Module M2101
Réseaux locaux et équipements actifs
IUT de Villetaneuse
Département Réseaux et Télécommunications
Année 2020/2021

https://lipn.univ-paris13.fr/~petrucci/M2101

Contenu du module

M2101 — Réseaux locaux et équipements actifs

`https://lipn.univ-paris13.fr/~petrucci/M2101`

- Objectifs :
  - comprendre le fonctionnement des réseaux Ethernet
  - savoir configurer des switchs
- plus précisément :
  - Partie 1 : Ethernet standard
    - protocole de communication sur Ethernet (détection/résolution des collisions)
    - interactions entre la couche Ethernet et la couche IP (ARP)
  - Partie 2 : Ethernet commuté
    - commutation (relai des trames par les switchs)
    - protocole STP (utilisé par les switchs)
  - Partie 3 : Réseaux virtuels
    - segmentation d'un réseau Ethernet en réseaux virtuels
    - routage entre réseaux virtuels

Organisation du module

Enseignants

- Mickaël Calixte
- Myriam Graba
- Laure Petrucci
- Abdel Senoussaoui
- Ziad Yassin

Répartition des heures

- 3×1h30 de cours
- 4×3h de TDs
- 4×3h de TPs
- 1 contrôle de 2h

Plan

Plan

Qu'est-ce qu'un réseau local ?

- Des machines
  1. sur une zone géographique limitée (2–3 kilomètres au plus)
  2. reliées selon une certaine **topologie** (bus, anneau, étoile, . . .)
  3. avec un type de support physique : paire torsadée, fibre optique, air, . . .
  4. utilisant le même **mode d'adressage**
  5. et s'échangeant des messages appelés **trames**
  6. selon un **protocole** de communication.
- On parle de **LAN** (**L**ocal **A**rea **N**etwork).
- On parle aussi de réseau **physique**
  - En opposition, par exemple, au réseau internet, qu'on dit **logique**.
- LAN $\Leftrightarrow$ ensemble de machines qui peuvent communiquer sans routeur intermédiaire
- Problématique : assurer la communication entre deux machines sur le même LAN.

Pourquoi normaliser/standardiser ?

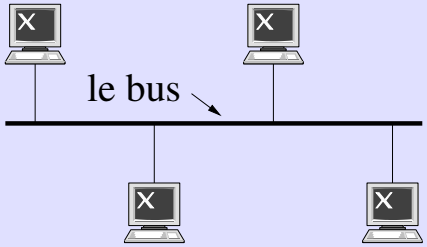
- norme = document qui décrit précisément le fonctionnement d'un réseau : câblage, débit, topologie, protocole, format des trames, ...
- garantie d'un certain niveau de qualité pour l'utilisateur
- On sait exactement quels sont les services rendus par l'équipement ou le logiciel.
- plus facile d'interconnecter des équipements ou logiciels qui respectent le même cahier des charges
  - sans norme : chaque système doit comprendre tous les autres
  - avec norme : chaque système doit comprendre la norme

L'IEEE

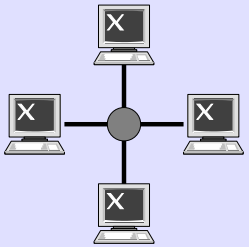
- **IEEE** = **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
- organisation de droit états-unienne à but non lucratif
- créée en 1963
- + de 400 000 ingénieurs et chercheurs en électronique, réseaux, informatique, ...
- activité : publication de normes
- dans le domaine des réseaux : série des normes IEEE 802 autour des réseaux locaux, dont :
  - **IEEE 802.3** = **Ethernet**
    (étudié dans ce module)
  - **IEEE 802.5** = **Token Ring**
    (protocole conçu par IBM, en voie de disparition)
  - **IEEE 802.11** = **Wifi**
    (étudié dans le module M3101)
  - **IEEE 802.1D** = **Spanning Tree Protocol**
    (protocole utilisé par les switchs, étudié dans ce module)

