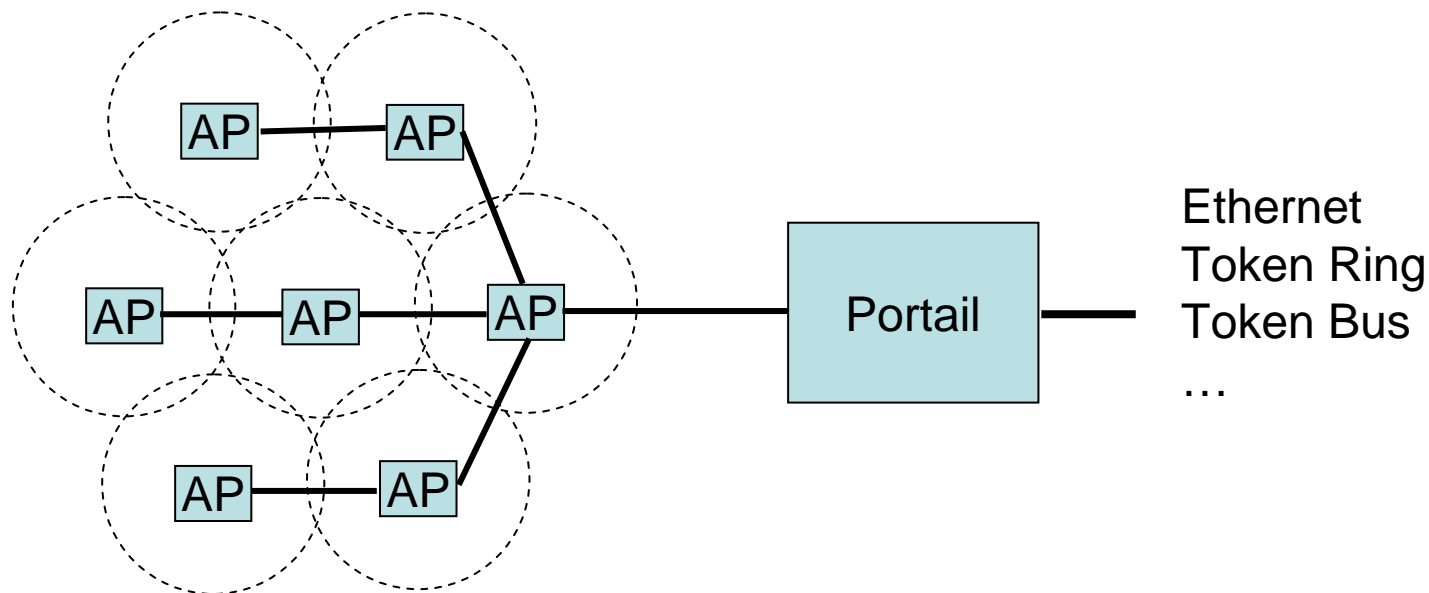
Mode Infrastructure

Les Accés point sont utilisés pour faire transiter l'information entre différents BSS par une entité appelée *Distribution system* (DS). L'ensemble s'appelle *Extended Service System* (ESS).



Mode Infrastructure

L'élément qui permet de connecter un point d'accès (AP) à un réseau qui n'est pas du type IEEE 802.11 s'appelle un portail (portal).



Mode Infrastructure

Un point d'accès (AP) permet de connecter environ 30 stations dans un rayon de communication de 100 mètres. Les chiffres varient en fonction:

- Communication intra – extra muros (outdoor, indoor)
- La position du point d'accès
- Les obstacles à la transmission des ondes
- La puissance d'émission des stations
- L'émission de signaux dans des bandes de fréquences proches
- ...

BSS

Pour rejoindre un BSS (Basic Service Set) une station à deux possibilités:

1. Scan Passif : la station attend de recevoir une trame *BEACON* qui est émise périodiquement par le point d'accès (AP). Cette trame contient des informations permettant la synchronisation de la station avec le point d'accès.
2. Scan Actif : la station émet une trame *Probe Request frame* et attend une éventuelle trame *Probe Response* en provenance d'un point d'accès (une station en mode Ad-Hoc).

Ensuite les stations échangent des informations d'*authentication* et d'*association* (défini les protocoles utilisés par les stations – AP)

La couche d'accès au canal MAC

MAC = Medium Access Control

La couche MAC fait le lien entre la couche physique et la couche liaison de données. Principalement, elle est responsable de gérer les accès au canal de transmission, c'est-à-dire d'implémenter un protocole qui permet d'éviter que plusieurs stations émettent simultanément.

Deux types de fonctionnement sont distingués et peuvent coexister

1. Des périodes de temps sans contention pour accéder au canal (**CFP**, Contention-Free Period). Le canal est géré de manière **centralisée** par le point d'accès (AP), on parle de mécanisme **PCF** (Point Coordination Function)
2. Des périodes de contentions (**CP**, contention period), les stations doivent par elle-même, de manière **distribuée**, accéder au canal au bon moment, on parle de mécanisme **DCF** (Distributed Coordination Function)

La couche d'accès au canal MAC

3 types de données sont transmis (MDPU, Mac Data Protocol Unit)

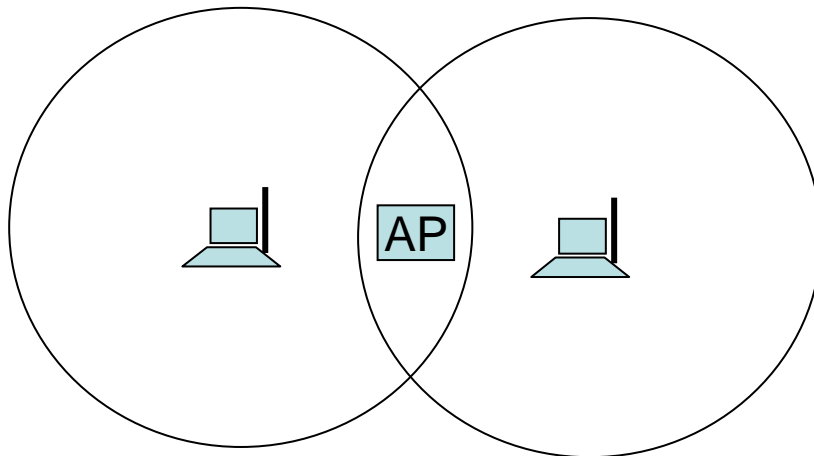
1. Des trames de gestion (management frame) qui permettent d'associer/dissocier une station à un point d'accès (AP)
2. Des trames de contrôle qui permettent d'arbitrer les accès au canal (RTS, CTS), d'acquitter des transferts (positif)
3. Des trames de données

La couche d'accès au canal MAC

Cette couche est similaire à celle implantée dans les réseaux Ethernet (CSMA/CD). Le mécanisme d'accès au canal est du type:

Ecouter avant d'émettre

Néanmoins, avec des communications sans fil il est possible qu'une station soit hors du rayon d'émission d'une station émettrice et donc qu'elle transmette sans que le canal soit libre (*hidden terminal*)



Protocole de base – MAC

Protocole distribué

Le protocole de base est du type CSMA utilisé pour les transmission de type unicast. Un temps d'attente aléatoire est tiré avant chaque transmission.

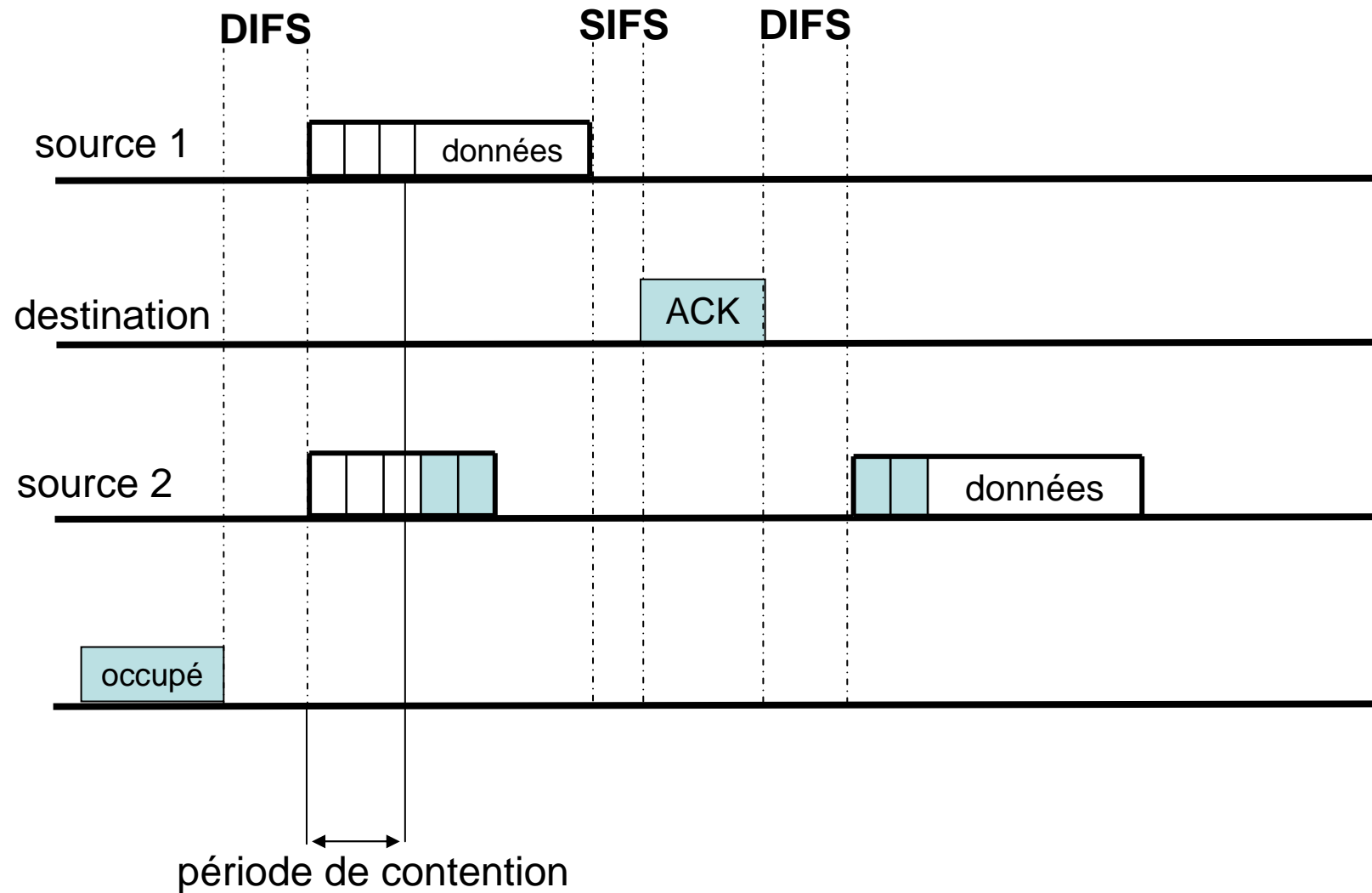
- Une station attend que le canal soit libre avant d'émettre
 - Si le canal est occupé la station attend – sans décrémenter le temps d'attente.
- Lorsque le canal est libre et le délai d'attente aléatoire écoulé la station émet une trame (frame).
- La station réceptrice acquitte le transfert immédiatement.
- Si la trame n'est pas acquittée la station ré-exécute le protocole.
 - le temps pendant lequel le canal est occupé n'est compté dans le délai d'attente.
 - Le délai d'attente maximum est doublé chaque fois qu'un paquet est perdu.

Protocole de base - MAC

Plus précisément, la gestion des temps d'attente se fait de la manière suivante:

- Lorsque le canal devient libre il doit le rester pour un intervalle de temps **DIFS** (DCF Inter-Frame Space, DCF = Distributed Coordination Function)
- Si le canal est resté libre les station attendent une temps aléatoire compté en unité de temps (time slots) de 20μs, par défaut le nombre de slots d'attente appartient à [0:31] (Contention window).
- Après réception le paquet est acquitté dans un intervalle de temps appelé **SIFS** (Short inter-Frame Space) plus court que le temps DIFS.
- Les stations en attente du canal recommence à compter le temps d'attente seulement après un intervalle de temps **DIFS** après l'acquiescement.

Protocole de base - MAC



Protocole de base - MAC

Le délai d'attente est compris dans un intervalle $[0, x]$ avec x compris dans l'intervalle $[CW_{min}, CW_{max}]$.

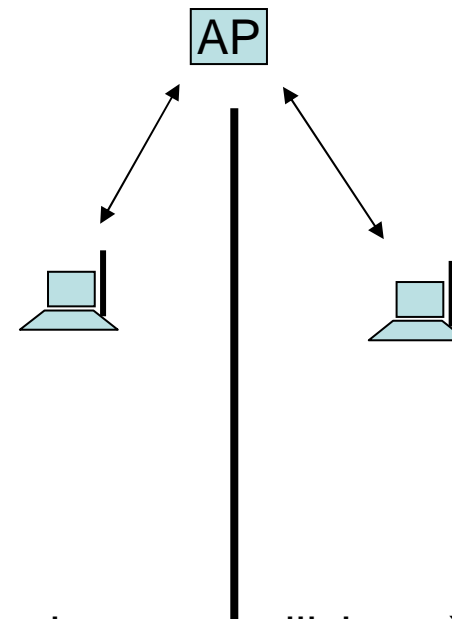
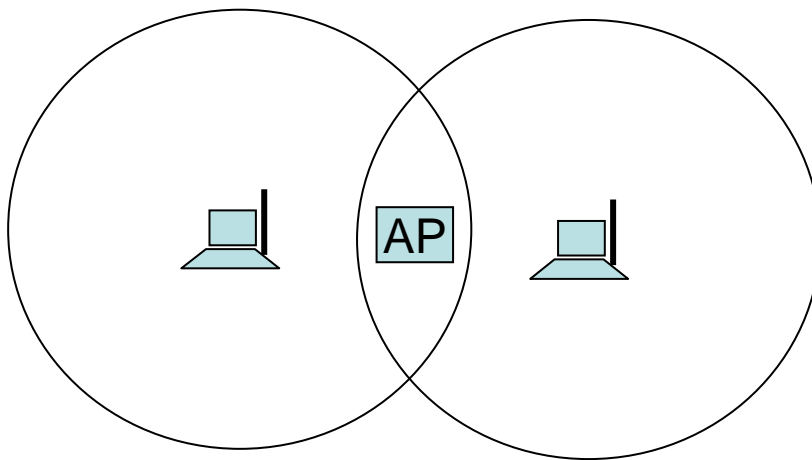
Les tailles de fenêtres sont toujours des puissance de deux.

Si un paquet est transmis avec succès le délai maximum est réinitialisé à CW_{min} ($x = CW_{min}$).

A cause du doublement du délai maximum d'attente le protocole est qualifié de binary backoff exponential

Stations cachées

Il existe des configurations dans lesquelles une stations ne perçoit pas le signal d'une station émettrice.



Dans ces situations la probabilité d'avoir de nombreuses collisions à répétition est très élevée car un émetteur ne détecte jamais la présence de l'autre. La norme 802.11 propose un mécanisme RTS/CTS.

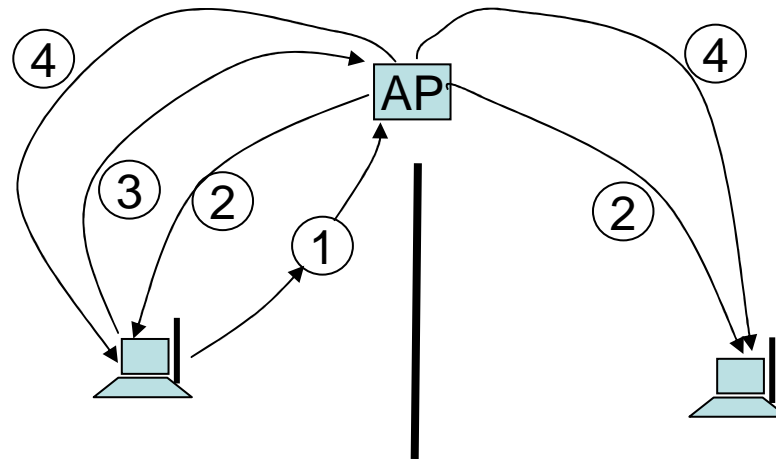
RTS/CTS

La norme 802.11 prévoit un mécanisme basé sur des paquets de contrôle pour attribuer le canal de transmission à un émetteur.

Un émetteur commence par émettre un petit paquet **RTS** (Request To Send) pour annoncer qu'il désire émettre. La taille du paquet est petite pour limiter les collisions.

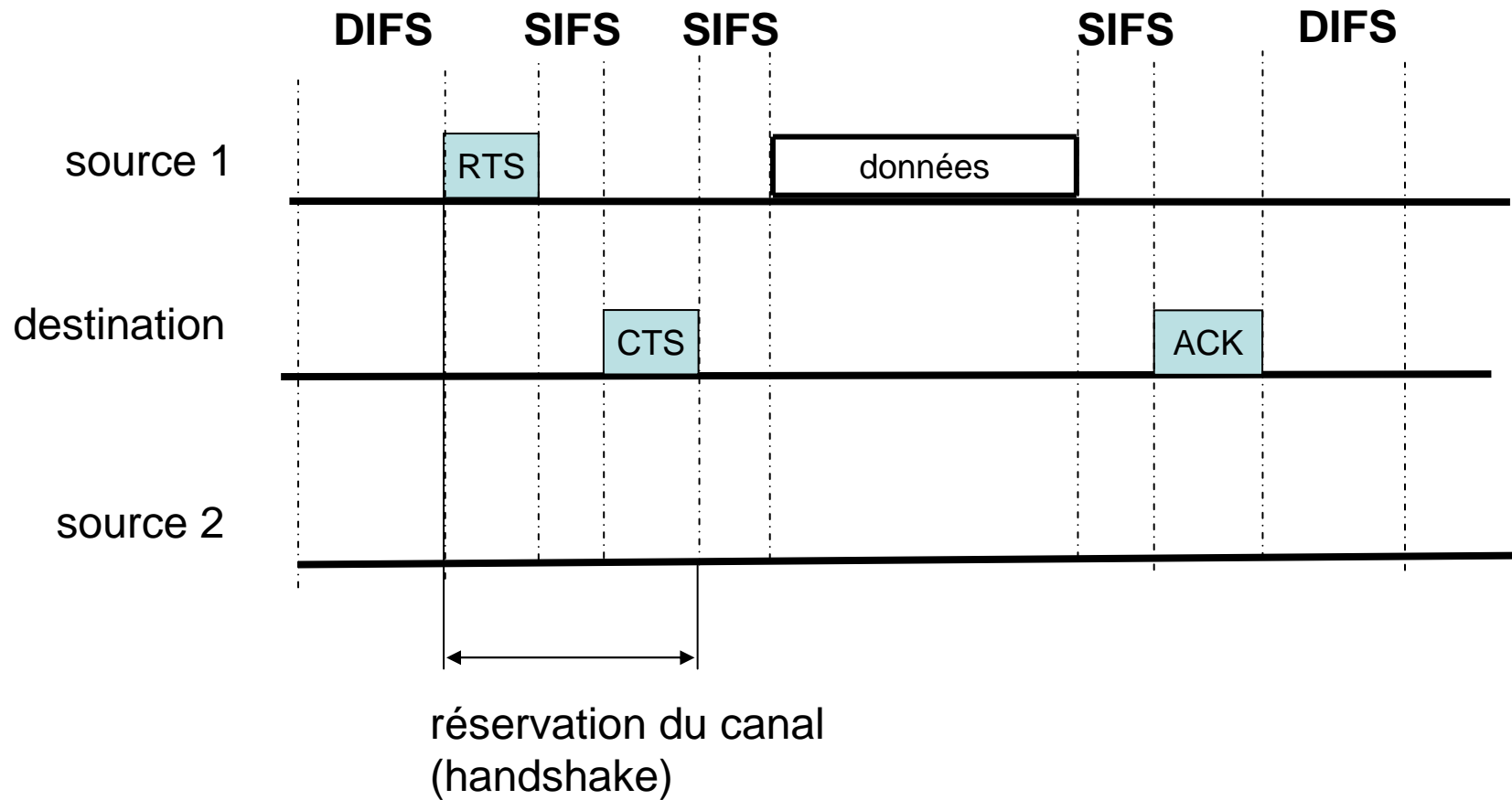
Le récepteur émet un **CTS** (Clear To Send) à une station pour lui signifier qu'elle peut utiliser le canal. Le paquet CTS est diffusé à toutes les stations qui considère le canal comme occupé.

RTS/CTS



- ① RTS Request To Send
- ② CTS Clear To Send
- ③ Données
- ④ ACK, Acquittement

RTS/CTS



RTS/CTS - NAV

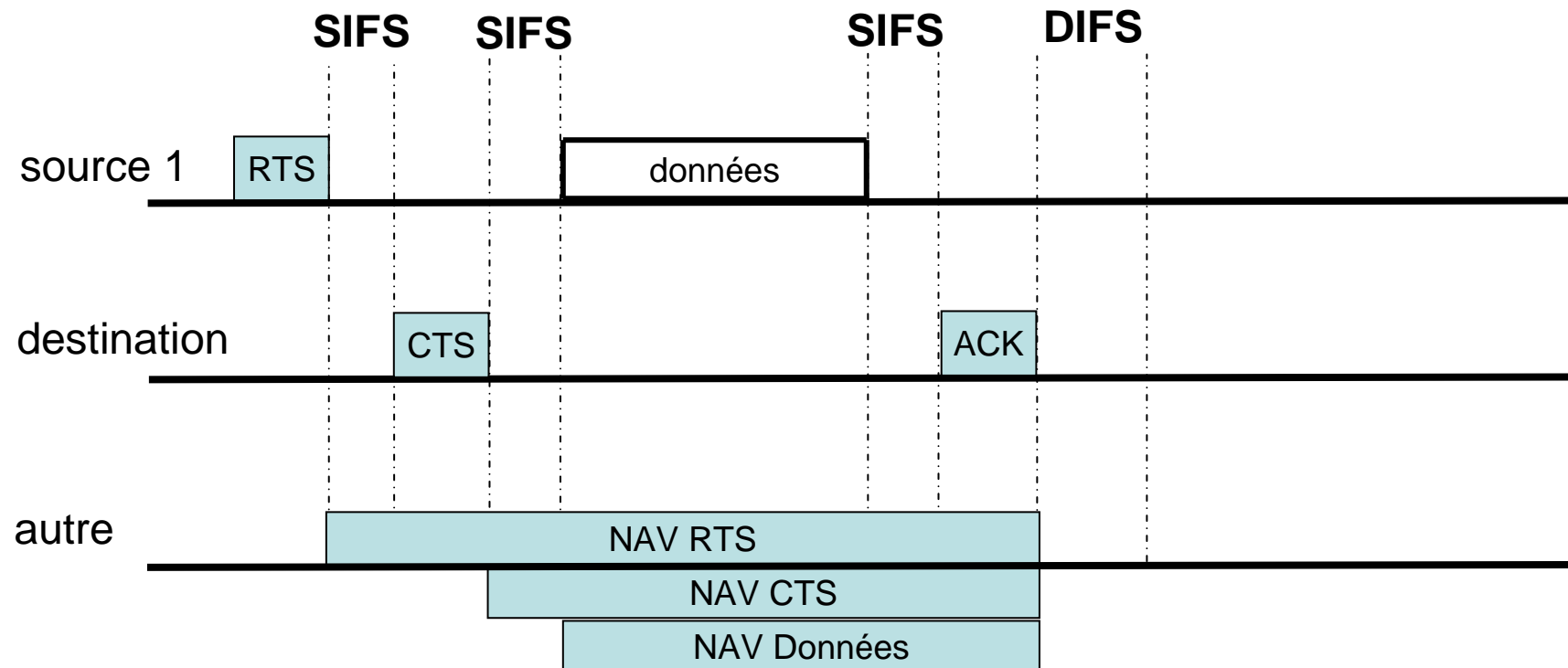
Les paquets RTS, CTS et Données contiennent des informations sur la durée des émissions.

Ce mécanisme permet à une station de connaître l'état du canal (occupé) même si elle ne reçoit pas de données (le canal est perçu comme étant libre).

Chaque station maintient cette information dans une structure appelée *Network Allocation Vector (NAV)*.

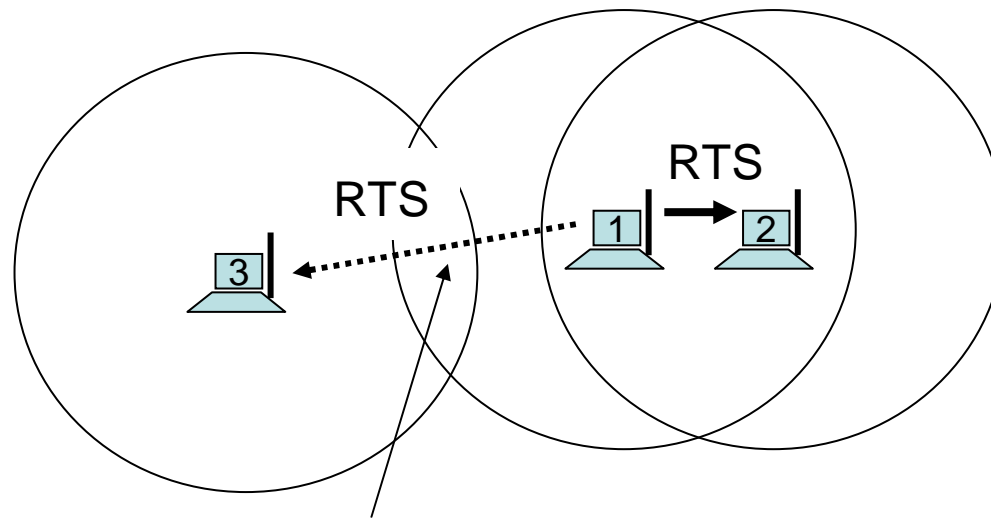
Ce mécanisme est un mécanisme de **détection de porteuse virtuelle** (virtual carrier detection).