Topologies de réseaux

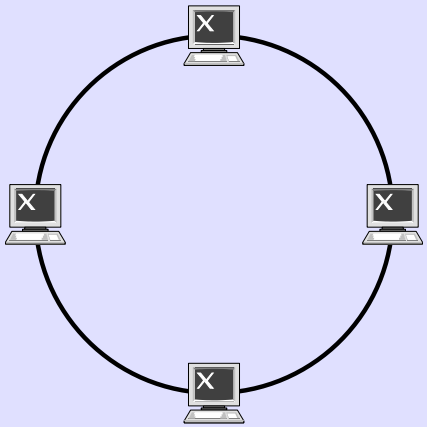
- le bus



- l'étoile



- l'anneau



Topologie physique/topologie logique

- Dans les réseaux locaux, on distingue la **topologie physique** de la **topologie logique**.
  - physique = câblage utilisé (⇒ topologie réelle du réseau)
  - logique = celle sur laquelle repose le protocole de communication
- Pour quelques protocoles de l'IEEE :

Protocole	Norme	Topologie physique	Topologie logique
Ethernet	IEEE 802.3	bus	bus
Ethernet avec hubs		étoile	bus
Ethernet avec switchs		étoile	étoile
Token Bus	IEEE 802.4	bus	anneau
Token Ring	IEEE 802.5	étoile	anneau

- Exemple : sur un réseau Ethernet avec des hubs
  - les machines sont connectées en étoile ;
  - mais comme le hub retransmet les trames qu'il reçoit à toutes les machines, il simule un bus : dès qu'une machine transmet des trames toutes les machines du réseau les reçoivent.

L'adressage MAC

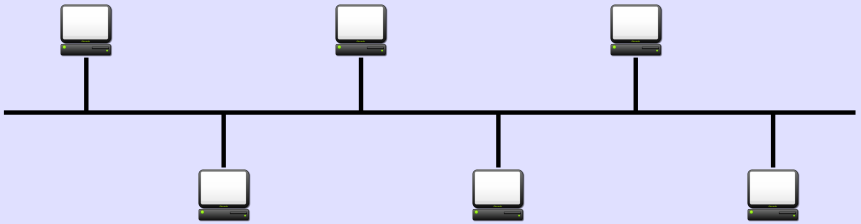
- Mode d'adressage des protocoles réseau de l'IEEE.
- **MAC** = Medium Access Control
- Adresse inscrite sur la carte réseau.
- Elle sert à identifier une machine sur un réseau local.
- codée sur 48 bits = 6 octets
- écrite en hexadécimal par 6 mots de 2 lettres séparés par un ":"
  - octets 1, 2 et 3 = identifiant du constructeur de la carte
  - octets 4, 5 et 6 = identifiant de carte attribué par le constructeur
- Exemple : `00:18:DE:10:FA:87`

Identifiant constructeur	`00:18:DE`	(Intel)
Identifiant carte	`10:FA:87`	(attribué par Intel)

- L'adresse MAC est (en principe) unique.

Caractéristiques des réseaux Ethernet

- développé par Digital, Intel et Xerox dans les années 1970
- normalisé par l'IEEE en 1983 : norme IEEE 802.3
- dans la première version de la norme :
  - débit de 10 Mbit/s
  - topologie logique et physique en **bus** (longueur max. = 2 500 mètres) :



- utilise la technique CSMA/CD
  - **CSMA** — **C**arrier **S**ense **M**ultiple **A**ccess
    accès multiple au support avec écoute de la porteuse
  - **CD** — **C**ollision **D**etection
    détection des collisions
- évolutions ultérieures de la norme pour des débits supérieurs