RTS/CTS - NAV



RTS/CTS - EIFS

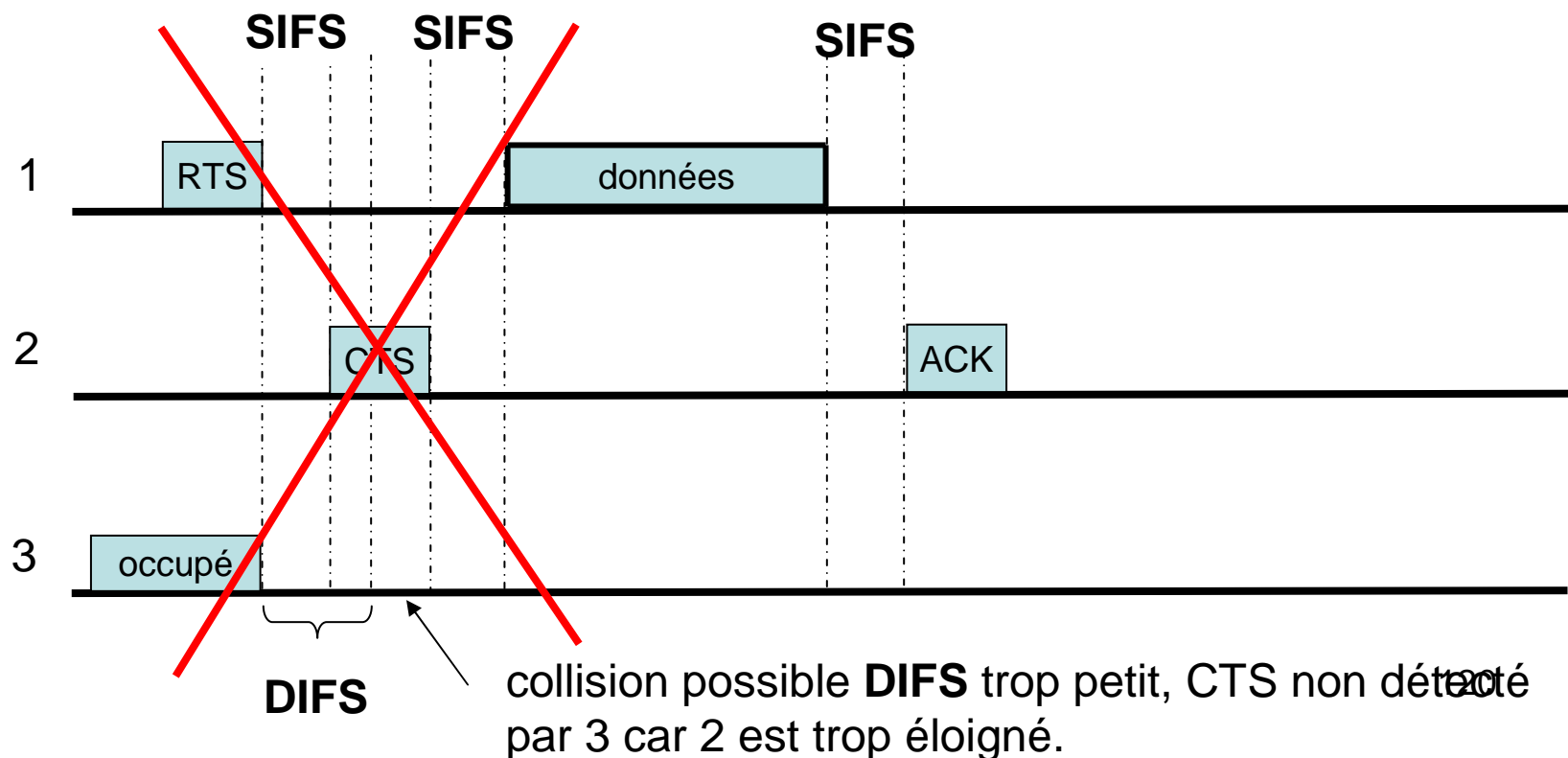
Dans certaines configurations, une station perçoit que le canal est occupé sans pouvoir décoder les informations.



signal très faible, reçu mais non décodé
le CTS émis par 2 ne sera pas détecté

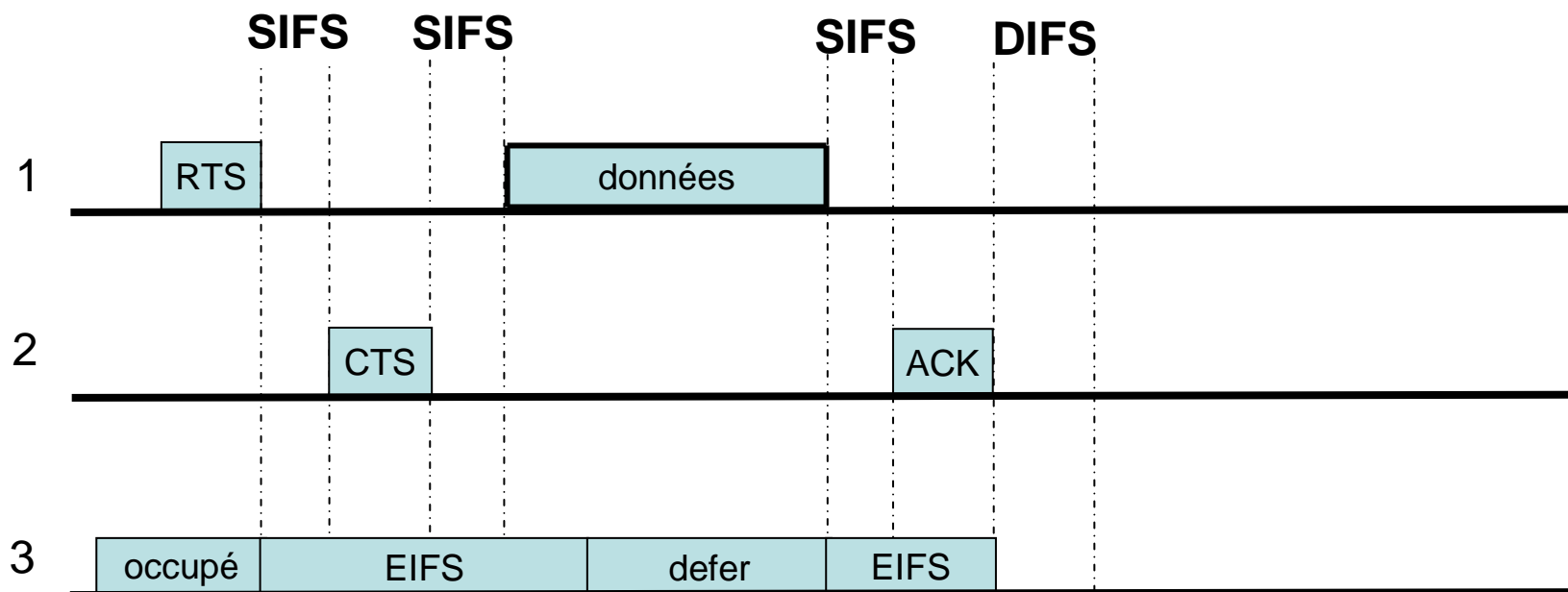
RTS/CTS - EIFS

Dans cette situation la station 3 perçoit le canal comme occupé et devrait attendre une période de temps égale à DIFS avant d'essayer d'accéder au canal. Le problème est que $DIFS < SIFS + CTS$, et la station 3 peut donc perturber la réception du signal CTS.

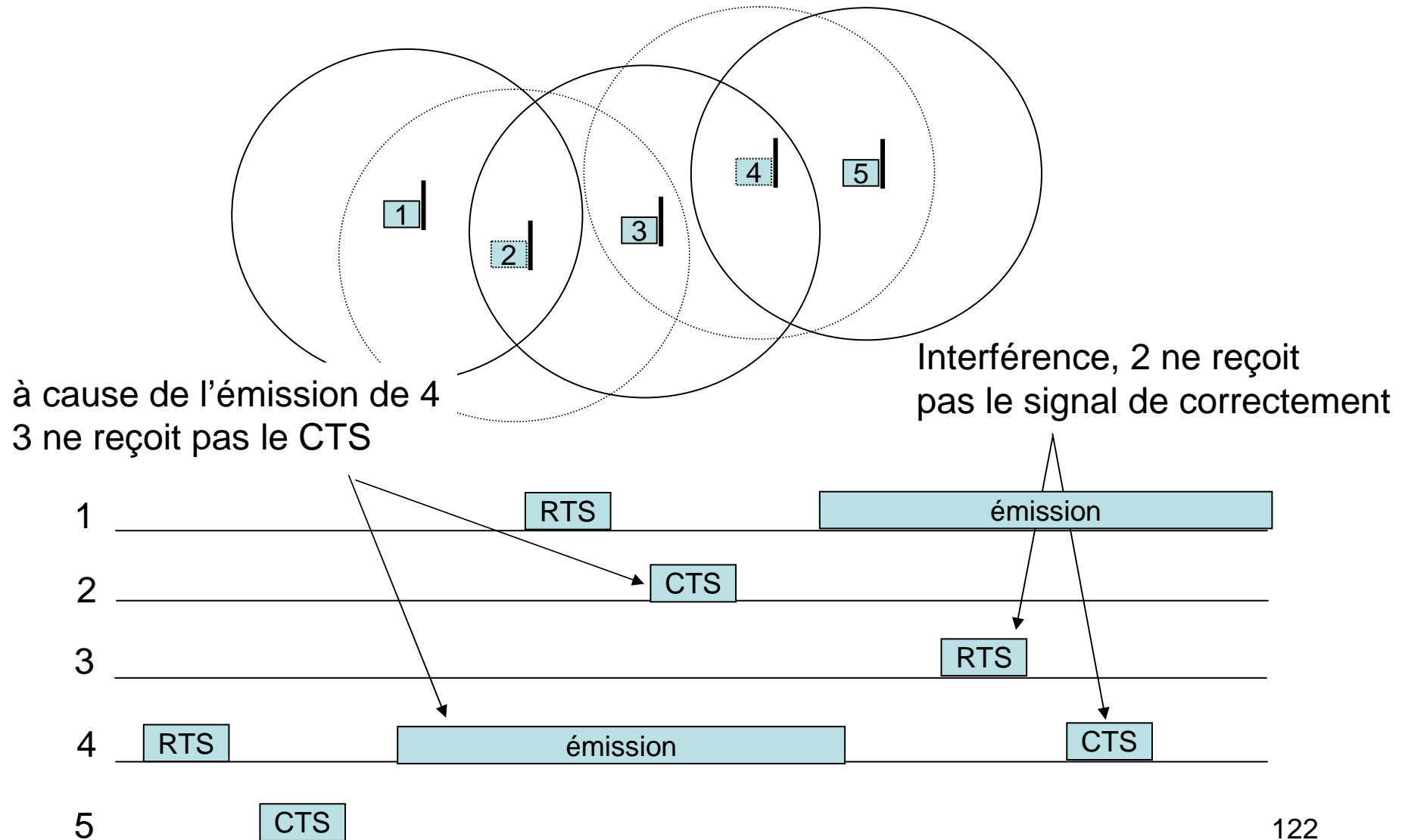


RTS/CTS - EIFS

Dans la situation où une station perçoit le canal occupé sans pouvoir décoder les informations, la norme 802.11 prévoit que la station 3 attende une durée **EIFS** (Extended Inter Frame Spacing) plus grande que SIFS + CTS



Scenario ou RTS/CTS est défaillant



Fragmentation

Dans la situation où les données à transmettre sont trop volumineuses la couche MAC fragmente les données en unités plus petites pour augmenter la fiabilité du transfert.

Dans cette situation tous les fragments sont transmis séquentiellement sans que le canal soit attribué à une autre station.

Chaque fragment est acquitté (positif) par la station réceptrice.

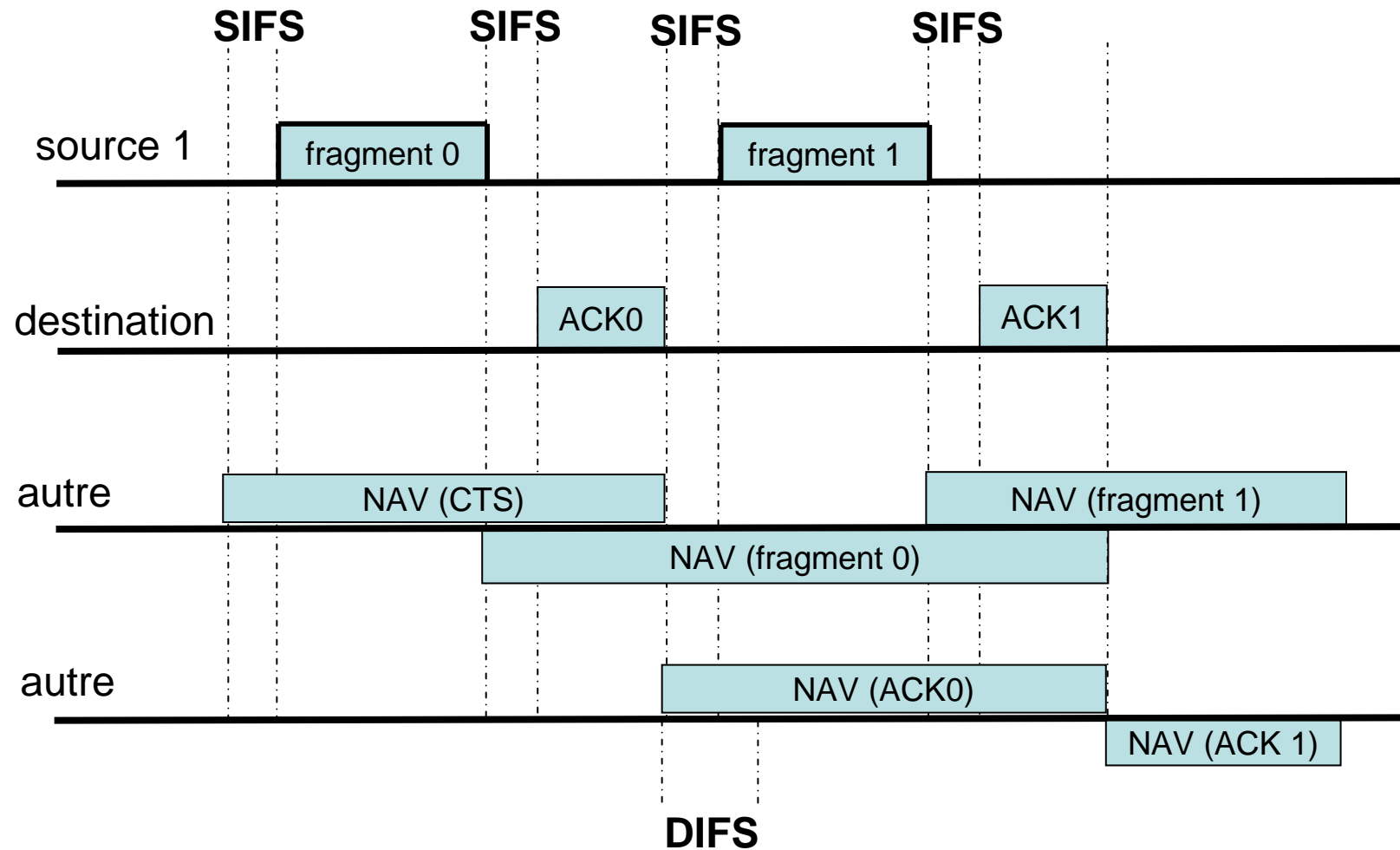
La station émettrice garde le contrôle du canal en attendant une période de temps SIFS seulement entre l'émission des trames (les autres stations attendent $DIFS > SIFS$)

Fragmentation

Si un acquittement est perdu, la station émettrice stop le transfert, ré exécute un RTS/CTS pour accéder le canal et retransmet les fragments en commençant par le paquet qui n'a pas été acquitté.

Pendant la période handshake (RTS/CTS) les informations pour la porteuse virtuelle concernent seulement le premier fragment.

Fragmentation



Remarques

Dans la littérature le mécanisme 'écouter avant d'émettre' s'appelle **CSMA/CA** pour Carrier Sense Multiple Access/Collision Avoidance (par exemple, réseaux Ethernet)

MACA (Multiple Acces with Collision Avoidance) désigne le protocole CSMA/CA avec le mécanisme RTS/CTS.

MACAW améliore le protocole en ajoutant des acquittements ainsi qu'un tirage aléatoire pour chaque connexion (initialement le temps d'attente aléatoire est valable pour toutes les transmissions de la station)

Mode PCF centralisé

Les accès au canal par les mécanismes présentés utilisent toujours un temps d'attente aléatoire pour éviter les collisions. Ce temps d'attente n'est pas toujours adéquat:

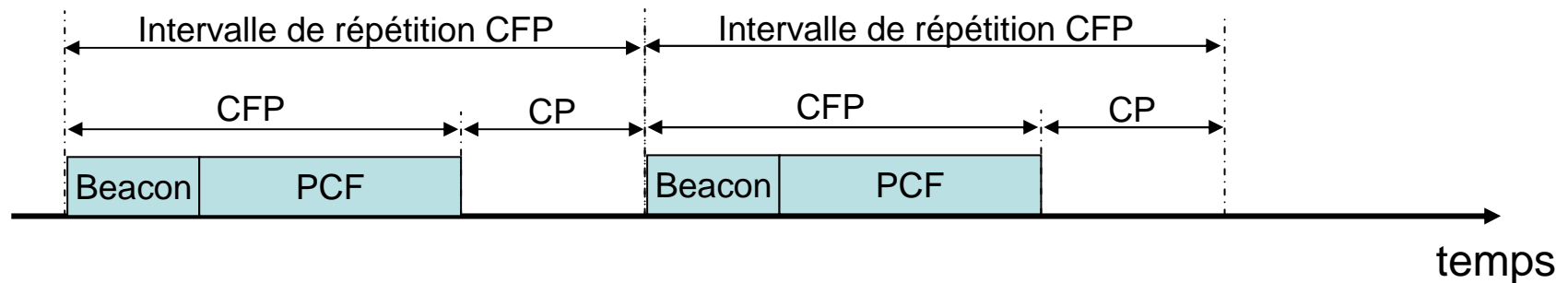
- Le temps d'attente diminue le débit des transferts.
- Pas de mécanismes temps-réel possible.
- Il est difficile d'assurer le transfert synchrones des données, par exemple pour transmettre du son ou des images.

Ces accès au canal sont distribués, il n'y a pas de station qui centralisent la gestion du canal. C'est le mode **DCF** (Distributed Coordination Function)

La norme 802.11 prévoit la possibilité de faire transiter du trafic synchrone, c'est le mode **PCF** (Point Coordination Function)

Mode PCF centralisé

Le mode **PCF** coexiste avec le mode **DCF**. En fait, toutes les stations doivent impérativement implémenter le mode DCF, le mode PCF n'est pas obligatoire.



En mode PCF, toutes les transmissions sont initiées par le point d'accès (AP).

La trame de gestion BEACON signifie le début du mode PCF, permet la synchronisation des stations et fixe le temps.

L'intervalle de répétition est un paramètre du protocole et doit être un multiple du temps de transfert de la trame BEACON

Mode PCF centralisé

La durée de la période CFP est aussi un paramètre du protocole, mais elle doit permettre de transmettre au minimum 2 MPDU (Mac Protocol Data Unit).

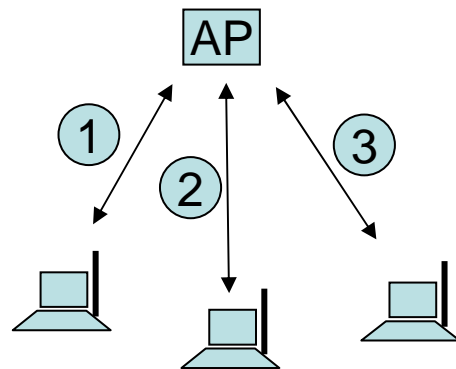
L'intervalle de répétition CFP doit permettre de transmettre au moins une MPDU.

Dans certaines limites le point d'accès peut faire varier ces paramètres en fonctions de la charge du réseau.

Le protocole d'accès au canal en mode PCF est un protocole de partage de ressources qui utilise le pooling par une station maître.

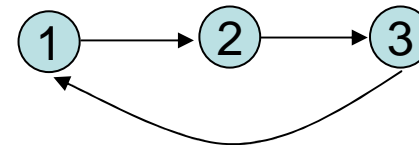
Mode PCF centralisé

Le point d'accès (AP) maintiens une '*pooling*' liste et tient le rôle de coordinateur (PC, Point Coordinator). Les stations qui appartiennent à cette *pooling* liste sont sollicitées par le point d'accès pour émettre des données pendant la période CFP.



Le schéma d'ordonnancement appliqué par le AP n'est pas défini dans la norme.

Par exemple:



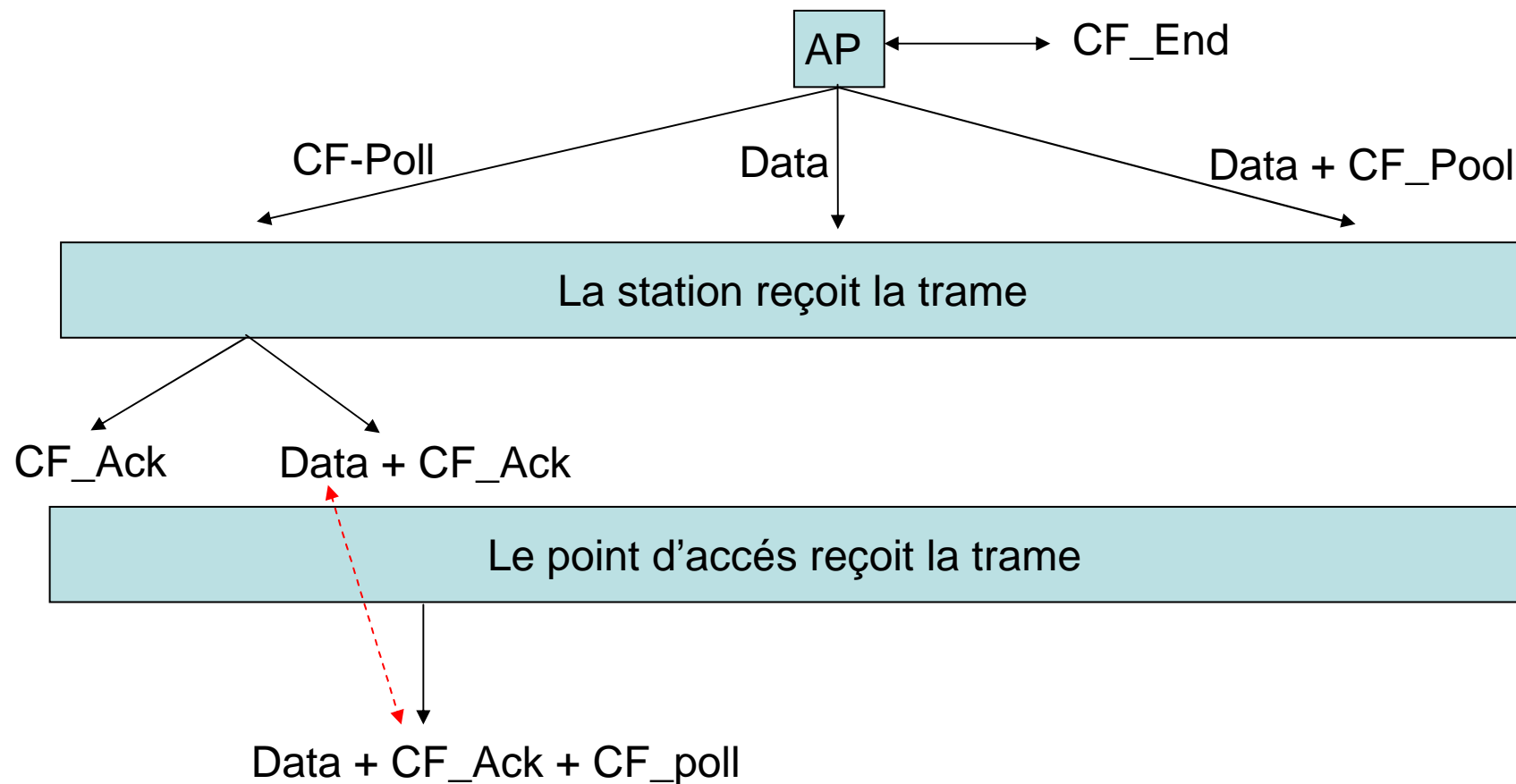
Mode PCF centralisé

Au début de la période CFP, le point d'accès émet une trame BEACON pour signifier le début du mode PCF (après avoir attendu un temps PIFS = Point-Inter Frame Space, $SIFS < PIFS < DIFS$).

La trame BEACON contient les informations sur la durée du cycle PCF et mettent-à-jour leur registre NAV en conséquence (pour ne pas transmettre durant cette période).