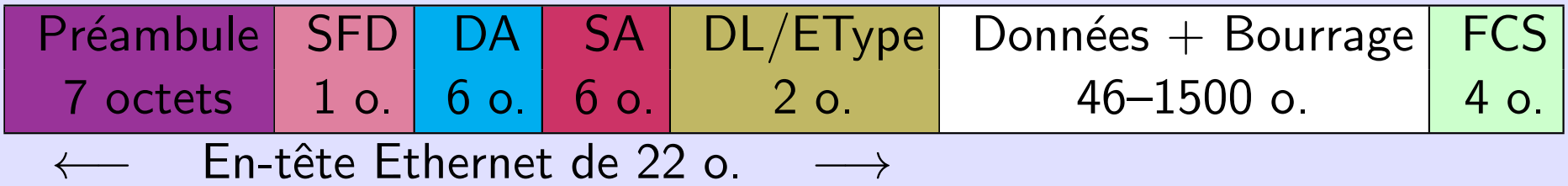
Plan

La trame Ethernet

- La trame Ethernet est une liste de **champs**.
- Chaque champ a une signification particulière définie dans le protocole.
- La trame Ethernet a une structure fixe :
  - un en-tête de 22 octets ;
  - un champ de données (le SDU Ethernet) : le contenu de la trame, généralement un paquet IP ;
  - des octets de bourrage insérés par la carte si la trame est trop petite ;
  - et un en-queue de 4 octets (le FCS).

⇒ Tous les champs mis bout à bout, une trame fait toujours entre 72 et 1526 octets.

- L'en-tête et le FCS forment le **PCI** de la couche Ethernet.



Structure de la trame Ethernet — Préambule et SFD

Préambule 7 octets	SFD 1 o.	DA 6 o.	SA 6 o.	DL/EType 2 o.	Données + Bourrage 46–1500 o.	FCS 4 o.

- Le préambule et le SFD ne font pas vraiment partie de la trame Ethernet.
  - Ce sont des moyens physiques pour marquer le début d'une trame.
  - Ils ne contiennent aucune information intéressante.

Préambule

- séquence de 7 octets à `1010 1010`
- Utilisée pour synchroniser les horloges de l'émetteur et du récepteur.

SFD (Start Frame Delimiter)

- Un octet à `1010 1011` indiquant le début de la trame.
- Seul un bit diffère des octets du préambule.
  - $\Rightarrow$ Le récepteur peut détecter le début de la trame même s'il a mal reçu les 7 octets du préambule.

Structure de la trame Ethernet — DA et SA

Préambule 7 octets	SFD 1 o.	DA 6 o.	SA 6 o.	DL/EType 2 o.	Données + Bourrage 46–1500 o.	FCS 4 o.

- **DA** = Destination Address = Adresse MAC de l'interface destinataire de la trame
- **SA** = Source Address = Adresse MAC de l'interface émettrice de la trame
- Dans le champ DA on peut avoir l'adresse de diffusion `FF:FF:FF:FF:FF:FF`.

$\Rightarrow$ La trame est alors destinée à toutes les interfaces du LAN.

Structure de la trame Ethernet — DL/EType

Préambule 7 octets	SFD 1 o.	DA 6 o.	SA 6 o.	DL/EType 2 o.	Données + Bourrage 46–1500 o.	FCS 4 o.

- Selon la valeur de **DL/Etype** on a 2 possibilités :
  - $\leq 1500 \Rightarrow$ on dit qu'on est en **Ethernet I**
    - Le champ s'appelle alors **DL** = Data Length.
    - Il indique la longueur du champ de données de la trame.
    - Les données encapsulées dans la trame Ethernet forment une trame LLC.
    - LLC = protocole de liaison (non étudié dans le module)
  - $> 1500 \Rightarrow$ on dit qu'on est en **Ethernet II**
    - Le champ s'appelle alors **Etype** = Ether Type.
    - Il indique le protocole auquel appartiennent les données encapsulées.
    - $\Rightarrow$ Le protocole LLC n'est pas utilisé.
    - Exemples : 2 048 $\Rightarrow$ IP, 2 054 $\Rightarrow$ ARP
- Exemples :
  - La trame contient un paquet IP $\Rightarrow$ DL/Etype = 2 048.
  - La trame contient une trame LLC de 1 000 octets $\Rightarrow$ DL/Etype = 1 000.
- Les codes Etype sont tous > 1500 pour éviter toute confusion.

Structure de la trame Ethernet — Données et bourrage

Préambule 7 octets	SFD 1 o.	DA 6 o.	SA 6 o.	DL/EType 2 o.	Données + Bourrage 46–1500 o.	FCS 4 o.