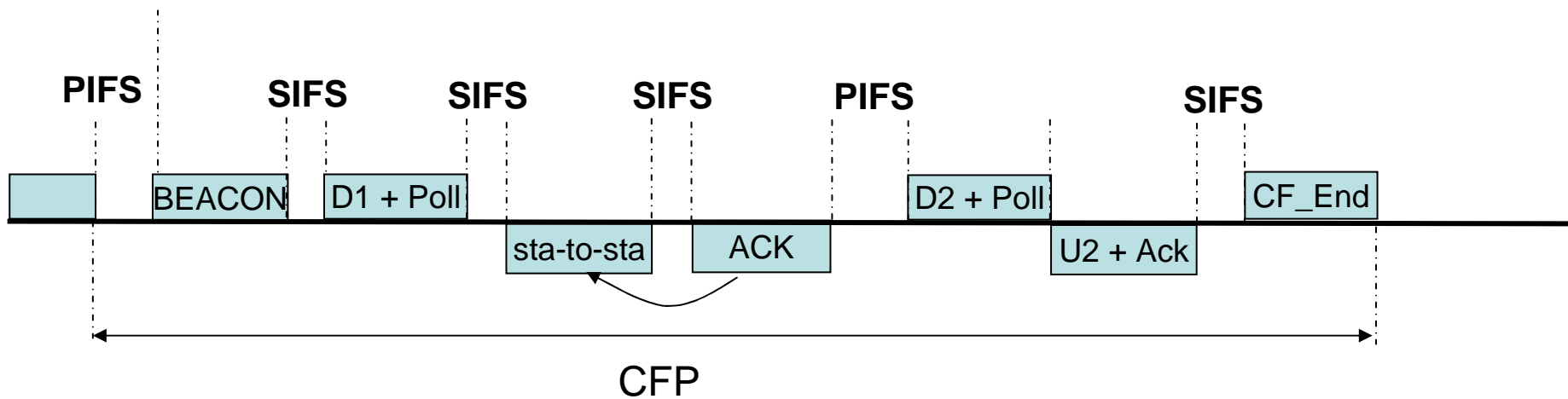
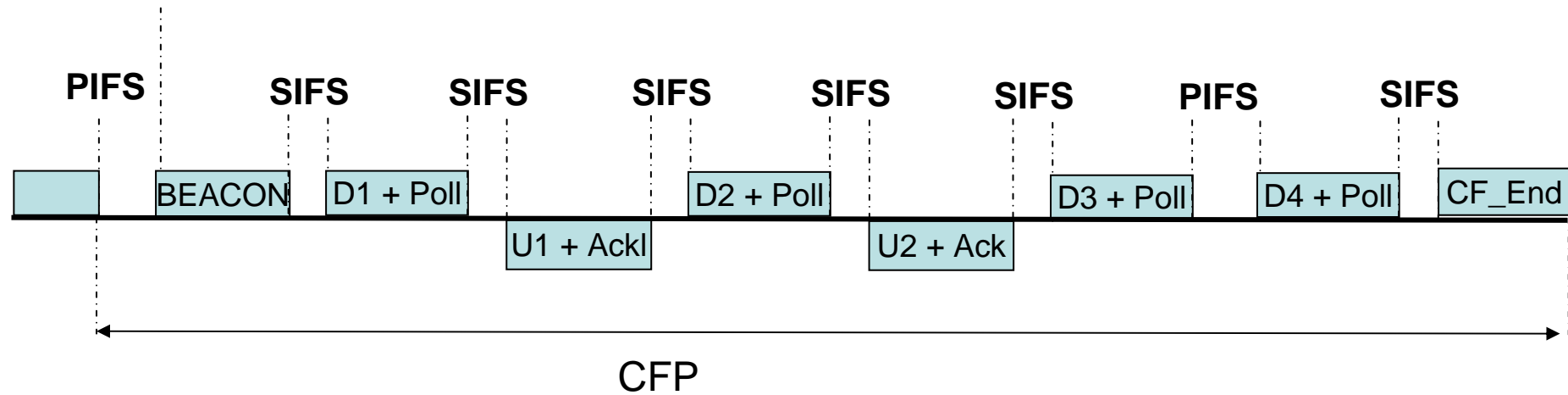
Durant cette période, les seules stations autorisées à émettre sont en réponse à une sollicitation (poll) du point d'accès ou pour émettre un acquittement après avoir reçu un MPDU (après avoir attendu SIFS)

Mode PCF



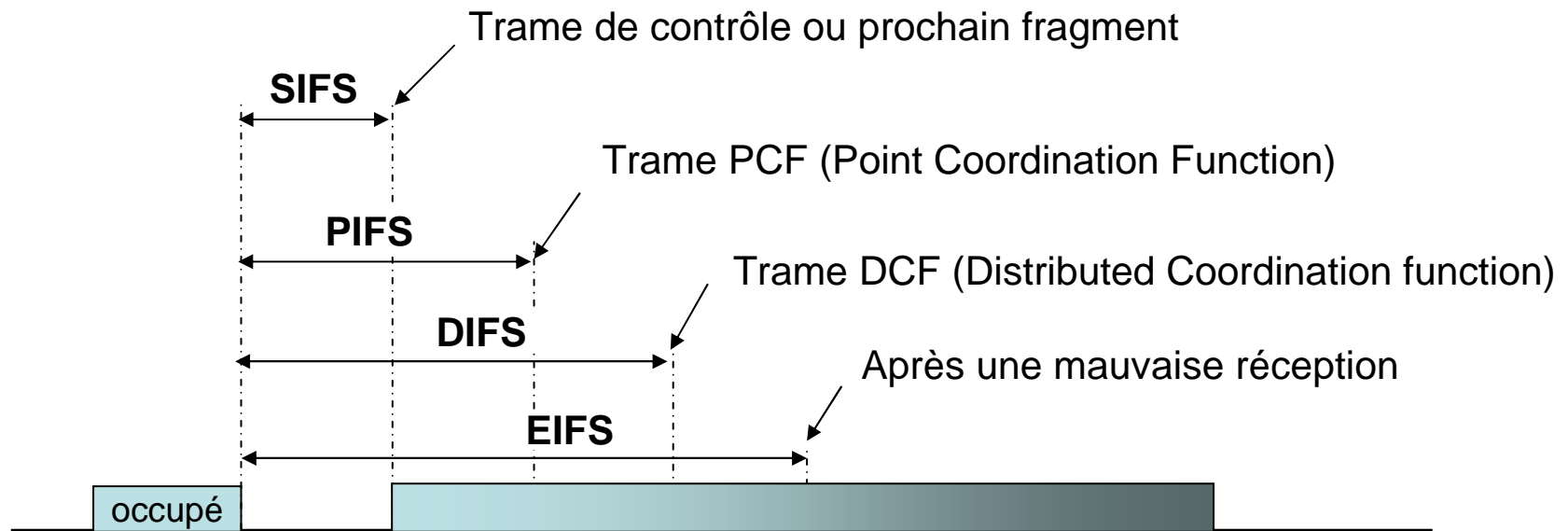
Le point d'accès a la possibilité de combiner dans une même trame envoi de données + poll d'une nouvelle station + Acquiescement d'une donnée reçue

Mode PCF



Les différents temps d'attente

Priorités



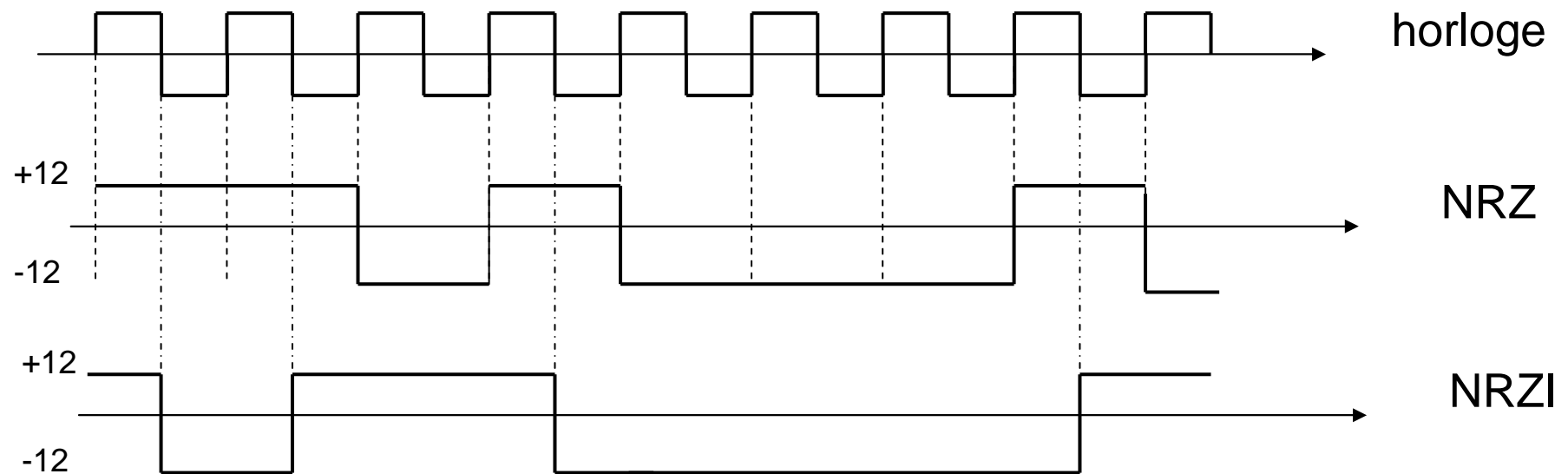
Modulation des signaux

Codage convolutif

Décodage de Viterbi

Transmission en Bande de base

En bande de base les signaux sont transmis 'tels quels'



Ethernet 10base-5, 10base-2, 10base-T, usb, ... utilisent la transmission en bande base.

Modulation

Pour transmettre un signal $s(t)$ en RF il faut moduler le signal. Il doit être de la forme:

$$s(t) = A \sin(\omega t + \varphi)$$

Pour transmettre de l'information, on peut modifier

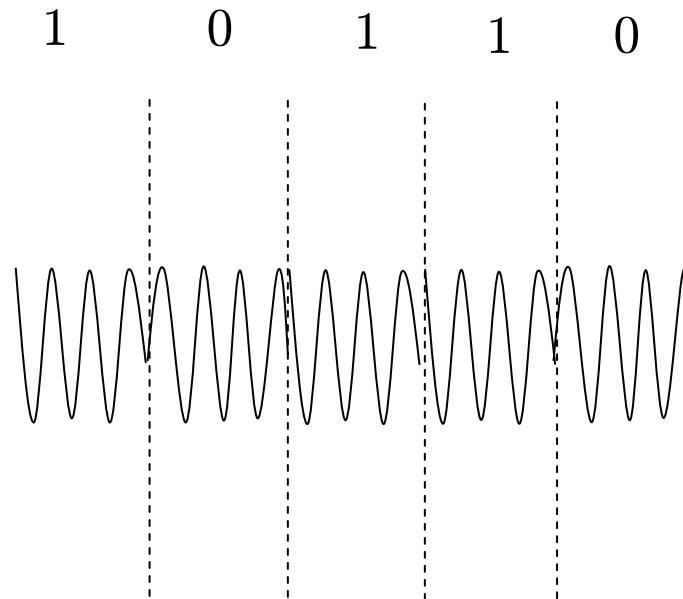
- L'amplitude
- La pulsation (fréquence)
- La phase

Il est très difficile de générer des signaux en changeant précisément la fréquence. Ce qu'on sait faire c'est générer un signal avec une fréquence précise mais constante.

Modulation phase

Phase-Shift Keying (PSK)

$$s_1(t) = \cos(\omega t) \quad s_2(t) = \sin(\omega t)$$



Modulation de phase

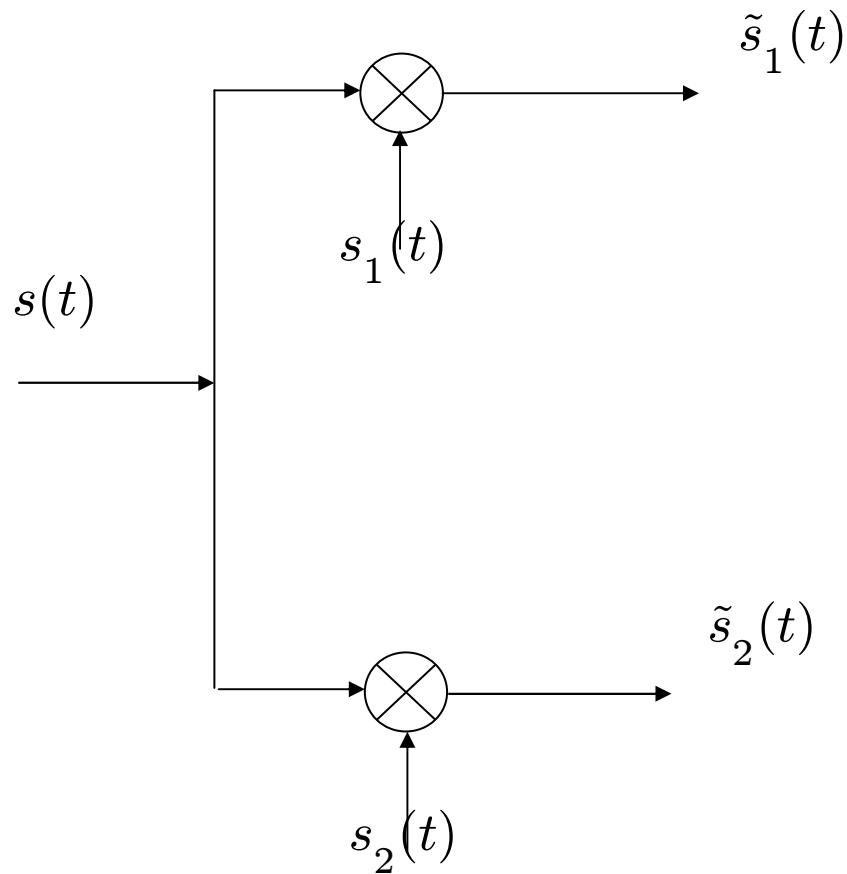
PSK

En modulation de phase, le signal reçu est comparé à un signal de référence, selon la phase observée du signal reçu le décodeur décide si un 1 ou un 0 à été transmit.

Pendant un intervalle de temps un seul bit est transmit.

Pour obtenir l'information concernant la phase, le décodeur multiplie le signal reçu par les signaux $s_1(t)$ et $s_2(t)$ et observe la phase en considérant la composante continue du signal obtenu.

Modulation de phase



Modulation de phase

1^{er} cas $s(t) = s_1(t)$:

$$\tilde{s}_1(t) = s(t)s_1(t) = s_1(t)s_1(t) = \cos(\omega t) \cos(\omega t)$$

$$= \frac{e^{-i\omega t} + e^{i\omega t}}{2} \frac{e^{-i\omega t} + e^{i\omega t}}{2}$$

$$= \frac{1}{2} + \frac{1}{2} \cos(2\omega t)$$

↑
composante continue

↑
composante haute fréquence peut-être supprimée par filtrage

Modulation de phase

2ème cas $s(t) = s_2(t)$:

$$\tilde{s}_1(t) = s(t)s_1(t) = \sin(\omega t) \cos(\omega t)$$

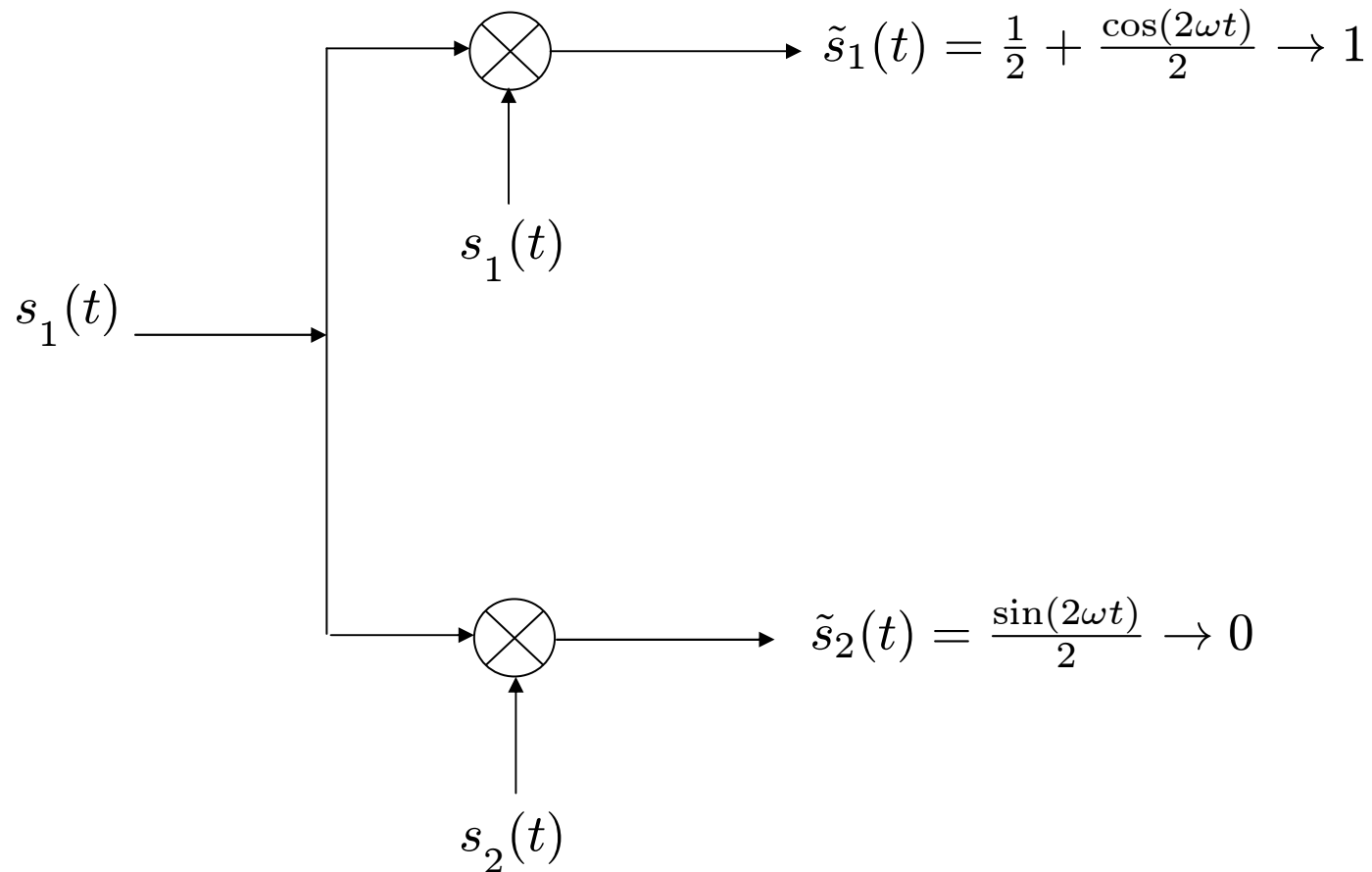
$$= \frac{e^{i\omega t} - e^{-i\omega t}}{2i} \frac{e^{i\omega t} + e^{-i\omega t}}{2}$$

$$= 0 + \frac{1}{2} \sin(2\omega t)$$

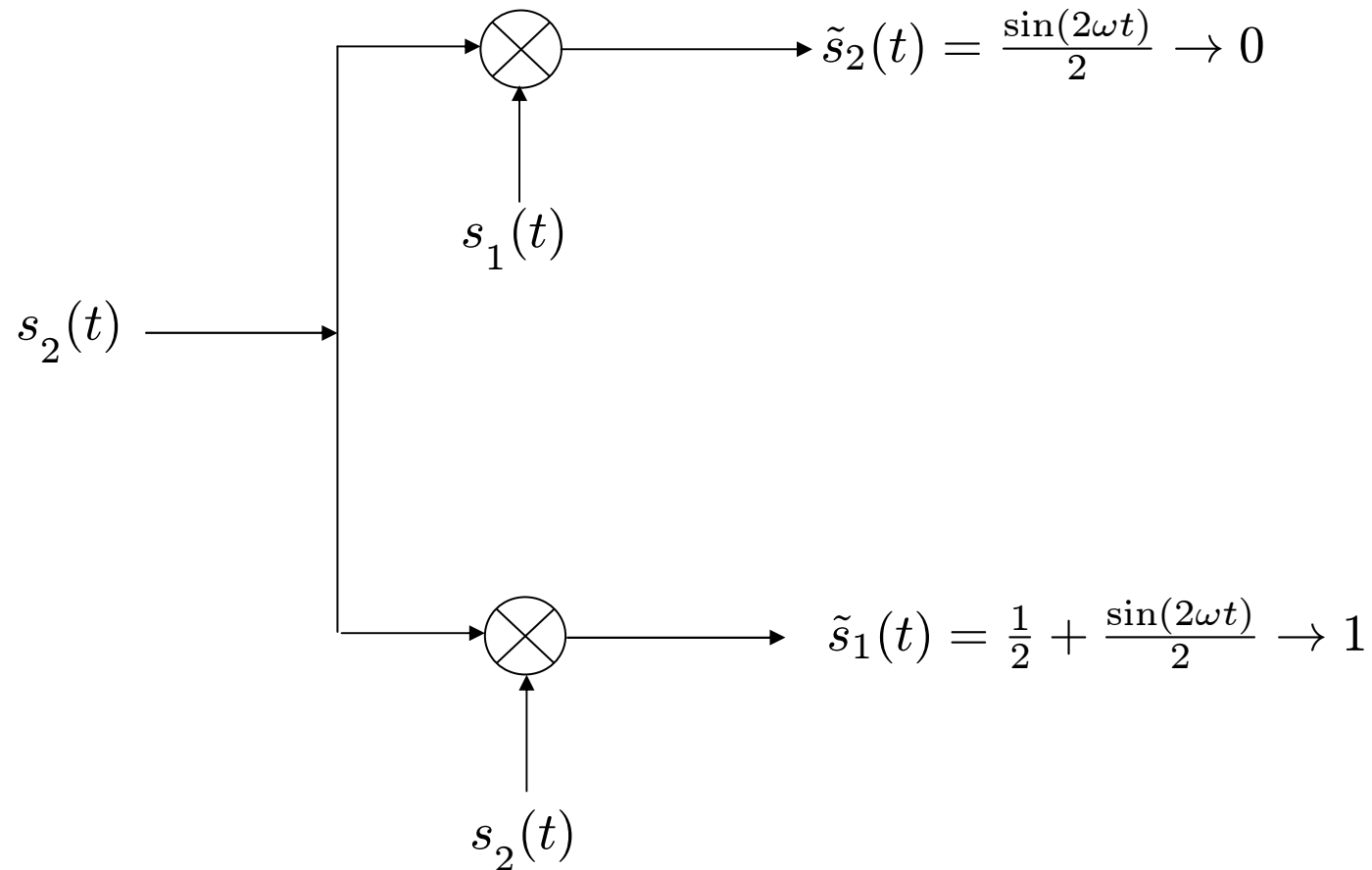
composante continue

composante haute fréquence peut-être supprimée par filtrage

Modulation de phase



Modulation de phase



Modulation de phase

Les signaux obtenus par le décodeur, $s_1(t)$ et $s_2(t)$ sont toujours complémentaires (0 - 1 ou 1 - 0).

On peut généraliser la modulation par phase pour transmettre un nombre complexe de la forme $a(\cos(\varphi) + i \sin(\varphi))$.

En effet si $s(t) = 2a \sin(\omega t + \varphi)$, on obtient:

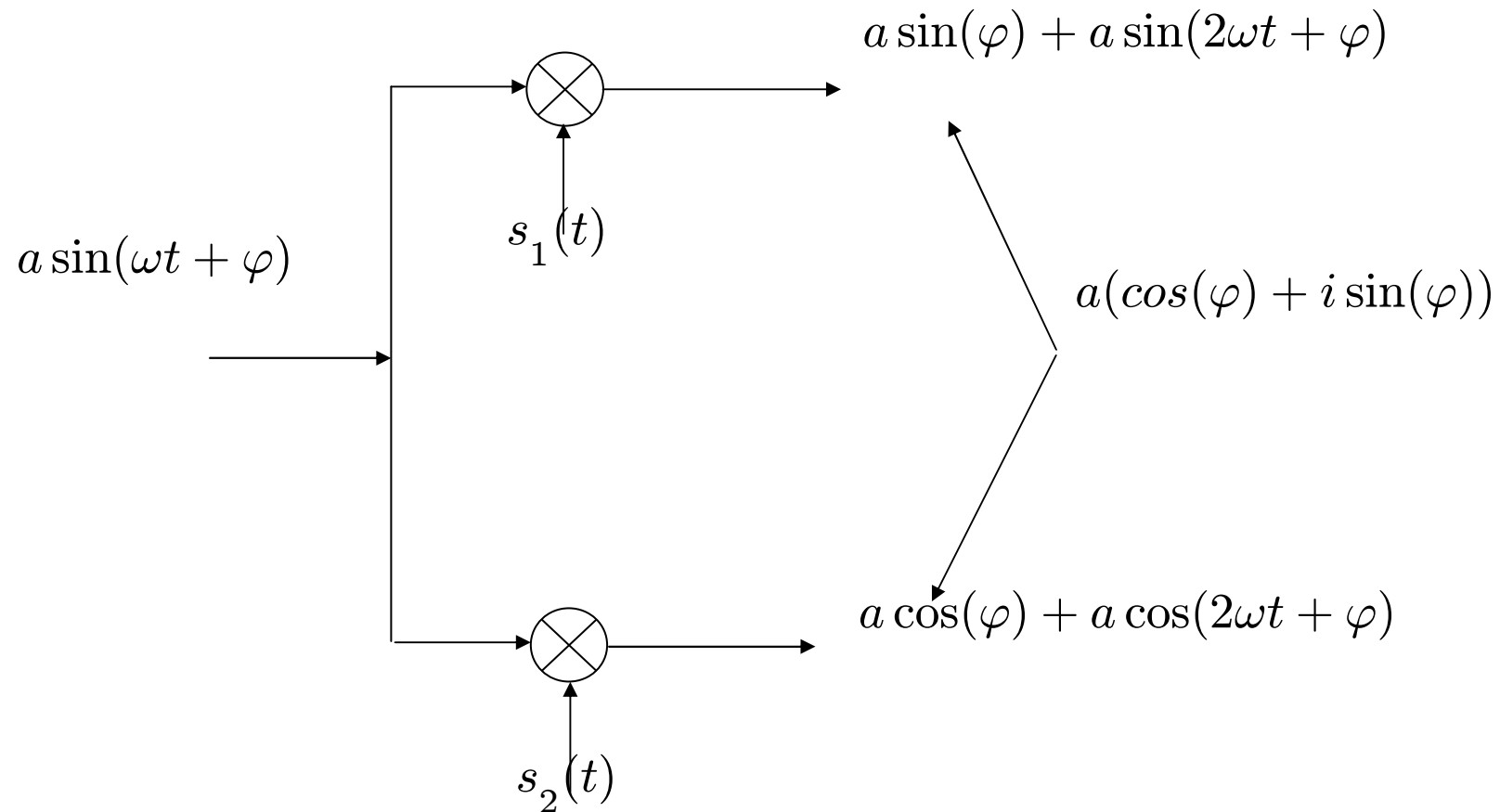
$$\tilde{s}_1(t) = s(t) \cos(\omega t) = \dots = a \sin(\varphi) + a \sin(2\omega t + \varphi)$$

et

$$\tilde{s}_2(t) = s(t) \sin(\omega t) = \dots = a \cos(\varphi) + a \cos(2\omega t + \varphi)$$

Rappel: $\sin(\omega t + \varphi) = \sin(\varphi) \cos(\omega t) + \cos(\varphi) \sin(\omega t)$

Modulation en quadrature de phase QPSK

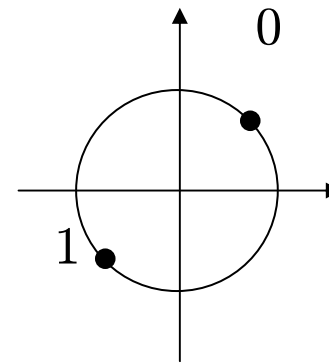


Modulation en quadrature de phase QPSK

On utilise la modulation en quadrature de phase pour transmettre plusieurs bits d'information avec un seul signal.

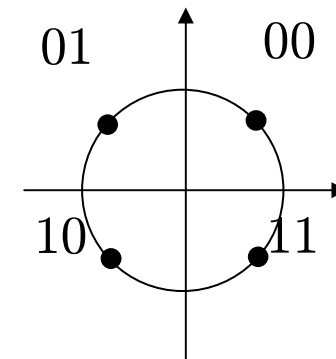
Exemple 1: 802.11 5.5 Mbits/sec (BPSK, Binary Phase Shift Keying)

$$0 \rightarrow 1 + i, 1 \rightarrow -1 - i$$



Exemple 2: 802.11 11 Mbits/sec (QPSK)

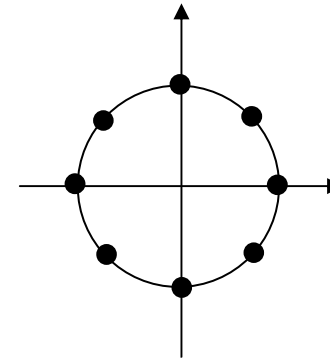
$$00 \rightarrow 1 + i, 01 \rightarrow -1 + i, 10 \rightarrow -1 - i, 11 \rightarrow 1 - i$$



Modulation en quadrature de phase QPSK

Exemple 3: 802.11 22Mbps/sec (8PSK)

code	phase du signal
000	$1+i$
001	$-1+i$
010	$-1-i$
011	$1-i$
100	$\sqrt{2}$
101	$i\sqrt{2}$
110	$-\sqrt{2}$
111	$-i\sqrt{2}$



Codage convolutif

La particularité du codage convolutif est que les mots de codes dépendent de ce qui a été transmis précédemment, ce sont des codes à mémoires.

Soit une séquence de bits à transmettre 0011 0011 1110 101001...

Avec un code par bloc par exemple (pas de mémoire) on divise la séquence en blocs, on associe à chaque bloc un mot de code (codeword) que l'on transmet.

0011 -> 00110011

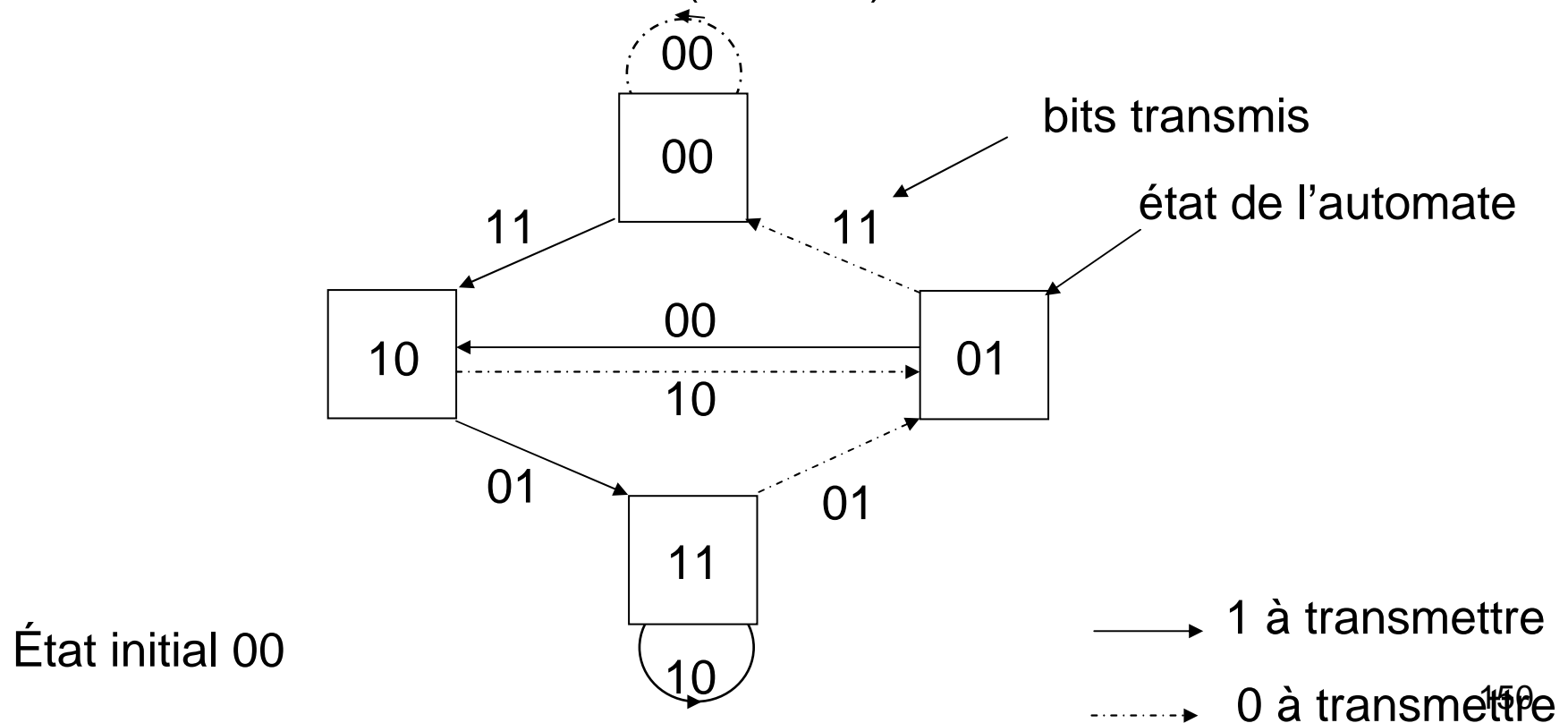
0011 -> 00110011

1110 -> 11101110

Les bits supplémentaires (redondance) permettent de détecter et quelque fois corriger les erreurs de transmission.

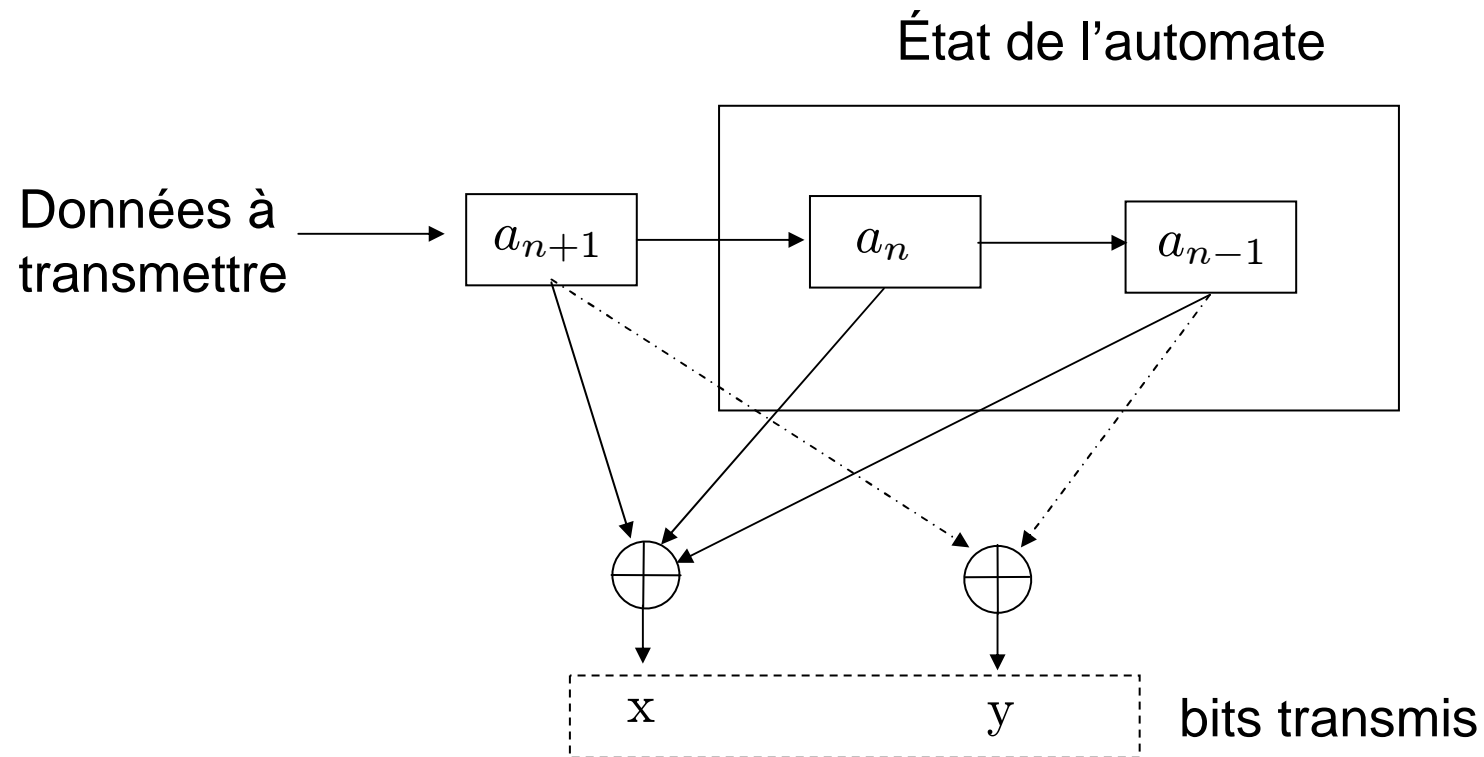
Codage convolutif machine d'état

Pour générer la séquence de bits à transmettre, on utilise un *automate à état finis*. Dans notre exemple, chaque état de l'automate est codé sur 2 bits (=4 états).

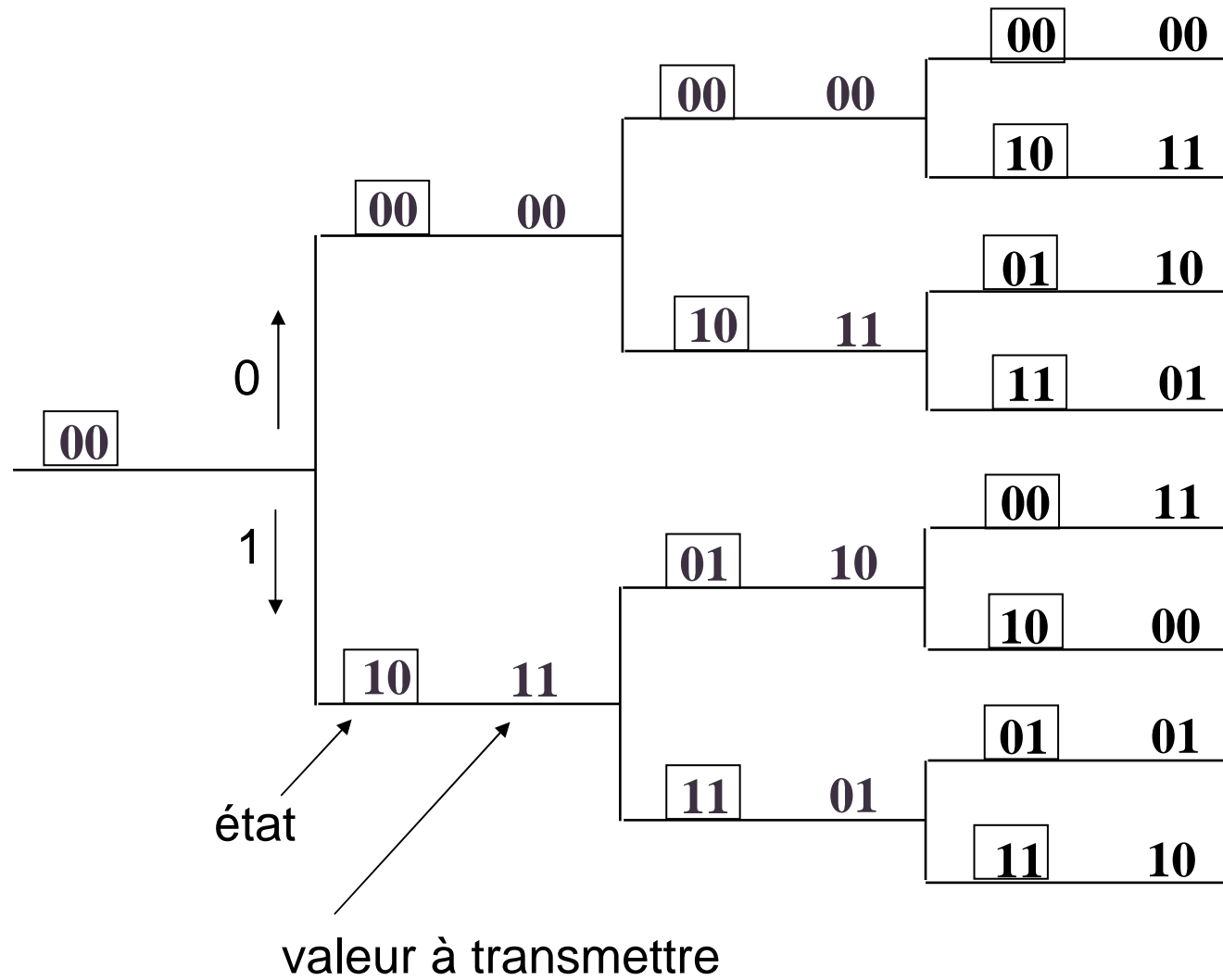


Machine d'état

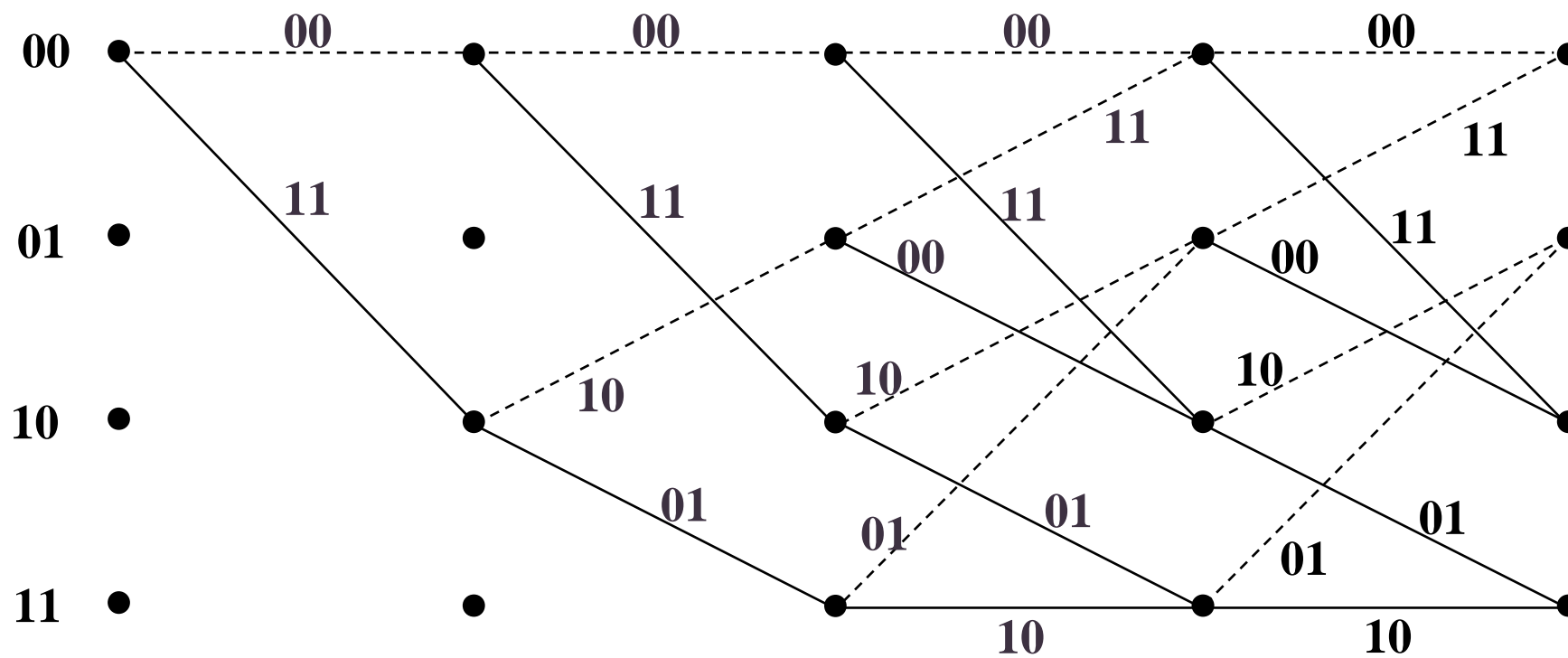
Données: $a_0, a_1, a_2, a_3, \dots$ suite semi-infinie $a_0 = a_1 = 0$



Arbre



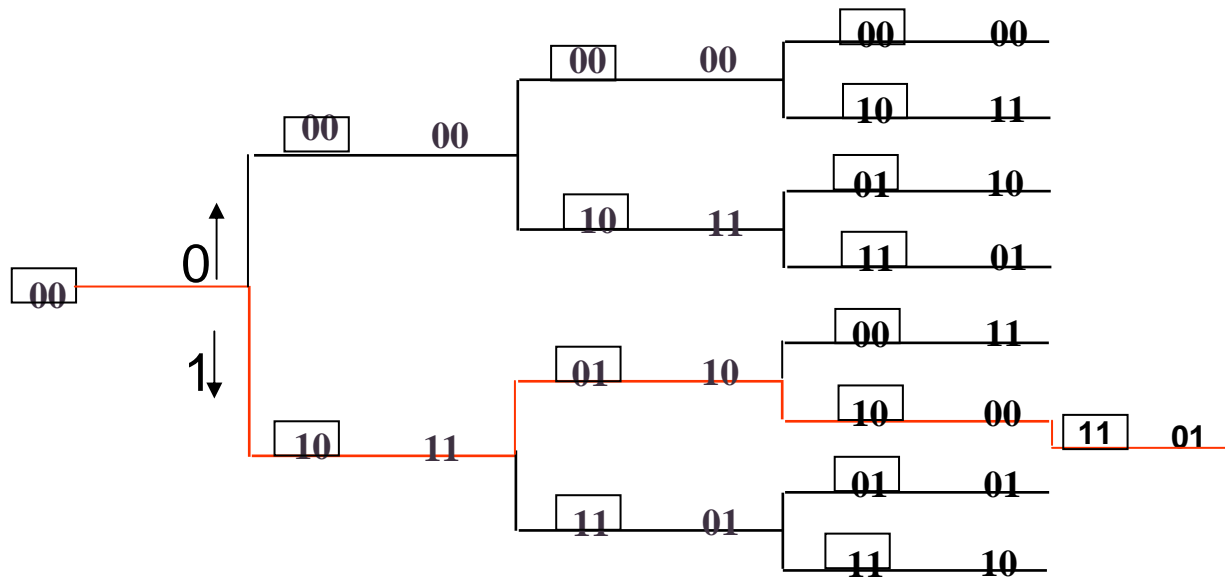
Treillis



Exemple

On désire transmettre la suite de bits **1 0 1 1**

En utilisant le codeur décrit précédemment, avec état initial 00, on obtient la suite **11 10 00 01**



Décodage

Propriété du treillis: chaque chemin dans le treillis correspond à une séquence de bits transmise/reçue et inversement.

Pour le décodage on va chercher le chemin dans le treillis qui est le plus proche de la séquence reçue

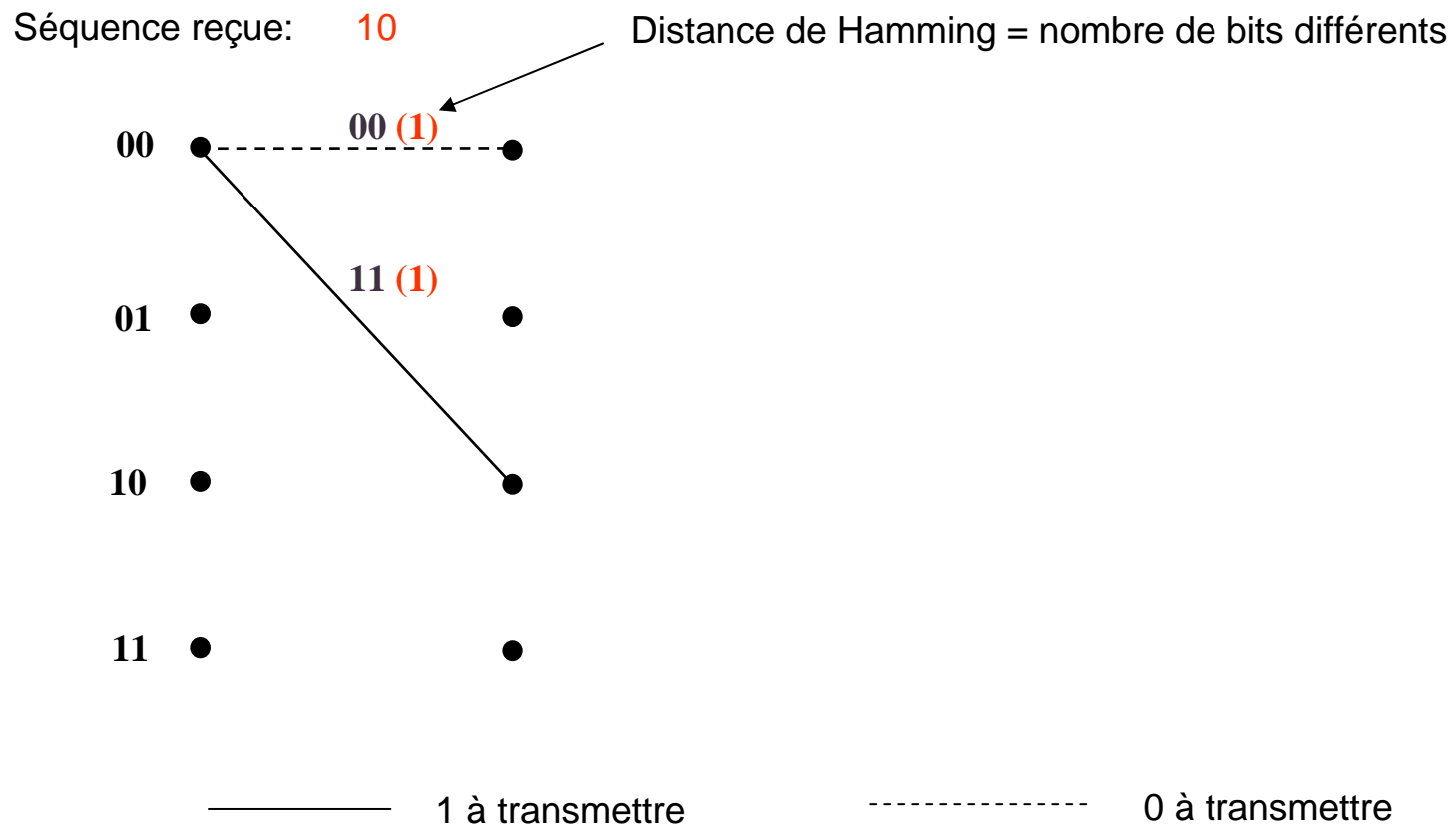
Exemple:

Séquence de bits à transmettre **1 0 1 1**

Séquence transmise **11 10 00 01**

Séquence reçue 1**0** 10 00 01

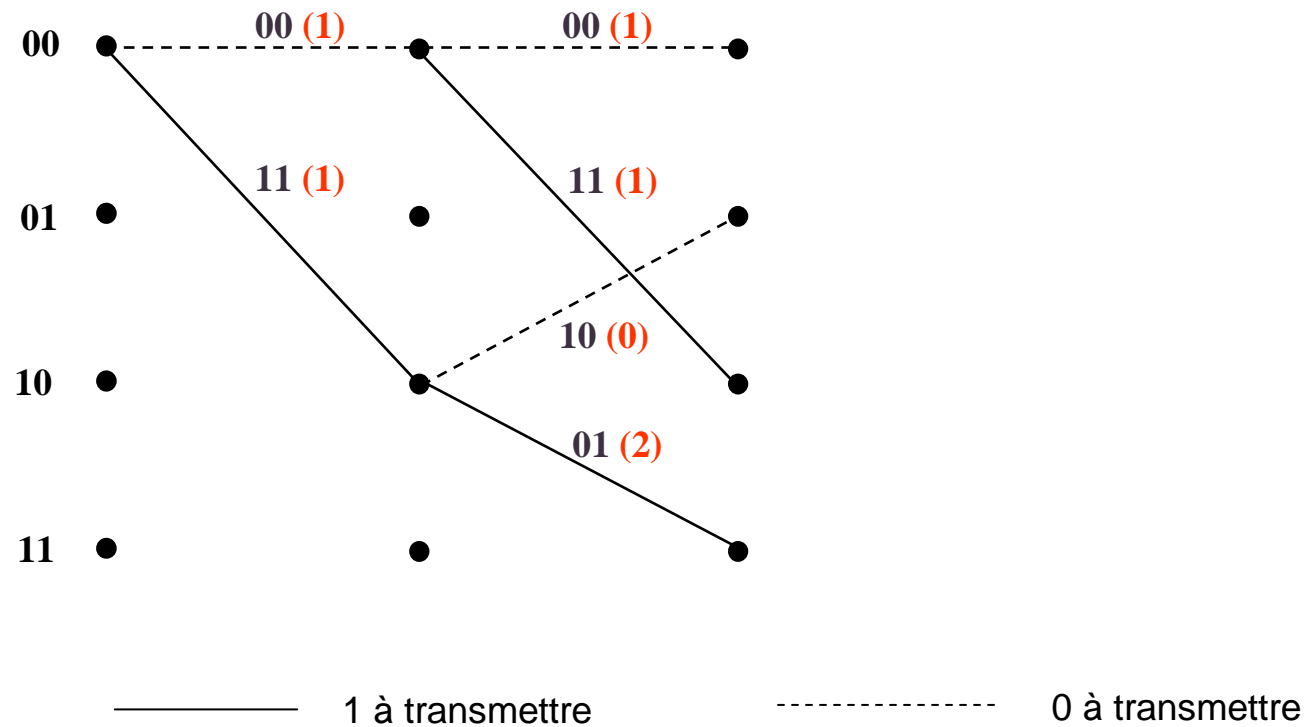
Décodage algorithme de Viterbi



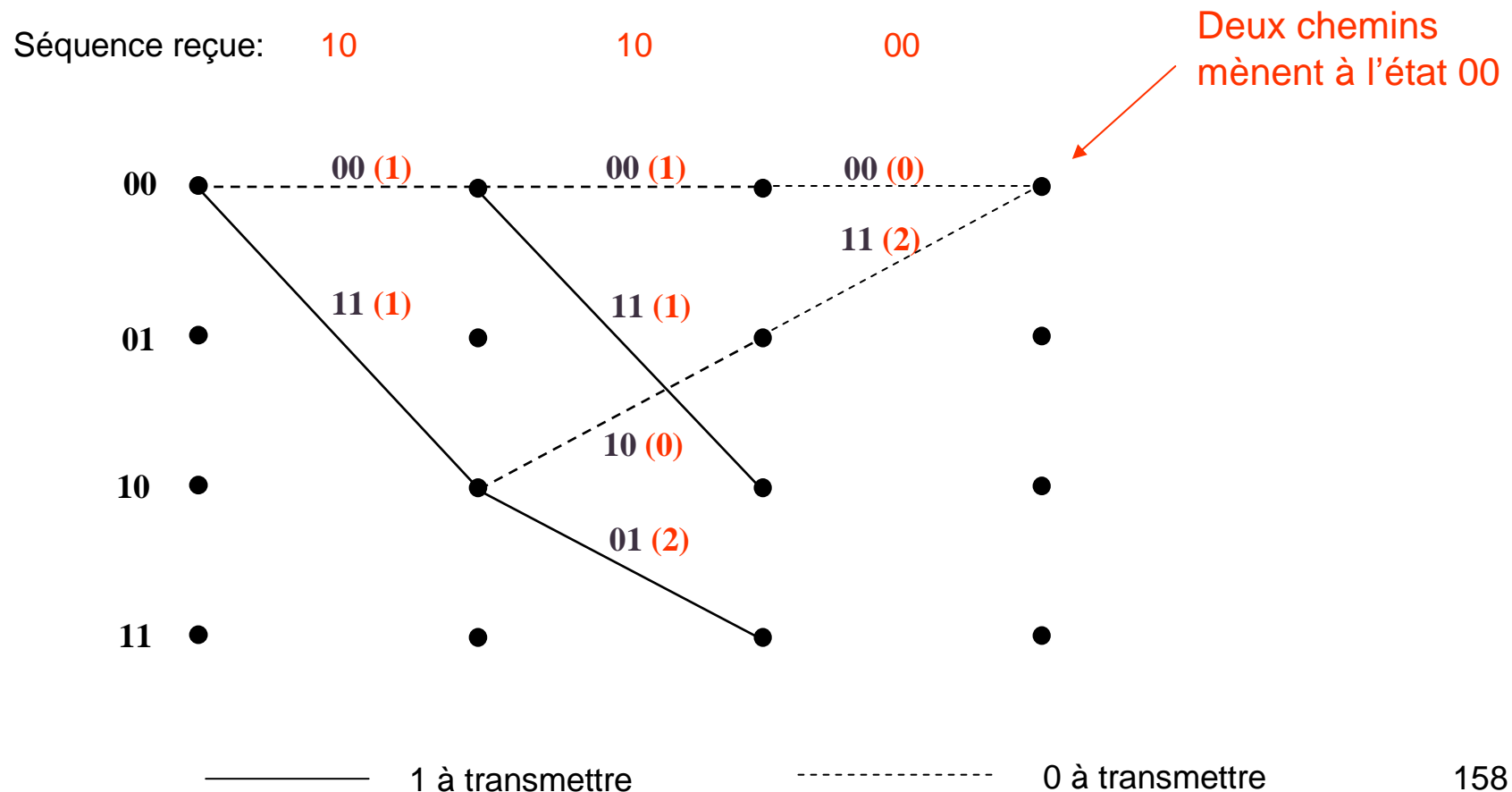
Décodage algorithme de Viterbi

Séquence reçue: 10

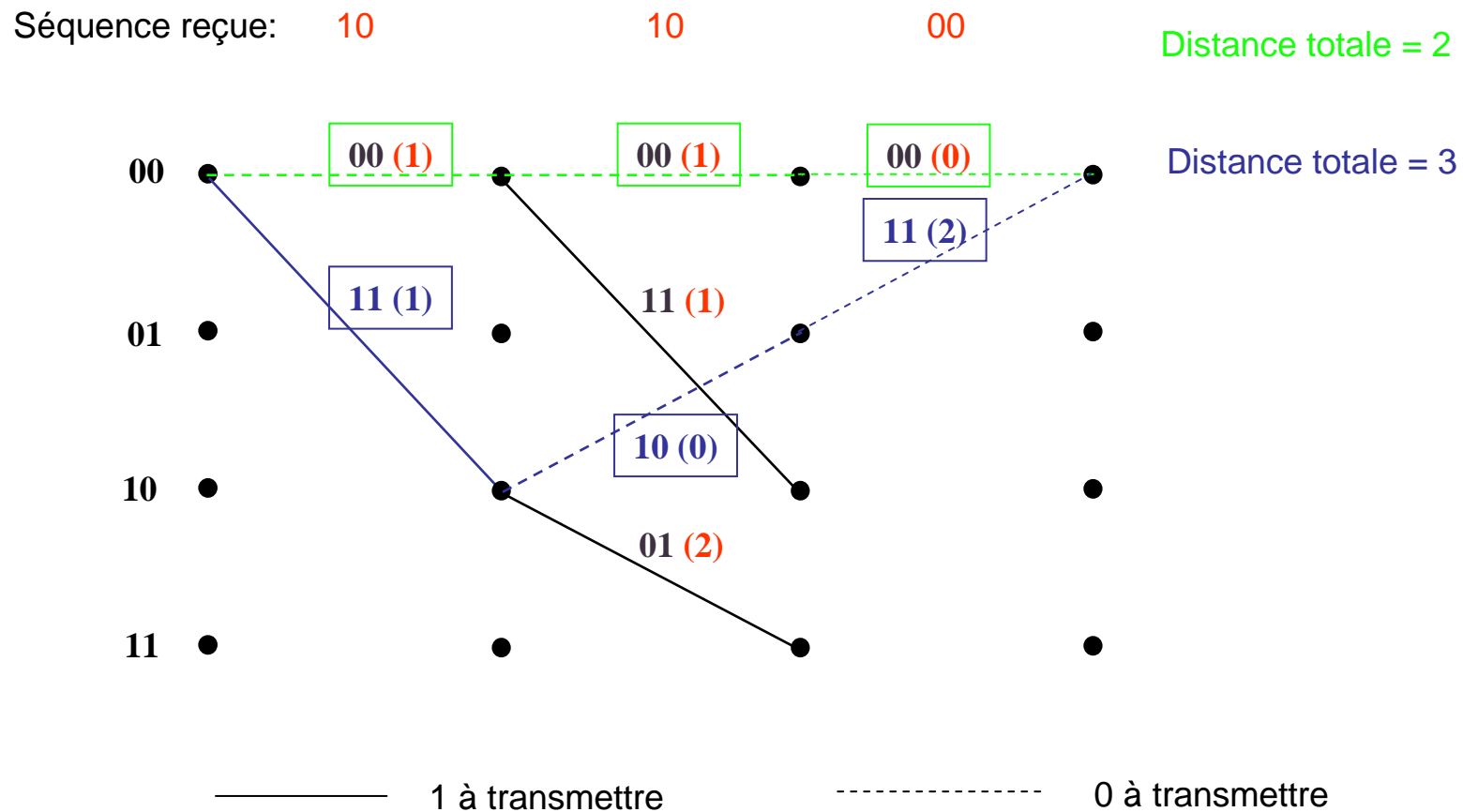
10



Décodage algorithme de Viterbi



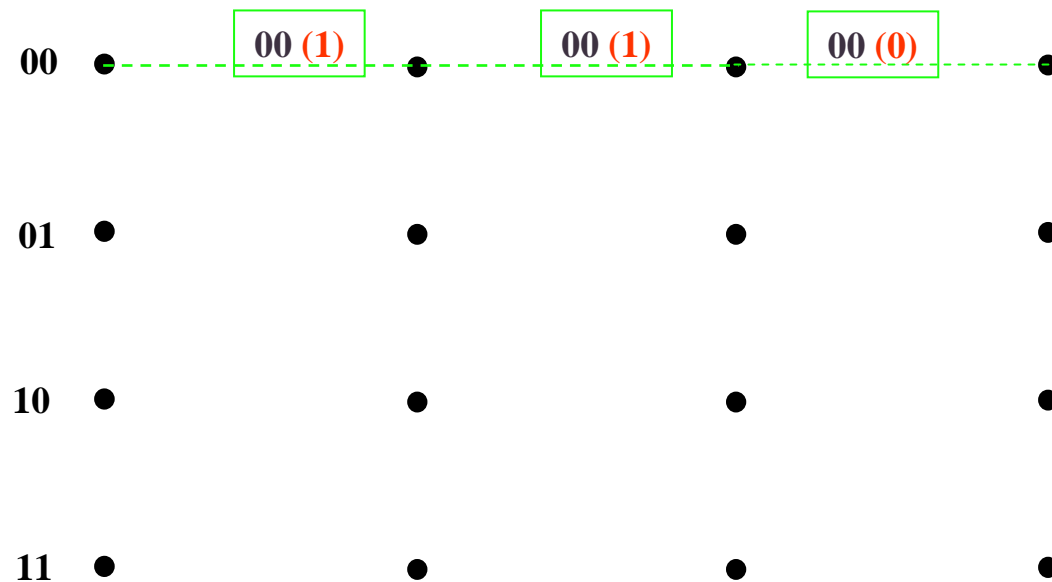
Décodage algorithme de Viterbi



Décodage algorithme de Viterbi

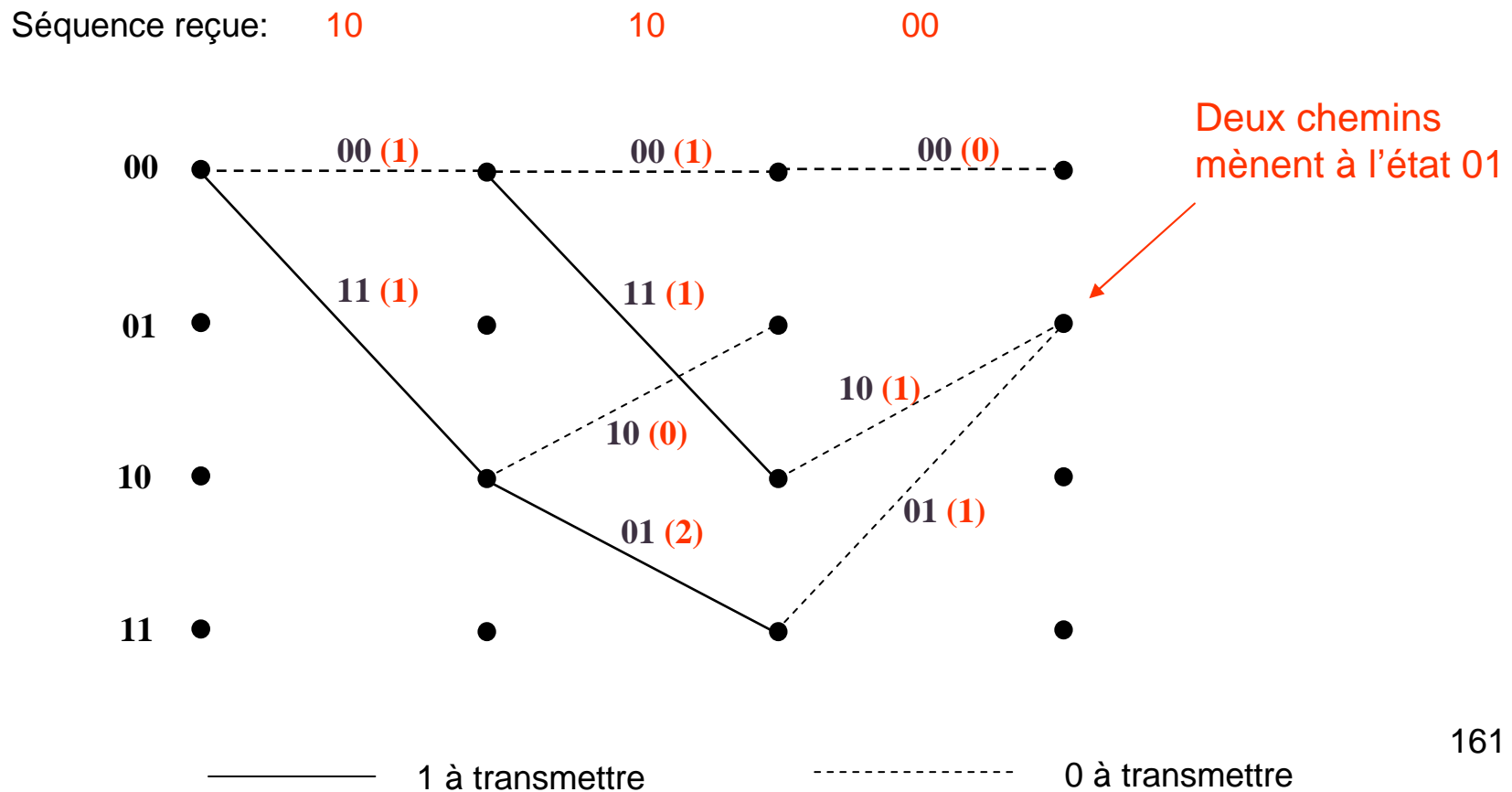
On mémorise le chemin le plus court

Séquence reçue: 10 10 00 Distance totale = 2



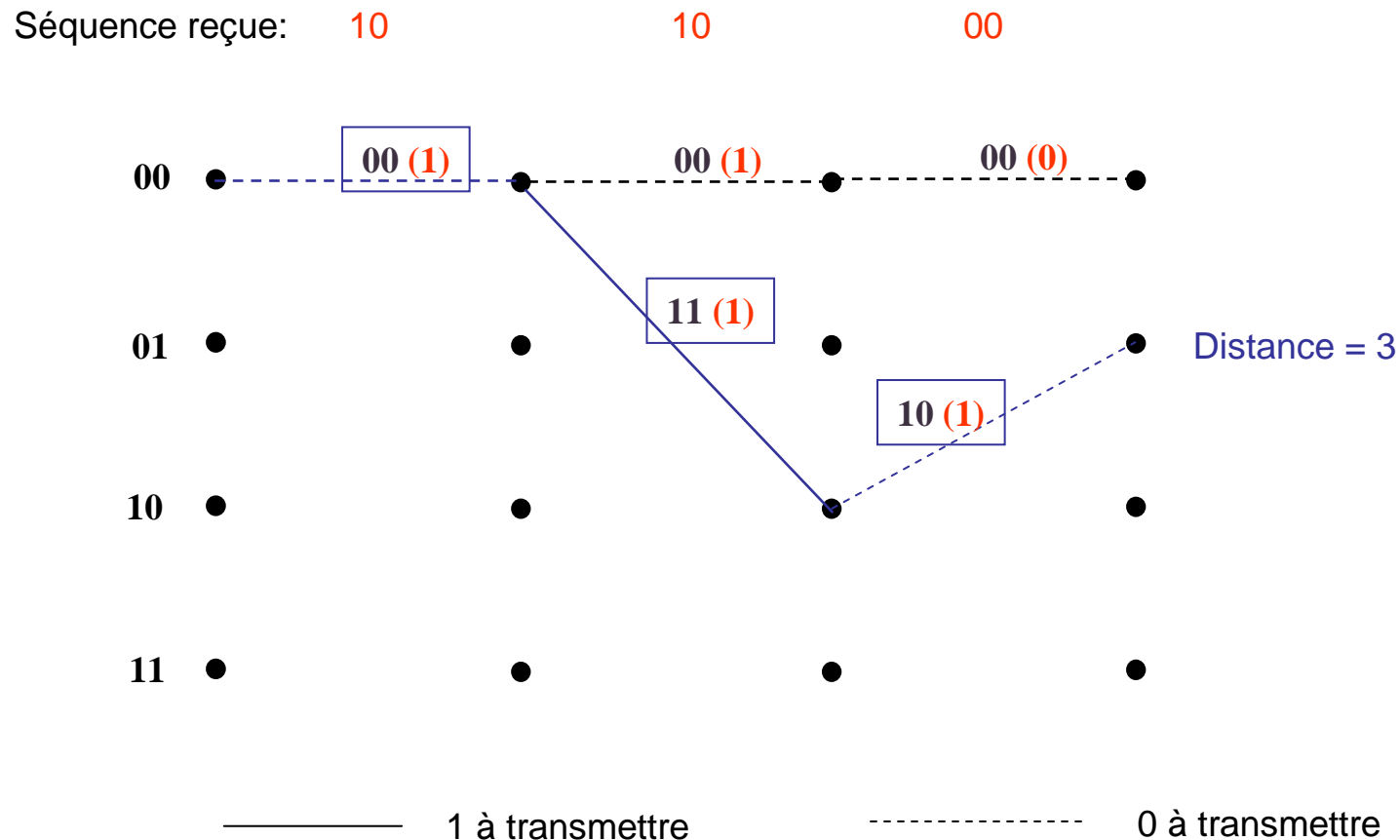
—— 1 à transmettre - - - - - 0 à transmettre