- Le champ **Données** contient le PDU de la couche supérieure.
- Le champ **Bourrage** contient des octets insérés par l'émetteur pour que la trame ait une taille d'au moins 72 octets.
- La taille des données ne peut pas excéder 1500 octets.

$\Rightarrow$ On doit toujours avoir 46 octets $\leq$ Données + Bourrage $\leq$ 1500 octets.

- 46 = 72 - (Préambule + SFD + DA + SA + SFD + CRC) = 72 - 26

Structure de la trame Ethernet — FCS

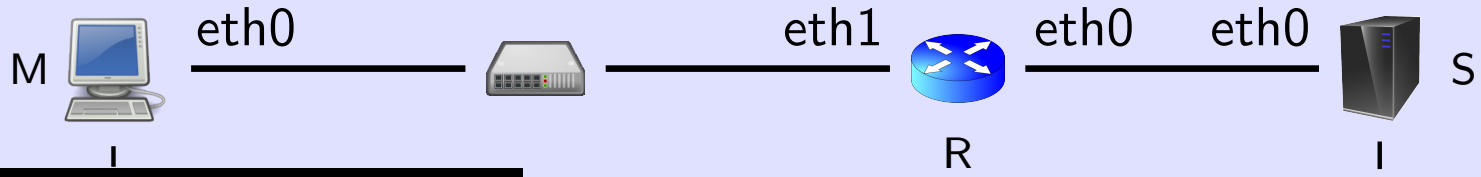
Préambule 7 octets	SFD 1 o.	DA 6 o.	SA 6 o.	DL/EType 2 o.	Données + Bourrage 46–1500 o.	FCS 4 o.

- **FCS** = Frame Check Sequence
- utilisé par le récepteur pour contrôler les erreurs de transmission
- CRC de 32 bits calculé sur toute la trame (hors préambule et SFD)
- utilise un polynôme générateur de degré 32

Rappels sur l'encapsulation

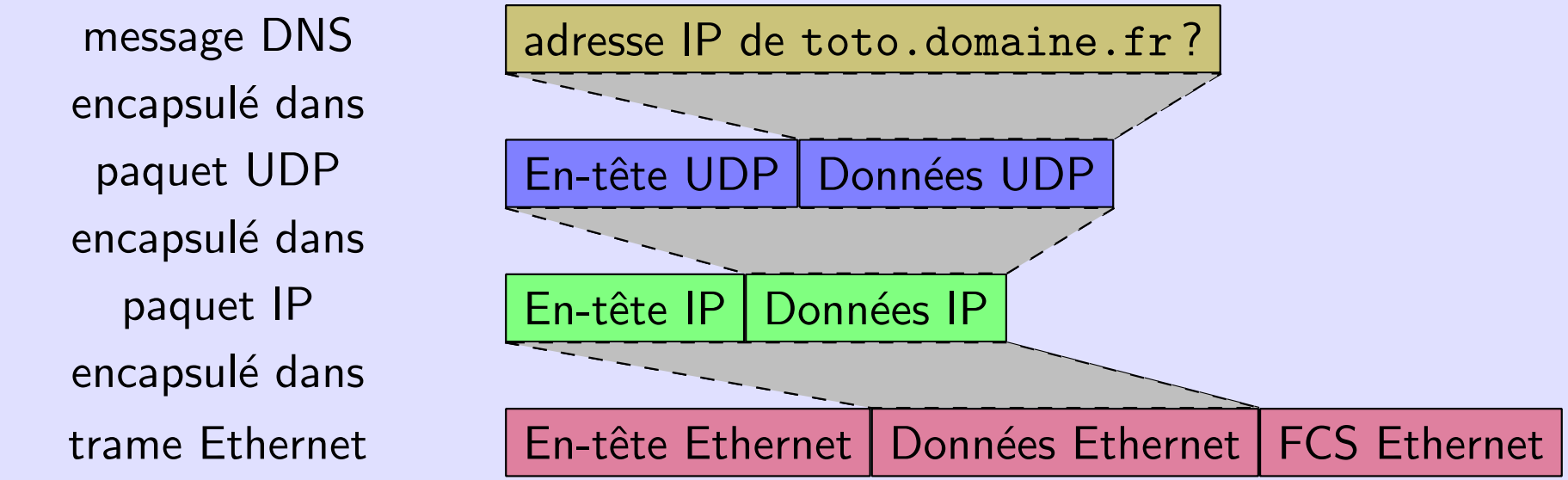
- **encapsulation** = insertion du message d'un protocole de niveau N dans le corps du message d'un protocole de niveau $N-1$
- Pour les communications sur Internet, on a généralement 4 niveaux :
  - application/service : DHCP, DNS, HTTP, ...
  - transport entre applications : UDP ou TCP
  - communication sur Internet : IP
  - communication sur le réseau local : Ethernet, Wifi, ...
- Qui fait quoi ?
  - protocoles d'application/service : les programmes (navigateur web, logiciel de messagerie, serveur DHCP, ...)
  - IP, UDP et TCP : le système d'exploitation
  - Ethernet, Wifi, ... : la carte réseau

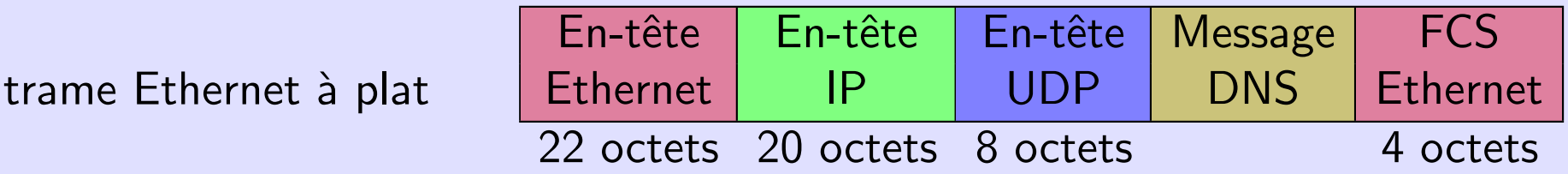
Rappels sur l'encapsulation — Exemple (1/2)



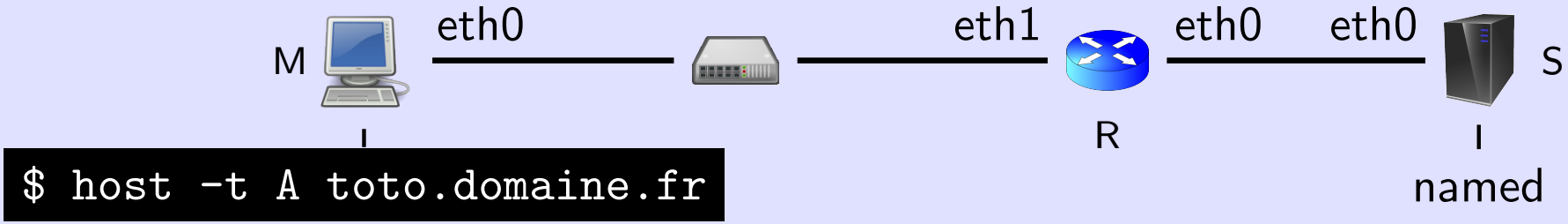

```
$ host -t A toto.domaine.fr
```

- host envoie une requête DNS au serveur named sur S
- encapsulations successives :