Décodage algorithme de Viterbi



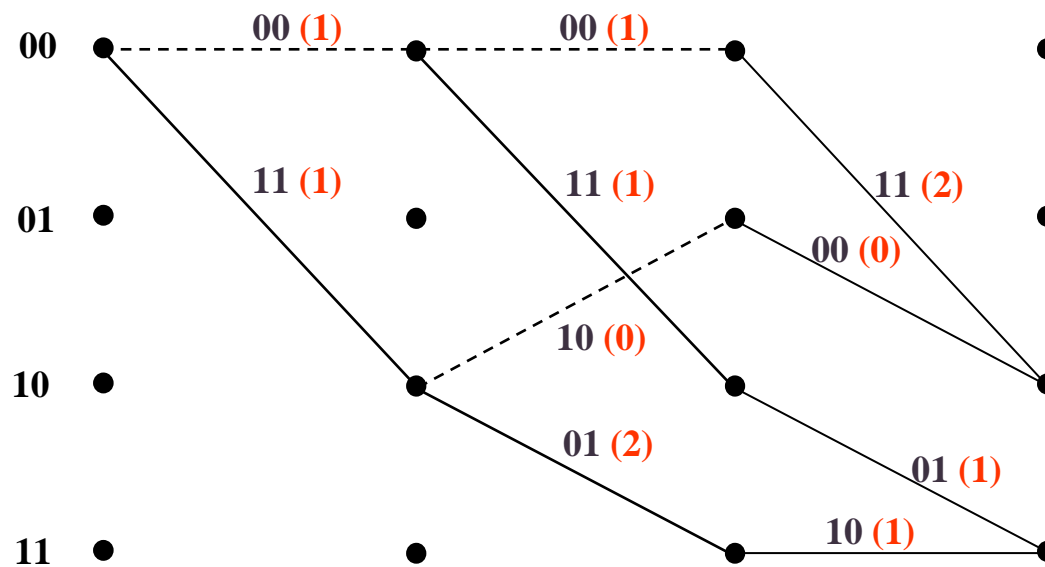
Décodage algorithme de Viterbi

On mémorise le chemin le plus court



Décodage algorithme de Viterbi

Séquence reçue: 10 10 00



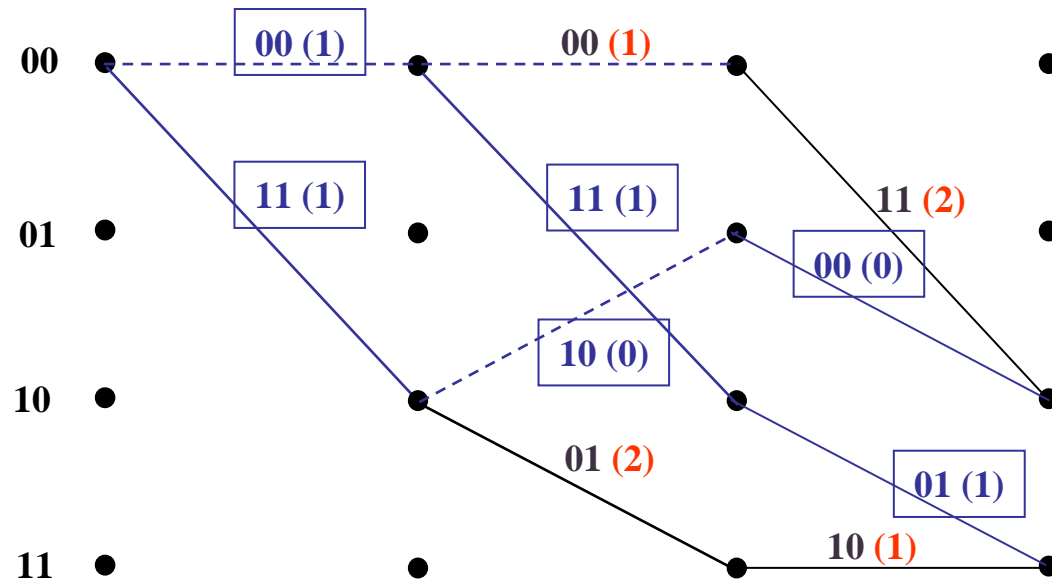
—— 1 à transmettre

----- 0 à transmettre

Décodage algorithme de Viterbi

On mémorise les chemins les plus courts

Séquence reçue: 10 10 00

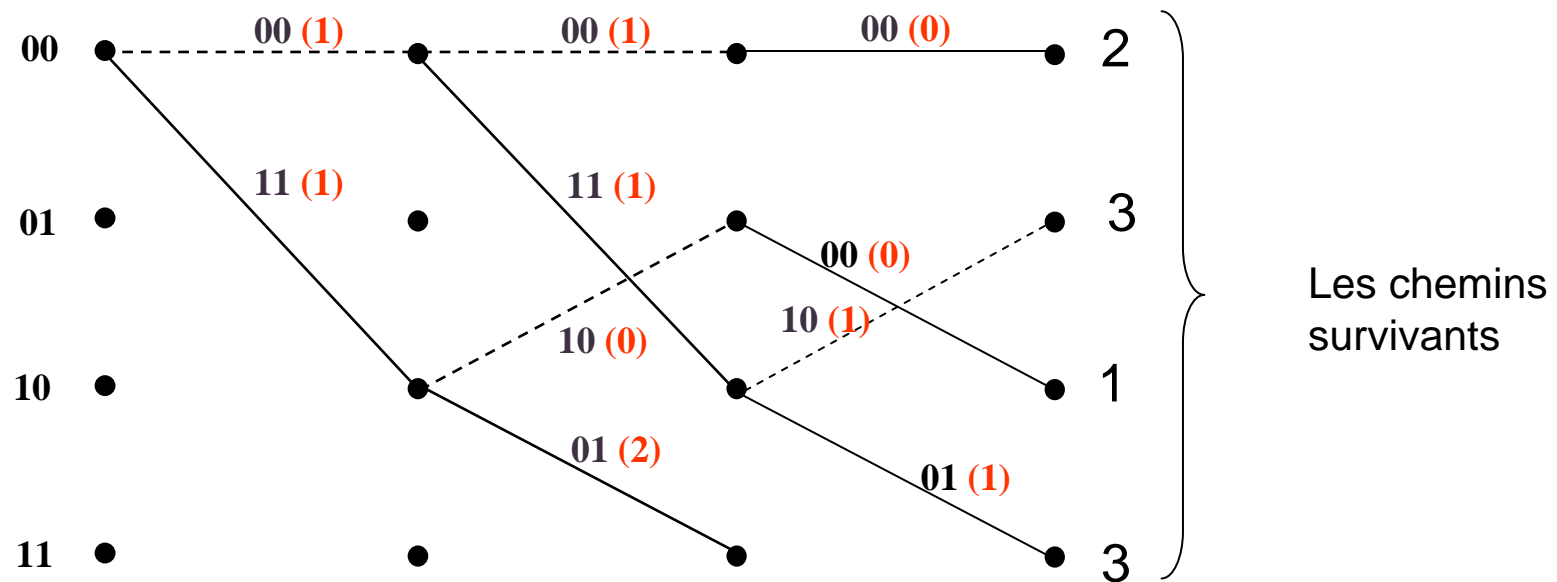


—— 1 à transmettre

----- 0 à transmettre

Décodage algorithme d Viterbi

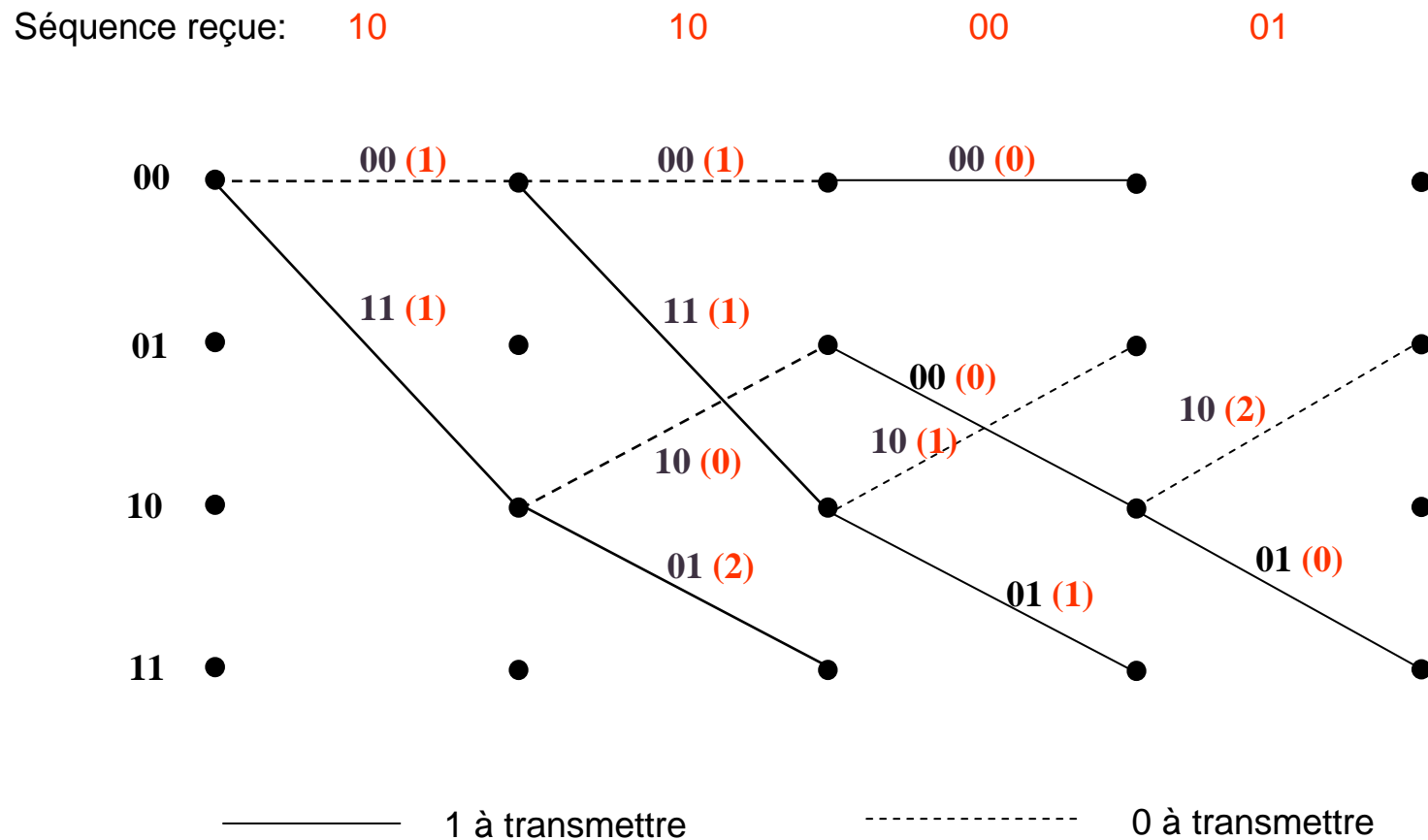
Séquence reçue: 10 10 00



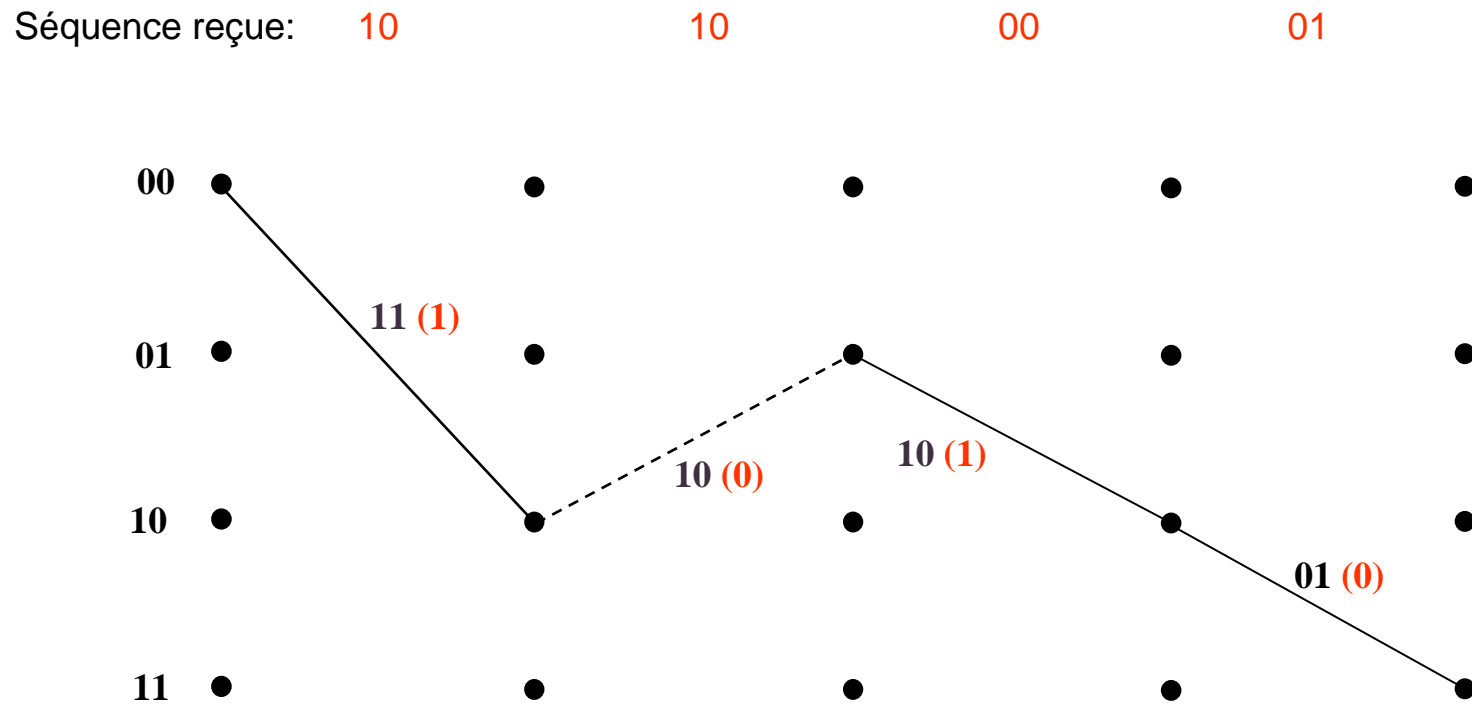
—— 1 à transmettre

----- 0 à transmettre

Décodage algorithme de Viterbi



Décodage algorithme de Viterbi



Séquence la plus probable 1 0 1 1

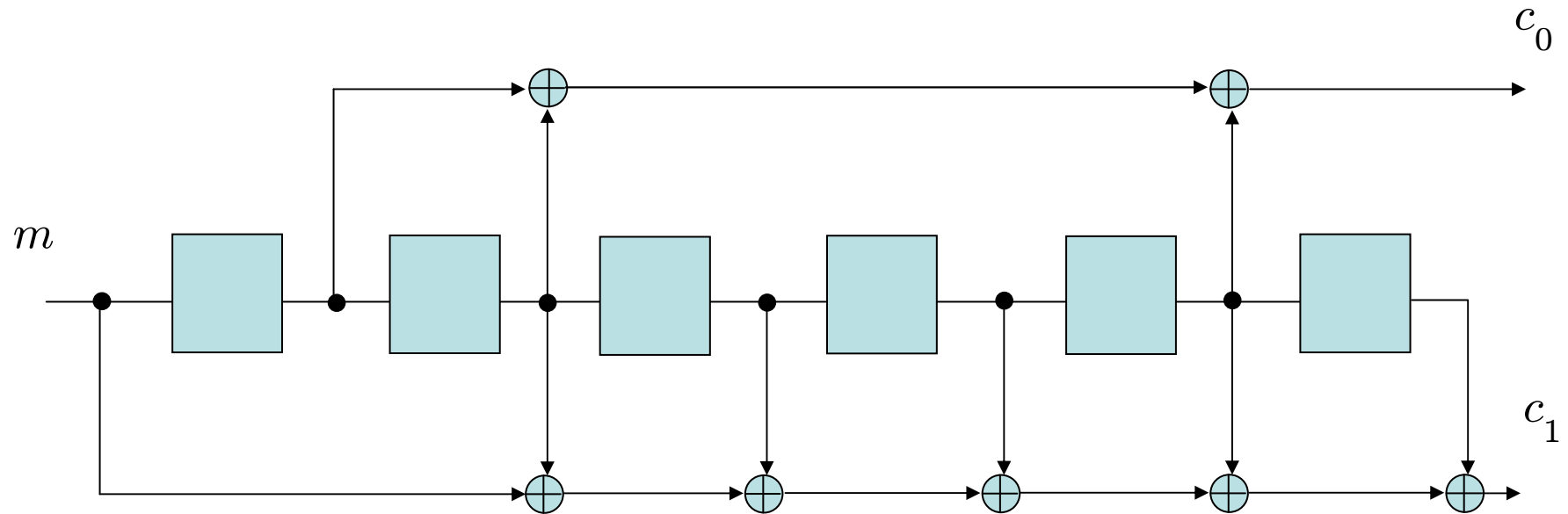
Décodage

algorithme de Viterbi

- Avec le décodage de Viterbi on ne doit pas mémoriser tous les chemins, seulement les plus courts. La complexité est proportionnelle au nombre d'états.
- Le codage convolutif et l'algorithme de Viterbi permettent de corriger les erreurs de transmissions quand elles ne sont pas 'en blocs'
- Les performances dépendent du nombre d'état de l'automate (effet mémoire)
- Les codes convolutifs sont la base des turbo code, parmi les codes les plus efficaces dont les performances s'approchent des limites théoriques.

802.11

Machines d'états

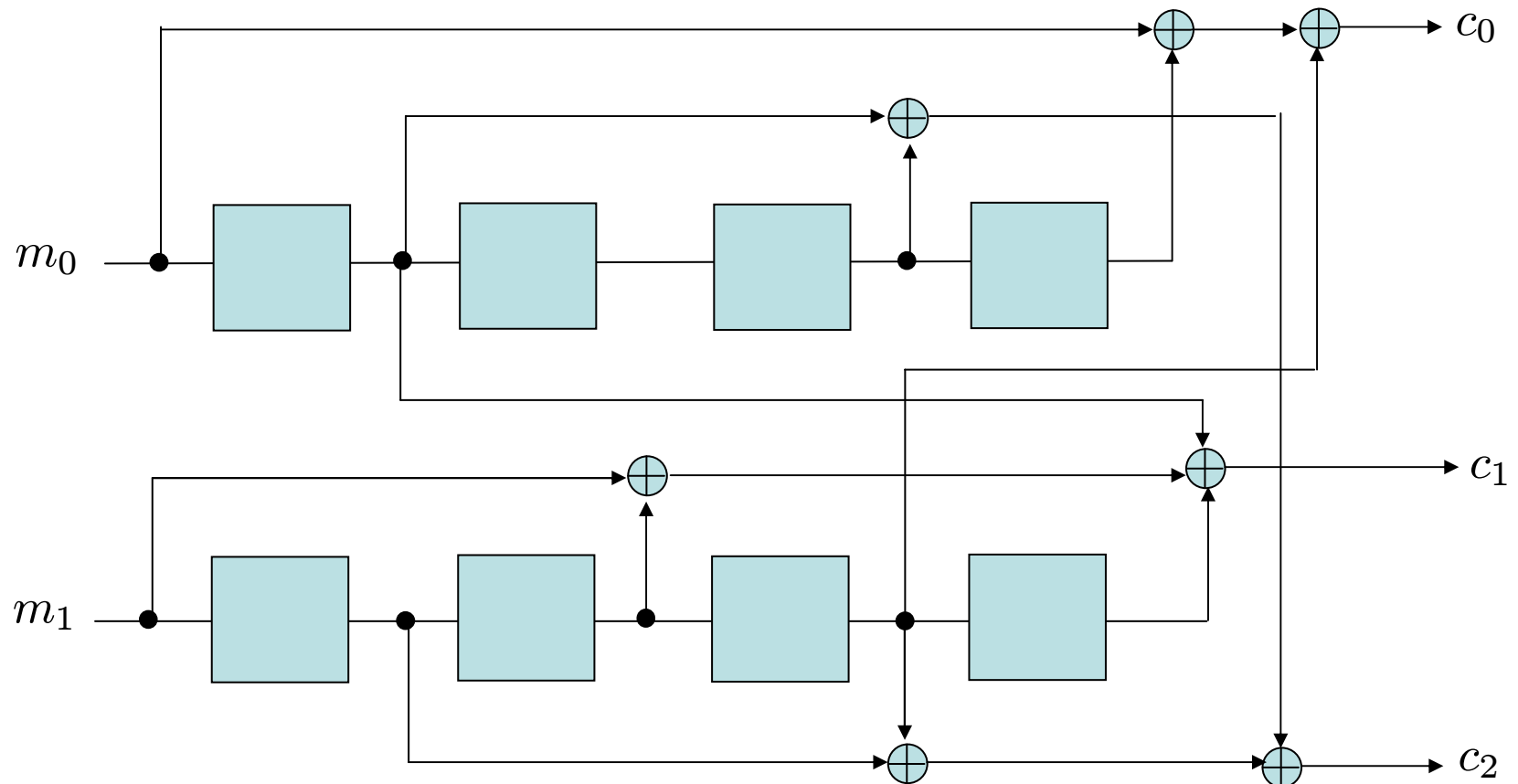


En mode PBCC 5.5 Mbits/sec les bits c_0 et c_1 sont transmis en série en utilisant une modulation BPSK

En mode PBSS 11Mbits/sec les bits c_0 et c_1 sont transmis en parallèle en utilisant une modulation QPSK

802.11

Machines d'états



Les bits c_0, c_1, c_2 sont transmis avec modulation 8PSK à 22Mbits/sec.

Remarques

Le décodage présenté est qualifié de *décodage ferme* (hard decoding) car à la sortie du démodulateur on obtient des valeurs binaires.

Certains décodeur sont qualifiés de *décodeur souple* (soft decoding), car à la sortie du décodeur on obtient une valeur réel qui peut être proportionnelle à la probabilité d'observer un 0 ou un 1.

Résumé

On a vu les modèles de chaînes de Markov homogènes et l'analyse du régime stationnaire.

On a vu comment analyser les performances de protocoles en régime stationnaire comme des processus cycliques.

On a vu différents types de protocoles de la couche MAC pour l'accès au canal de transmission:

- Protocole à accès aléatoire (Aloha)
- Protocole CSMA/CD (Ethernet)
- Protocole CSMA/CA (802.11)
- Protocole à partage de ressources (token)

On a vu le principe des codes convolutifs et le décodage de Viterbi.

Interconnexion des réseaux de niveau 2 (LAN)

répéteurs, hubs, ponts, switch
protocole 802.1D

Adresses MAC

Dans la présentation des réseaux Aloha, Ethernet, 802.11 les adresses n'ont pas été discutées.

On a considéré uniquement l'accès au canal.

Dans les trames des champs sont occupés pour communiquer les adresses de la station émettrice et réceptrice.

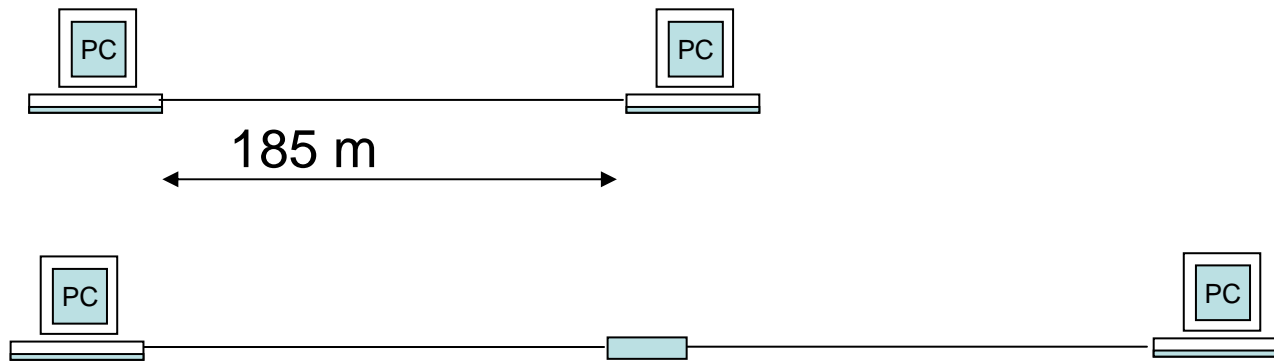
Ces adresses doivent être utilisées pour **router** les trames si les stations n'appartiennent pas au même domaine de collision (= ne communiquent pas directement).

En particulier, lorsqu'on interconnecte différents réseaux LAN (Ethernet par exemple).

Interconnexion

Répéteur: (couche 1)

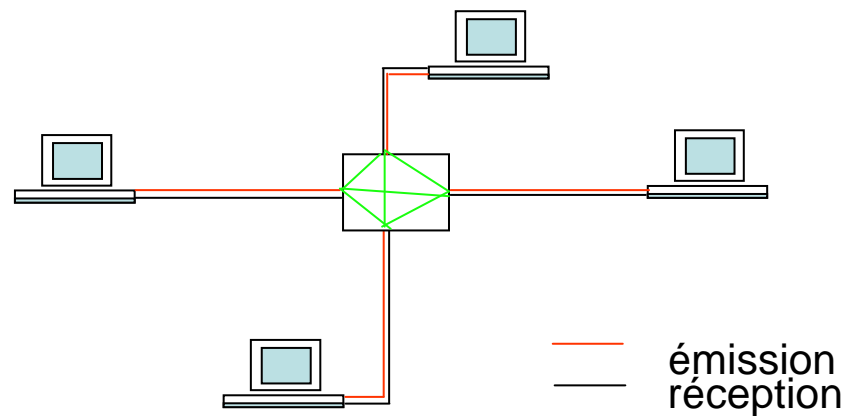
Les répéteurs sont utilisés pour régénérer le signal électrique, ils sont aussi appelés *concentrateurs*. On les utilise pour augmenter la distance séparant deux machines. Par exemple, la distance maximale est de 185m pour un réseau Ethernet/802.3 avec câble coaxial (10BASE2). Ce sont des éléments analogiques



Interconnexion

les hubs: (couche 1)

Les hubs opèrent au niveau de la couche physique. Ils disposent de plusieurs lignes d'entrées toutes interconnectées. les hubs représente un domaine de collisions. Généralement les lignes entrantes ont le même débit et ils n'amplifient pas nécessairement le signal.



Interconnexion

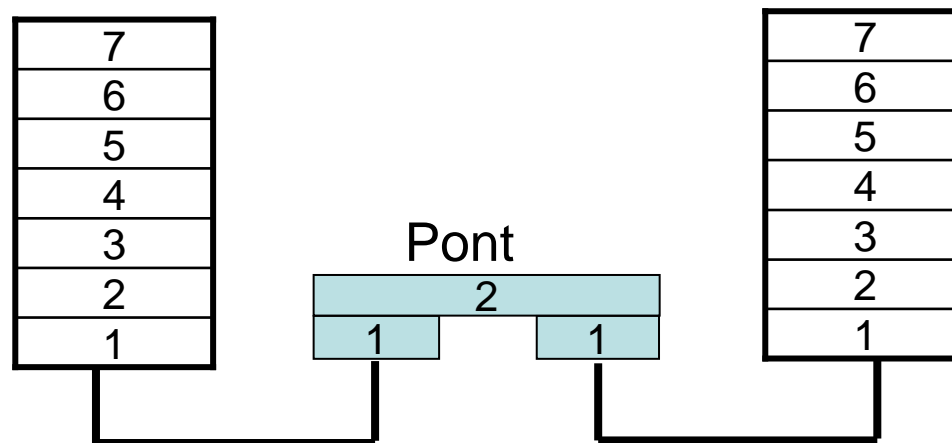
Pont : (bridge, couche 2)

Un pont (bridge) permet l'interconnexion de LAN ou segment de LAN.

Les ponts sont utilisés pour connectés des LAN de types différents. Par exemple, un LAN Ethernet/802.3 avec un Token Ring/802.5.

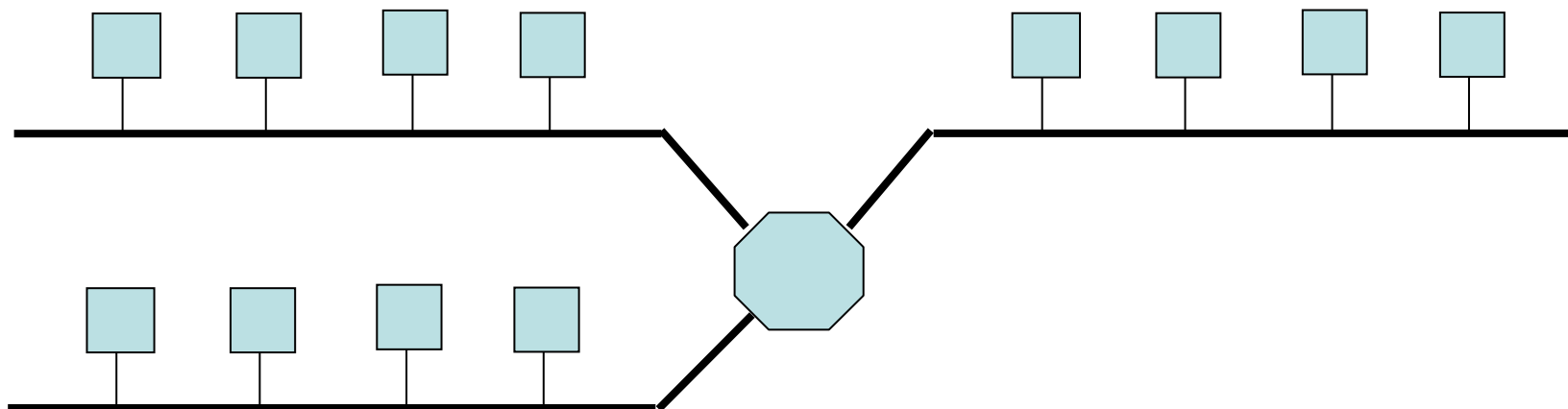
Les ponts **utilisent les adresses de trames** pour acheminer les données.

Chaque lignes d'entrées d'un pont représente un domaine de collision.



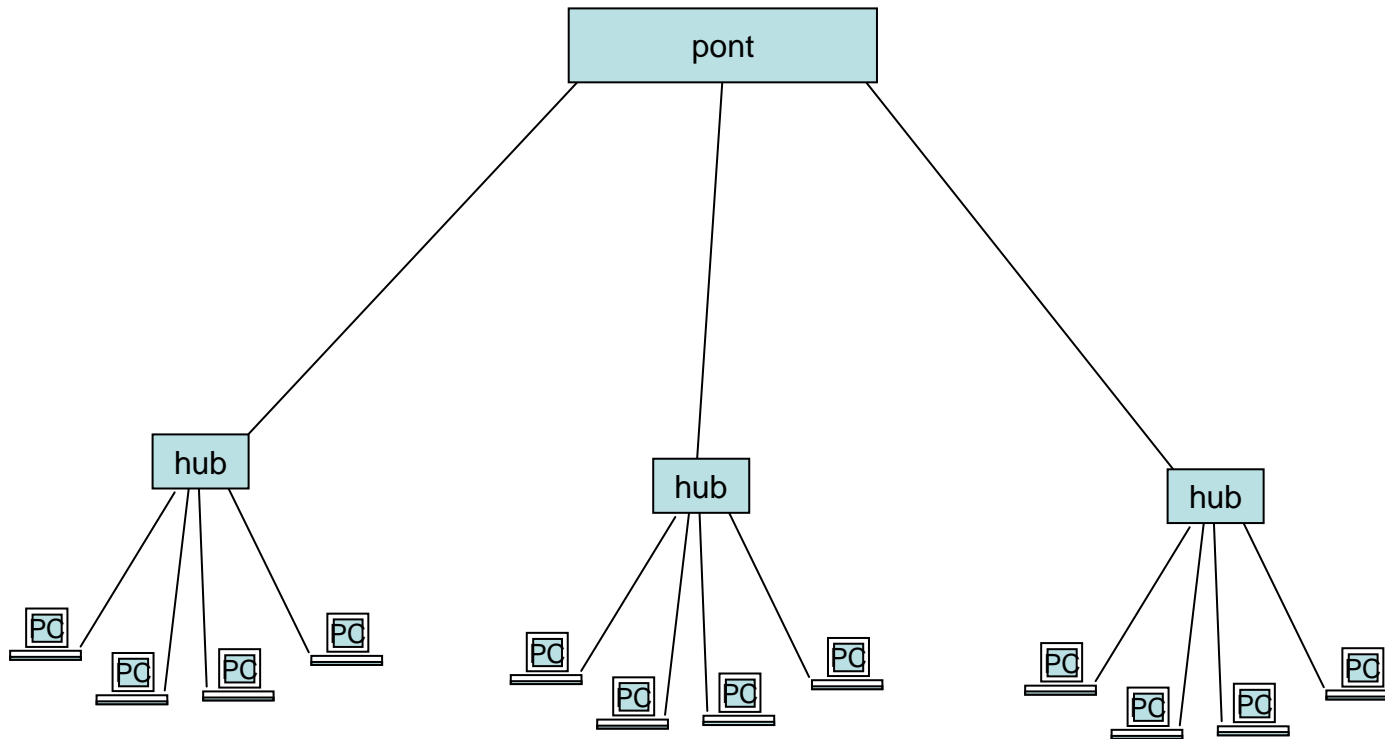
Ponts

pont (suite):



Ponts

Trois domaines de collisions, les hub peuvent avoir des caractéristiques différentes
10 Mbits/sec., 100Mbits/sec., 1Gbits/sec.



Ponts

Les ponts implémentent les fonctions de **filtrage** et de **réexpédition**.

Le filtrage consiste à déterminer si une trame doit être transmise sur un port (interface) ou non.

La réexpédition consiste à déterminer sur quelle interface doit être transmise une trame à l'aide d'une **table** qui contient

- L'adresse LAN des stations
- L'interface de pont qui y conduit
- L'heure d'actualisation des données

Table de pont

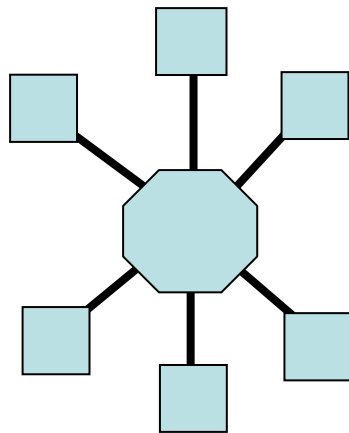
- Initialement la table de pont est vide
- Lorsqu'une adresse de destination ne se trouve pas dans la table, le tampon transmet la trame sur toutes ces interfaces.
- Lorsque le pont reçoit une trame il insère dans sa table l'adresse de l'expéditeur, le numéro de l'interface par laquelle est arrivée la trame et l'heure (c'est l'apprentissage)
- Lorsqu'une trame arrive sur une interface et que l'adresse de la trame se trouve dans la table, le pont transmet la trame sur l'interface correspondante (si elle est différente de celle par laquelle la trame est arrivée – c'est le filtrage)
- Le contenu de la table est périodiquement effacé.

Un pont réalise l'**auto-apprentissage** de la structure du réseau, on dit aussi qu'il est **auto-adaptatif**. On parle aussi de technologie **plug-and-play** et de ponts **transparents**.

Interconnexion

Commutateurs: (switches, couche 2)

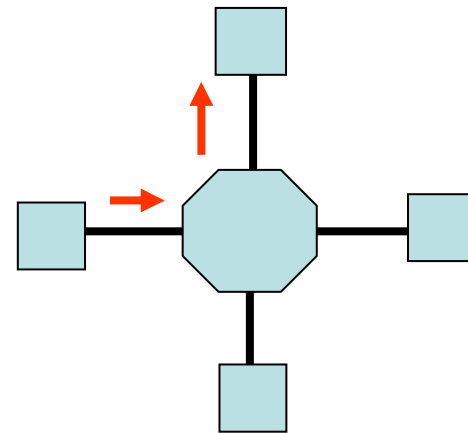
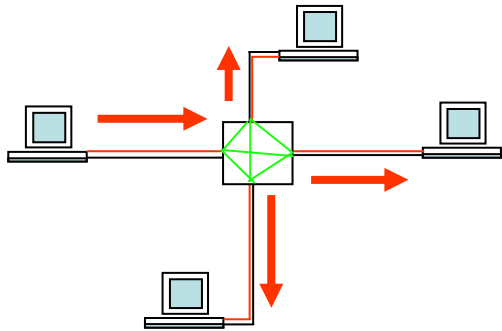
Les commutateurs fonctionnent comme les ponts (routage sélectif en fonction de l'adresse de destination), mais généralement on y connecte directement les ordinateurs. En général ils disposent de plus d'interfaces qu'un pont et ont des performances élevées. Les commutateurs sont apparus après les ponts, principalement les **commutateurs Ethernet**. La plupart des commutateurs travaillent en mode duplex intégral.



Interconnexion

commutateurs: (suite)

On utilise les commutateurs dans **les réseaux LAN** comme on utilise les hubs. La différence est que le commutateur transmet les trames uniquement sur le port concerné.



Interconnexion

commutateurs: (suite)

Avec un commutateur plusieurs station peuvent émettre en même temps, par exemple avec 4 ports 1-2 et 3-4 on double le débit avec 8 ports on multiplie par quatre, ...

Les commutateurs mémorisent les trames à transmettre.

Le domaine de collision est réduit aux liens entre le commutateur et une station.

Les communications sont full-duplex avec un commutateur.

Commutateurs

Pour le routage des données le commutateur dispose d'une table qui associe les adresses MAC aux ports sur lesquels il doit transmettre l'information.

Découverte: Chaque fois qu'un commutateur reçoit une trame à transmettre, il examine l'adresse de l'expéditeur et associe cette adresse au numéro de port sur lequel il a reçu la trame.

Les trames contiennent les adresses source et destination.

Le commutateur effectue une opération de **transmission** en faisant suivre la trame sur le bon port et une opération de **filtrage** en ne la transmettant pas sur les autres ports.

Commutateurs

Inondation: (flooding)

Lorsque le commutateur reçoit une trame à transmettre à une adresse MAC ne figurant pas dans sa table de correspondance, il transmet la trame sur tous ces ports sauf sur le port de réception (comme un hub).

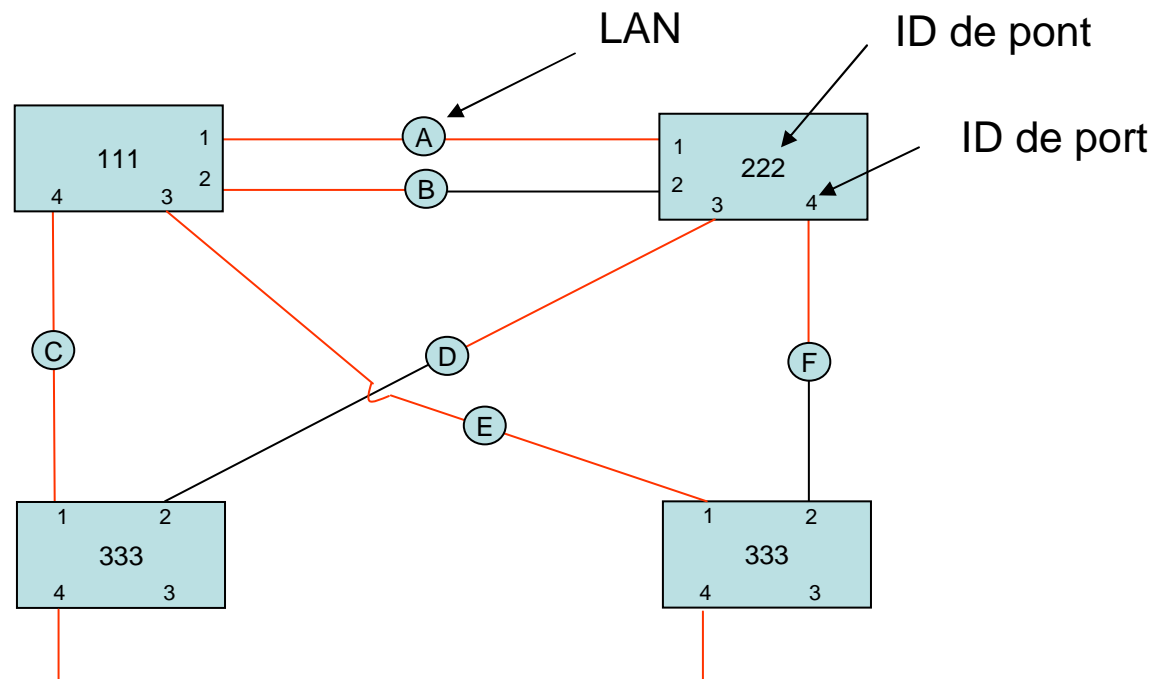
Adresse de broadcast FFFF.FFFF.FFFF.

Les commutateurs permettent aussi d'interconnecter des cartes réseaux qui fonctionnent à des vitesses différentes. A l'initialisation, il y a une phase de négociation entre le commutateur et les hôtes pendant laquelle la vitesse de transmission est déterminée. Par exemple, les cartes Ethernet 10/100 Mbits/s, permettent la négociation avec le commutateur

Spanning tree protocol 802.1D

Arbre de recouvrement

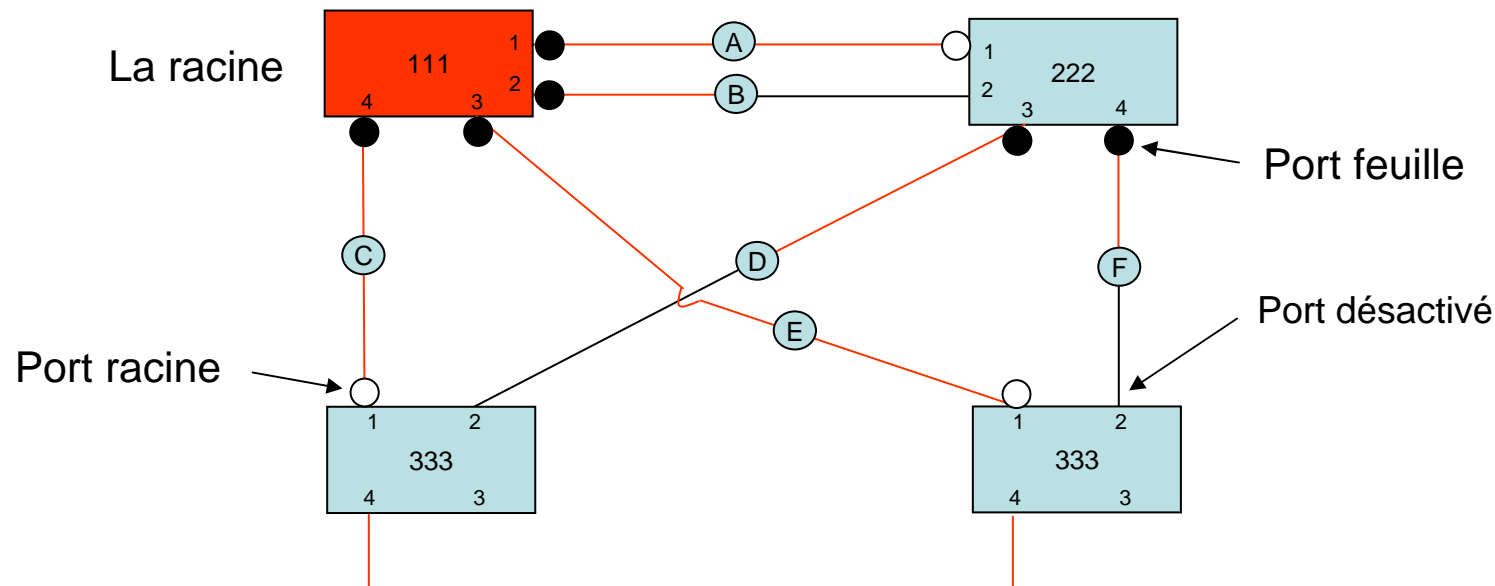
La propagation des messages peut se faire en boucle lorsque des LAN sont interconnectés par des ponts transparents. On restreint la topologie du réseau à un arbre de recouvrement.



Arbre de recouvrement – 802.1D

Pour construire l'arbre de recouvrement on doit :

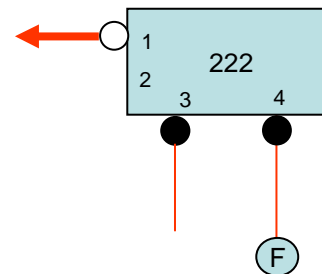
- Désigner un nœud comme étant la racine de l'arbre
- Pour chacun des ports décider si ce port est actif/désactivé, si c'est un port *racine* ou *feuille*



Ports

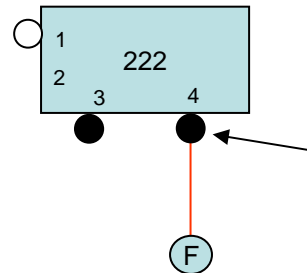
Un pont possède un unique port racine, c'est le port qui relie ce pont au pont racine par le chemin le plus court.

Chemin le plus court
vers le pont racine.



Ports

Un **port feuille** (designated port) est attribué à chaque réseau LAN.



Port feuille attribué au réseau F.

Contraintes algorithmiques

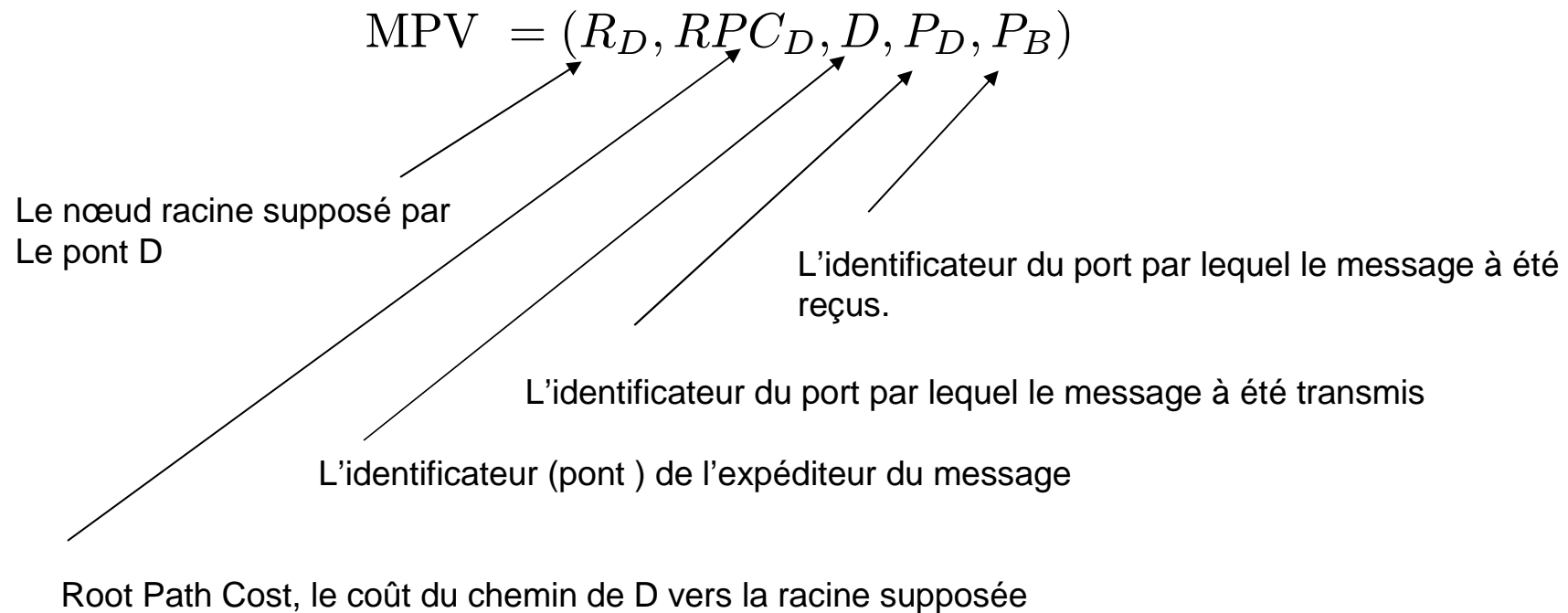
- L'algorithme doit sélectionner les ports des ponts pour qu'il transmettent/ignorent les données de manière à ce que la topologie résultante soit un arbre de recouvrement (arbre = pas de cycle, recouvrement = complète)
- Les ponts doivent *trouver* la bonne topologie de manière indépendante, sans l'aide d'un coordinateur extérieur et s'adapter aux changements de topologie (panne)
- L'algorithme doit se stabiliser rapidement, ici proportionnellement au diamètre du graphe de communication.

Contraintes algorithmiques

- Les ressources nécessaires aux ponts pour exécuter l'algorithme doivent être indépendantes du nombre de ponts et de LAN dans le réseau.
- Les ponts doivent uniquement posséder un identificateur, une adresse MAC.
- Chaque port d'un pont doit être identifié de manière unique
- Une adresse commune doit permettre à tous les ponts connectés à un même LAN de communiquer ensemble.

Structure de données

Les ponts communiquent en échangeant des BPDUs (Bridge protocol Data Unit) comme les messages MPV (Message Priority Vector)



Structure de données

On définit un ordre total sur ces données

$$(a, b, c, d, e) < (v, w, x, y, z) \iff$$

$$a < v \parallel$$

$$(a = v) \&\& (b < w) \parallel$$

$$(a = v) \&\& (b = w) \&\& (c < x) \parallel$$

$$(a = v) \&\& (b = w) \&\& (c = x) \&\& (d < y)$$

$$(a = v) \&\& (b = w) \&\& (c = x) \&\& (d = y) \&\& (e < z)$$

Les deux premiers champs vont permettre de désigner le pont racine ainsi que de calculer un coût pour atteindre la racine. Les autres données sont principalement introduites pour éviter que deux ports aillent des priorités identiques.

Transmission

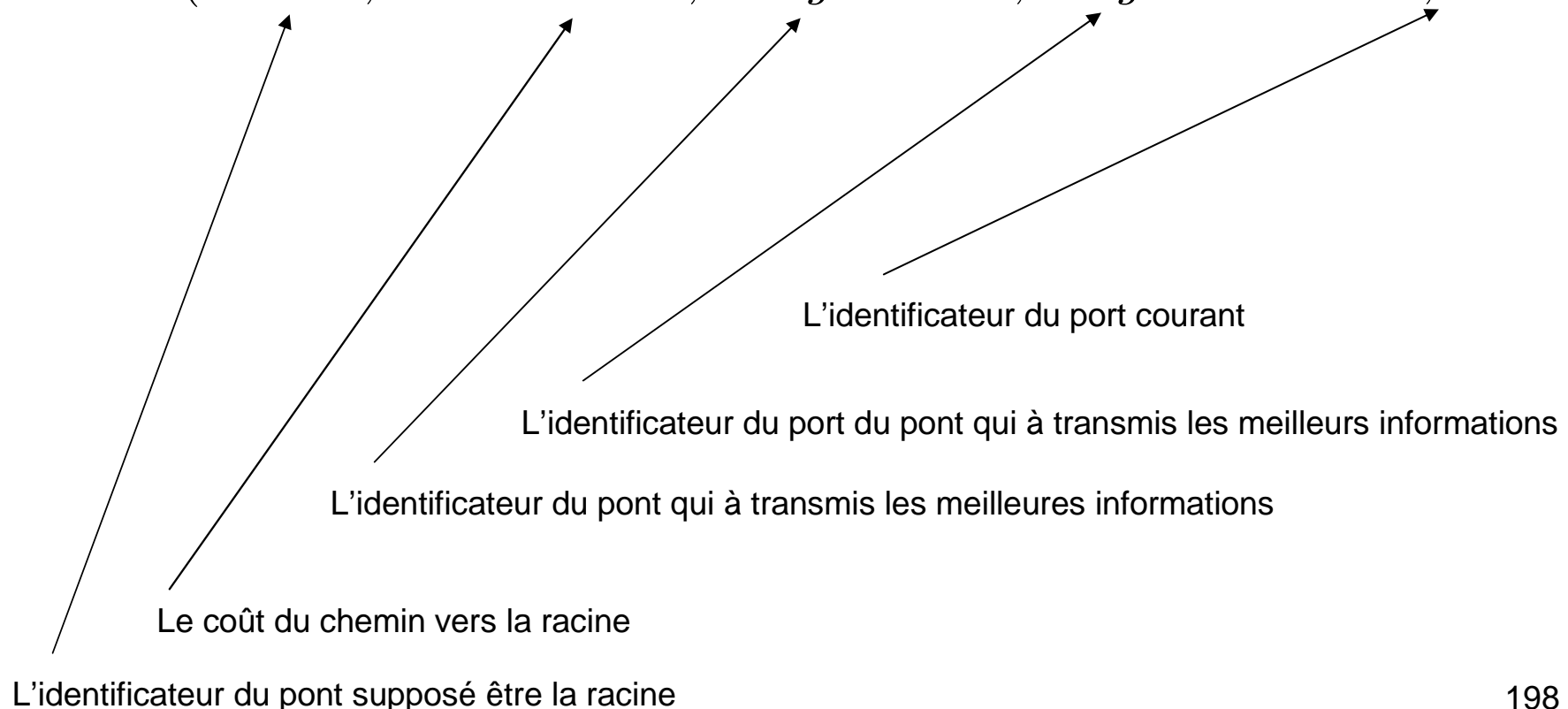
Les ponts participent localement au processus d'apprentissage. Lorsqu'un pont reçoit par un de ces ports une information meilleure (c'est-à-dire plus petite) que celle qu'il aurait transmise par le même port, il cesse de transmettre sur ce port.

Lorsqu'un réseau LAN est connecté à plusieurs ponts ou ports, le port qui transmet en dernier et celui qui est élu le port feuille.

Ports et priorité

A chacun des port est assigné un vecteur de priorité PPV (Port Priority Vector) qui à la même structure que les MPVs.

$$PPV = (RootID, RootPathCost, DesignatedID, designatedPortID, PortID)$$



Ports et mise-à-jour

Initialement à un port i d'un pont B , on assigne le vecteur $PPV=(B, 0, B, i,i)$.

Lorsqu'un message MPV arrive sur un port, ce message est comparé au PPV du port.

- Si $MPV < PPV$ alors $PPV = MPV$
- Sinon, pas de changement.

Intuitivement, on essaye d'élire le pont racine en sélectionnant celui qui possède l'identificateur le plus petit.

A chaque instant et pour chacun des ports i on possède un vecteur

$$PPV_i = (R_{D_i}, RPC_{D_i}, D_i, P_{D_i}, P_B = i)$$

Qui lui permet de déterminer un $RPPV$ (root Path Priority Vector)

$$RPPV_i = (R_{D_i}, RPC_{D_i} + PPC_{P_B}, D_i, P_{D_i}, P_B = i)$$

Le coût pour relier B à D via le port i .

Election du nœud racine

Chaque pont gère un vecteur qui identifie le nœud racine le plus probable, c'est le RPV (Root Priority Vector). C'est le plus petit des vecteurs

$$RPPV_i = (R_{D_i}, RPC_{D_i} + PPC_{P_B}, D_i, P_{D_i}, P_b = i)$$

Et

$$(B, 0, B, 0, 0)$$

← Correspond au vecteur initial, B s'autoproclame être le nœud racine.

Election du nœud feuille

Pour élire un port feuille i d'un pont B , c'est-à-dire un unique port qui connecte un réseau LAN, on détermine

$$DPV_B = \min_i \min\{(B, 0, B, i, i), (R_{D_i}, RPC_{D_i} + PPC_{P_B}, B, i, i)\}$$



Le port i du pont B est désigné comme nœud feuille.

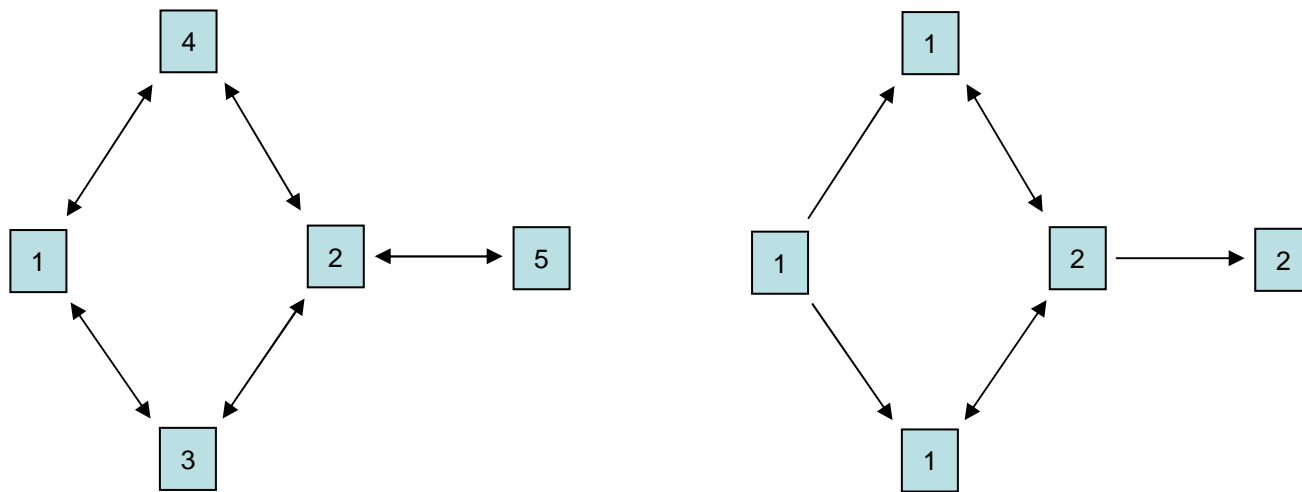
Pour qu'un port i du pont soit élu feuille, il faut que

$$DPV_B < PPV_i$$

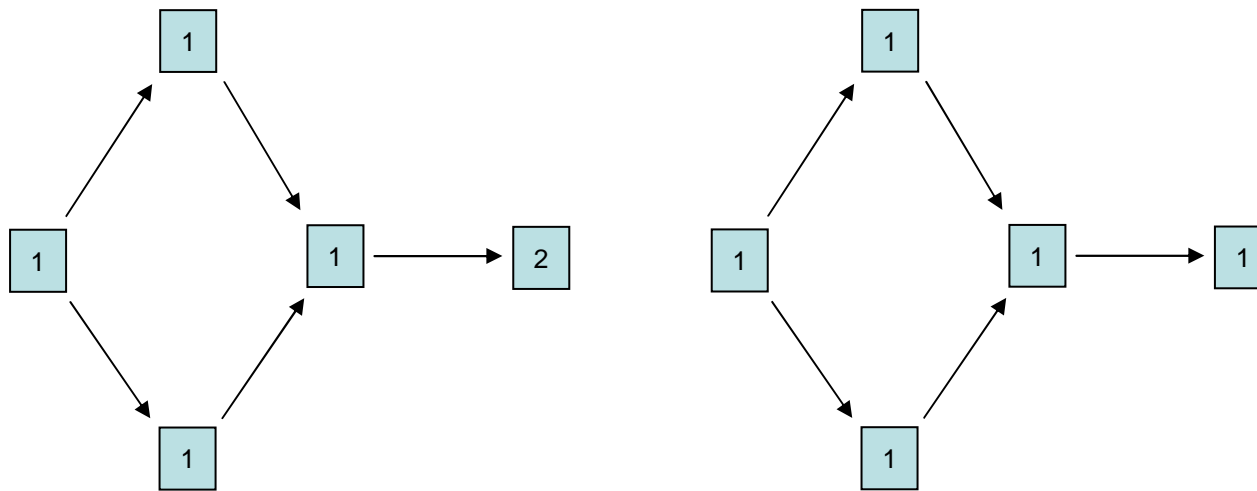
En effet, dans ce cas le pont est le mieux placé par rapport à la racine

Exemple

Propagation du plus petit identificateur



Example



Exemple II

Calcule du nœud racine et du coût du chemin.

On suppose que le pont B possède l'ID 18 et à reçu les PPV associés à chacun de ces ports sont (on ignore les numéros de ports)

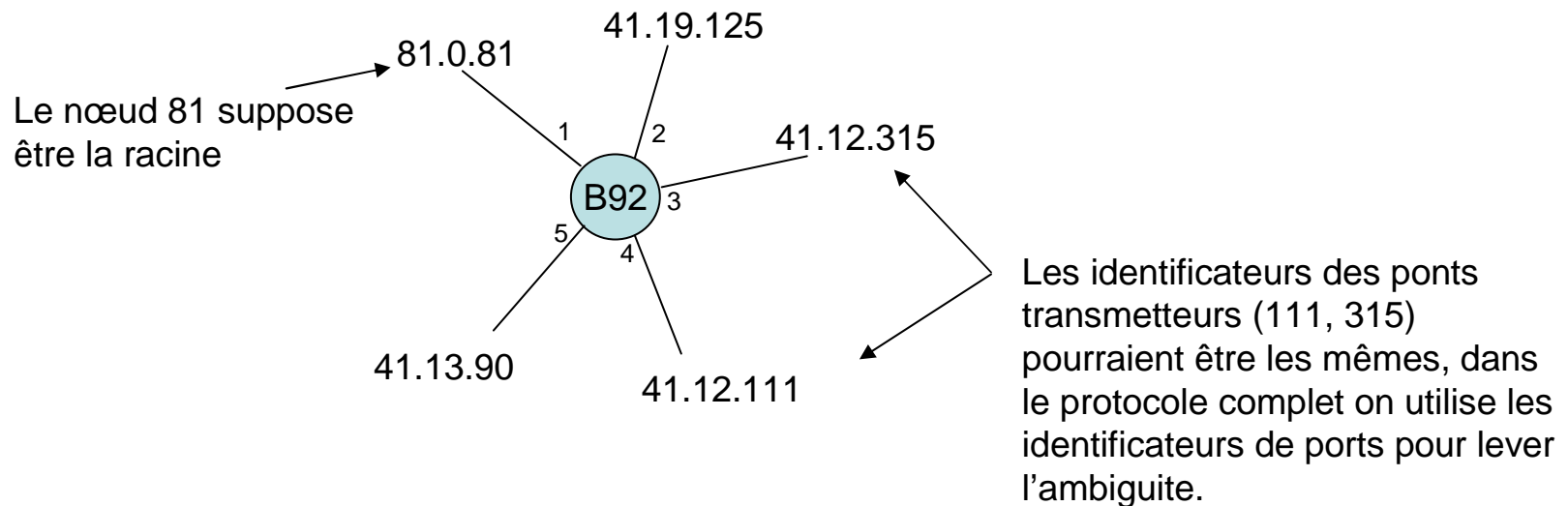
	Racine	Coût	Designated
Port 1	12	93	51
Port 2	12	85	47
Port 3	81	0	81
Port 4	15	31	27

Élu port racine, le pont B transmet le message MPV 12.86.18 sur les ports 1, 3 et 4 et suppose être le nœud feuille sur ces ports jusqu'à réception (peut-être) d'un meilleur vecteur de configuration sur un de ces ports

Exemple III

Le pont B92 possède 5 ports. Le format des vecteurs de configuration est racineID.coût.transmitterID.

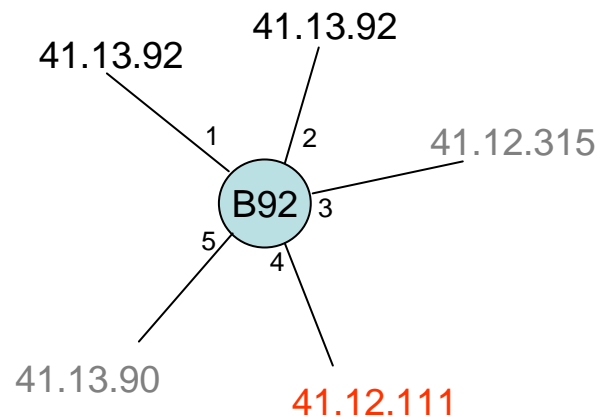
41 est l'identificateur racine pour le pont 92, le coût minimal est $12+1=13$, le port racine est le port 4.



Exemple III

Le pont B92 transmet 41.13.92 sur les ports 1 et 2 qui sont les ports feuilles.

Les ports 3 et 5 sont bloqués et ne participent plus à la topologie



Adresses MAC

Il y a trois format d'adresses MAC

- MAC-48
- EUI-48 (Extended Unique identifier)
- EUI-64

L'attribution des adresses est gérées par l'association IEEE.

Par exemple, les différentes technologies utilisent le format MAC-48:

Ethernet, 802.11 (WiFi), Bluetooth, 802.5 Token ring, FDDI (Fiber-Distributed Data Interface, réseau optique), ATM (Asynchronous Transfer Mode)

Adresse MAC

Le format MAC-48 vient de la spécification d'Ethernet, certaines fois on parle d'adresses MAC pour adresses Ethernet.

Le format EUI-64 est supporté par:

FireWire,

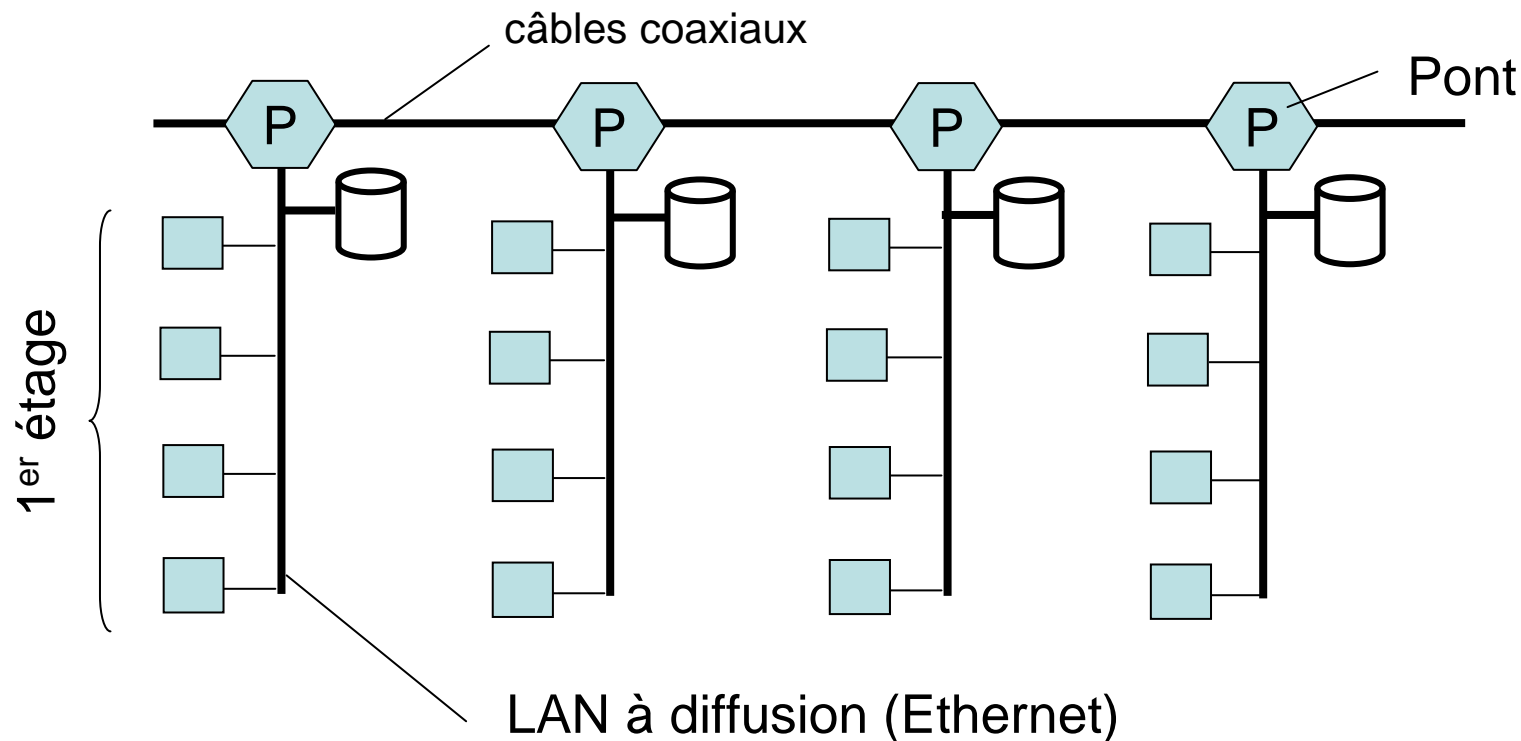
IPv6,

IEEE 802.15.4 (wireless personal-area networks)

Le format EUI-48 n'est pas vraiment une adresse MAC et permet d'attribuer des numéros à des entités logiciels et à d'autres unités matériel.

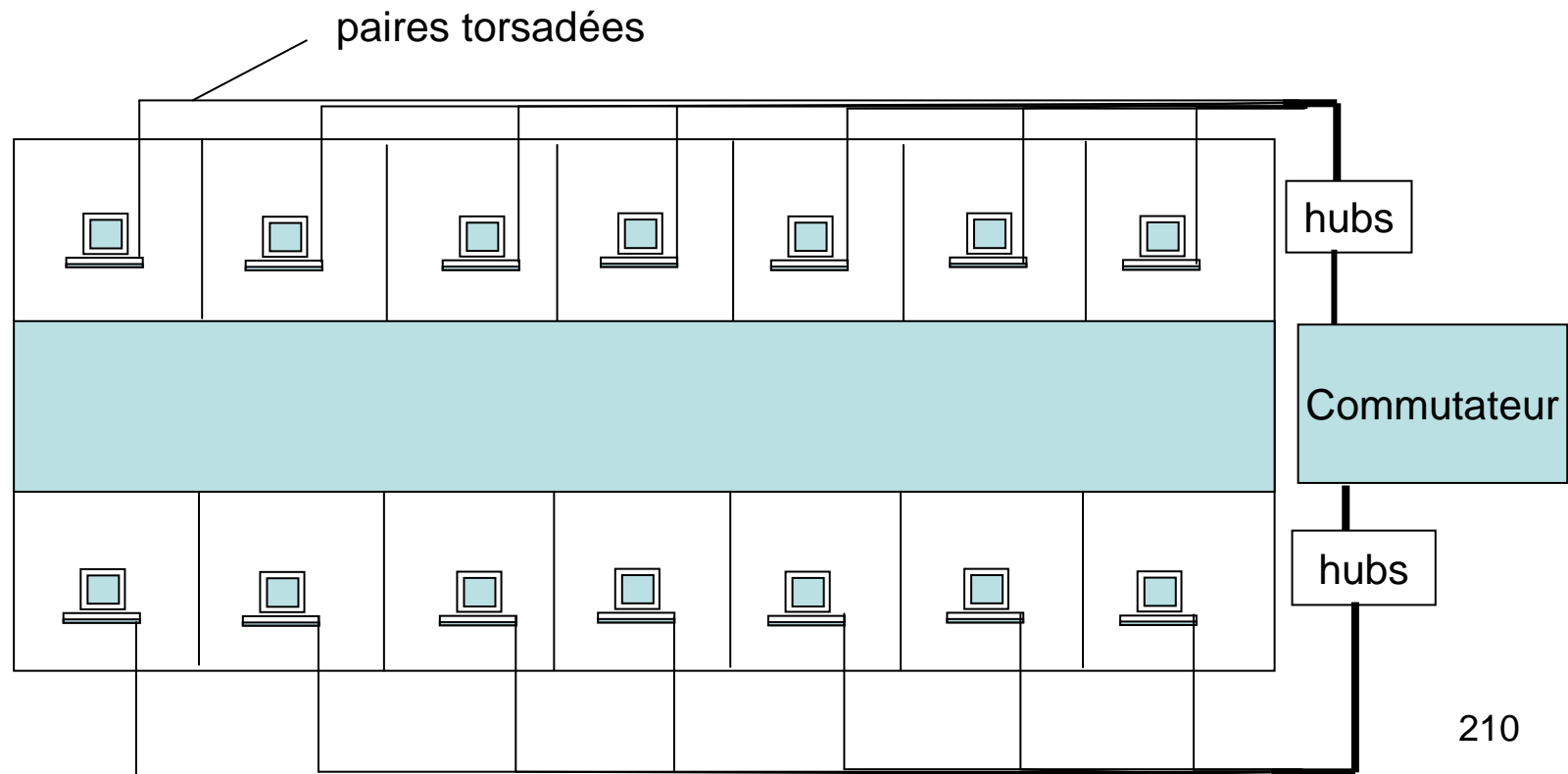
Structure des LAN anciennement

A l'origine le câblage des réseaux informatiques dans les entreprises dépendait fortement de la **position géographique** des ordinateurs. Des ordinateurs proches avaient beaucoup de chance d'être connectés entre eux par un hub. Une configuration classique des réseaux consistait à interconnecter des ordinateurs à un serveur de fichiers.



Structure des LAN

Ensuite, on a centralisé les hubs et les commutateurs dans une même pièce. Pour connecter un hôte à un LAN particulier, il faut le connecter au hub correspondant.



Structure des LAN

Le câblage centralisé consiste à équiper chaque bureau d'une ou plusieurs prises directement reliée à une armoire de câblage.

Malheureusement, cette technique n'est pas toujours possible. Par exemple si un occupant d'un bureau déménage dans un autre bureau depuis lequel le hub n'est pas accessible.

Les **LAN virtuels (VLAN)** ont été introduits pour permettre la division logique du réseau facile et toujours possible. Les VLAN utilisent des commutateurs spécialement conçus.

Division logique du réseau

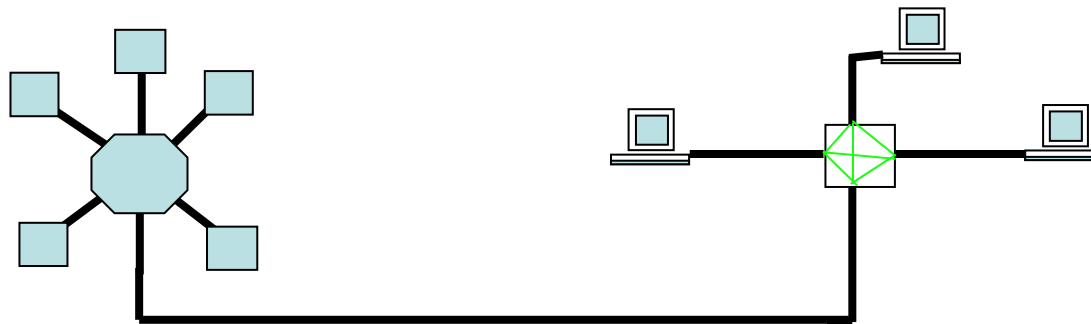
- La taille des domaines de broadcast peut être critique. Par exemple, si trop de trame de broadcast sont générées (découverte) la trafic peut-être pénalisé (augmente linéairement avec le nombre d'utilisateurs).
- Des défaillances des cartes réseaux peuvent provoquer la transmission de trame de broadcast qui réduise complètement la capacité du réseau.
- Pour des réseaux LAN à diffusion, tous les hôtes ont accès à toutes les données qui circulent sur le câble. Certaines informations ne doivent pas être accessibles à tous le monde.
- Si différents département d'une entreprise représente des charges importantes pour le réseau, on gagne en performance à les connecter à des LAN différents.

Domaine de broadcast

Définition: Un domaine de broadcast est un ensemble d'hôtes au sein duquel une trame de broadcast envoyée par l'un d'eux est reçue par tous les autres.

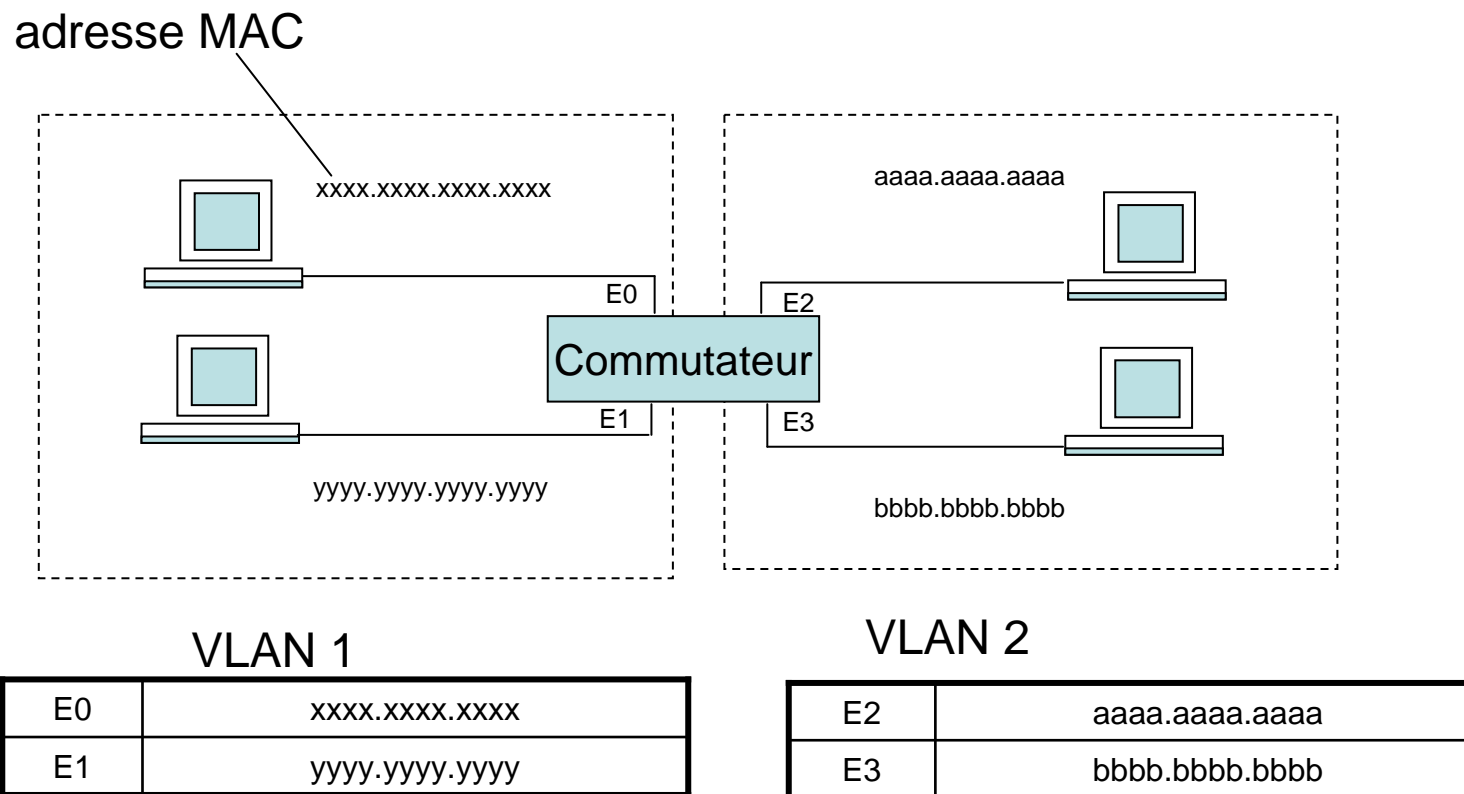
Par exemple:

- l'ensemble des machines connectées par un hub/commutateur
- d'autres connexions hybrides



LAN virtuel

Définition: Un LAN virtuel est constitué du domaine de broadcast créé par un commutateur en utilisant un sous-ensemble de ces ports.



Tables de commutation

LAN virtuel

Chaque VLAN possède sa propre table d'adresse MAC

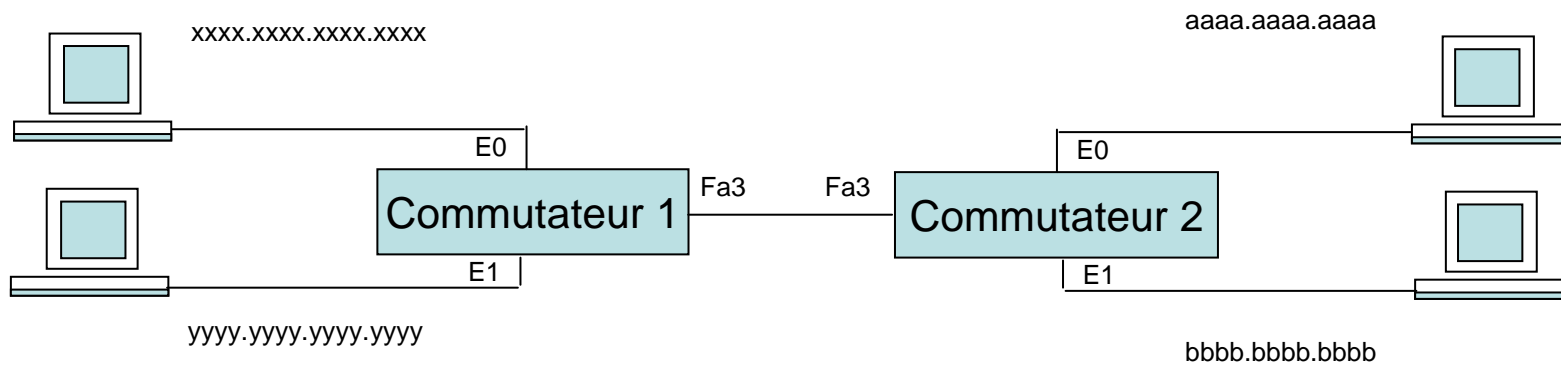
Le trafic broadcast émis au sein d'un VLAN est transmis par inondation à chacun de ces membres

Le trafic broadcast n'est pas émis d'un VLAN vers un autre VLAN.

Le trafic unicast émis au sein d'un VLAN n'est pas acheminé vers un autre VLAN

Transport de trames entre plusieurs commutateurs

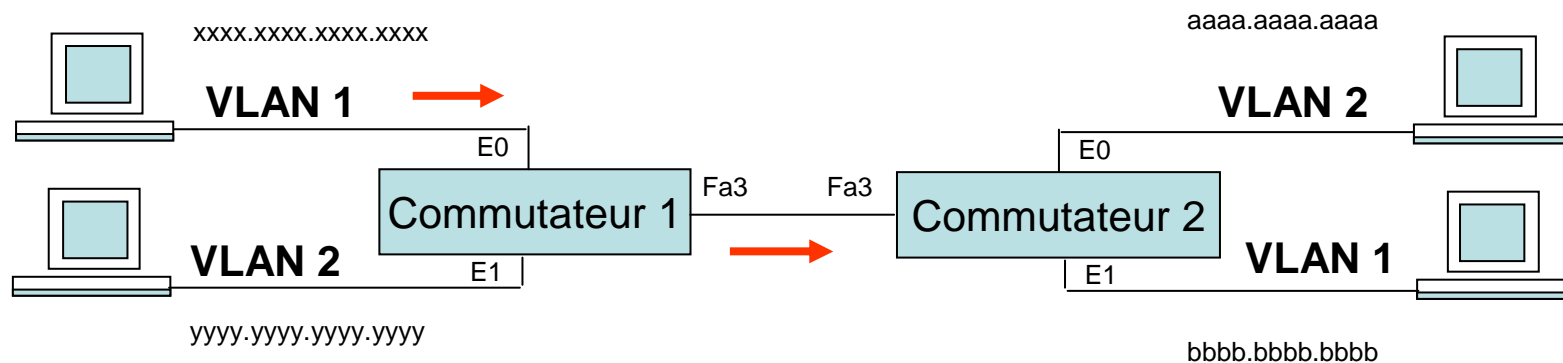
Il est possible d'interconnecter les commutateurs ensemble. La logique de fonctionnement reste la même.



E0	xxxx.xxxx.xxxx
Fa3	aaaa.aaaa.aaaa
Fa3	bbbb.bbbb.bbbb
E1	yyyy.yyyy.yyyy

Agrégation des liens

Les commutateurs sont confrontés à une difficulté lorsqu'ils sont cascades et que des VLAN sont définis.



En effet, lorsqu'une trame est émise par la station xxxx.xxxx.xxxx à l'attention de la station bbbb.bbbb.bbbb. Cette trame transite par les commutateurs.

Quelle table MAC le commutateur 2 doit utiliser?

Que faire s'il doit transmettre une trame de découverte?

802.1Q

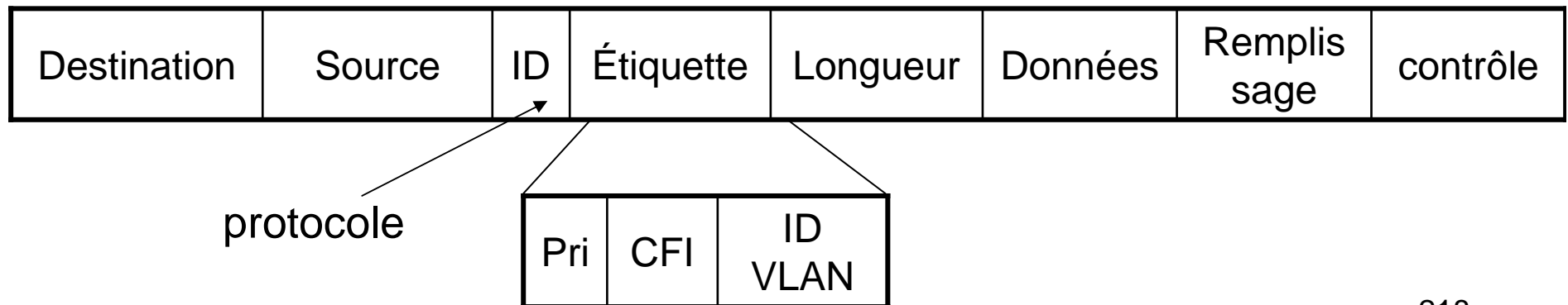
Le problème vient du fait que le commutateur ne peut pas identifier à quel VLAN la trame appartient.

Le comité IEEE propose le protocole 802.1Q qui modifie la trame Ethernet.

802.3



802.1Q



802.1Q

Le premier champ ajouter est *l'identifiant de protocole* qui vaut 0x8100.

Le deuxième champ est le champ *étiquette* qui contient

- L'identificateur de VLAN sur 12 bits
- Un champ Pri, priorité qui ne concerne pas les VLAN mais qui à été ajouté puisque l'occasion se présentait
- CFI Canonical Format Identifier, indique que le champ de données contient une trame 802.5 (Token Ring), est indépendant des VLAN.