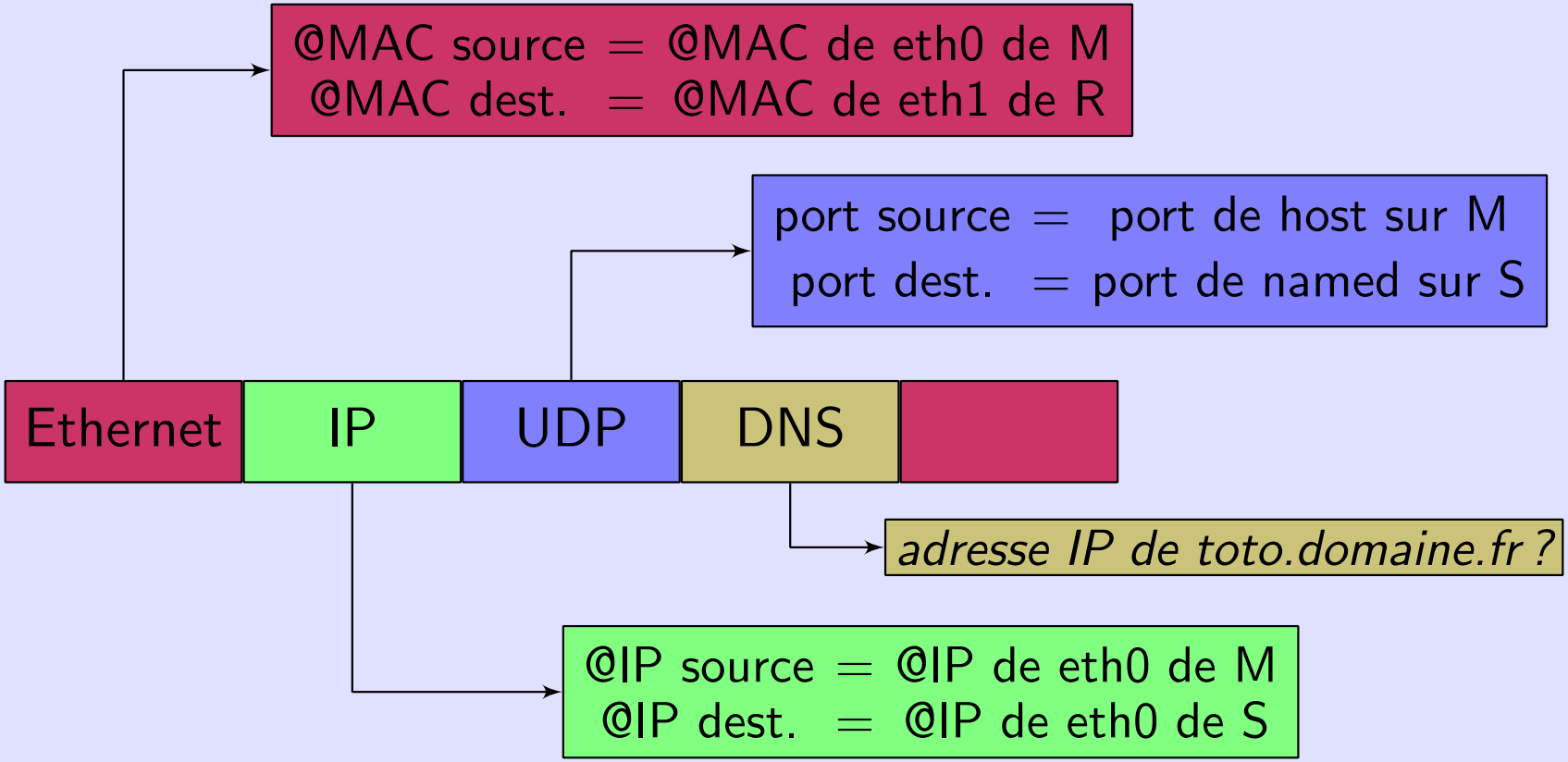




Rappels sur l'encapsulation — Exemple (2/2)



- Adresses et numéros de port apparaissant dans les PCIs :



Difficulté

- Le bus est partagé par toutes les stations.

$\Rightarrow$ 2 stations peuvent émettre simultanément et le signal est brouillé.

- On parle alors de **collision**.
- Problème : comment résoudre ces collisions ?

L'approche CSMA/CD (Ethernet)

- **CS** = Carrier Sense = écoute de la porteuse (le signal)

  Avant de transmettre, on écoute le bus pour savoir si des données arrivent.
- **MA** = Multiple Access

  liaison partagée par toutes les machines : le bus
- **CD** = Collision Detection

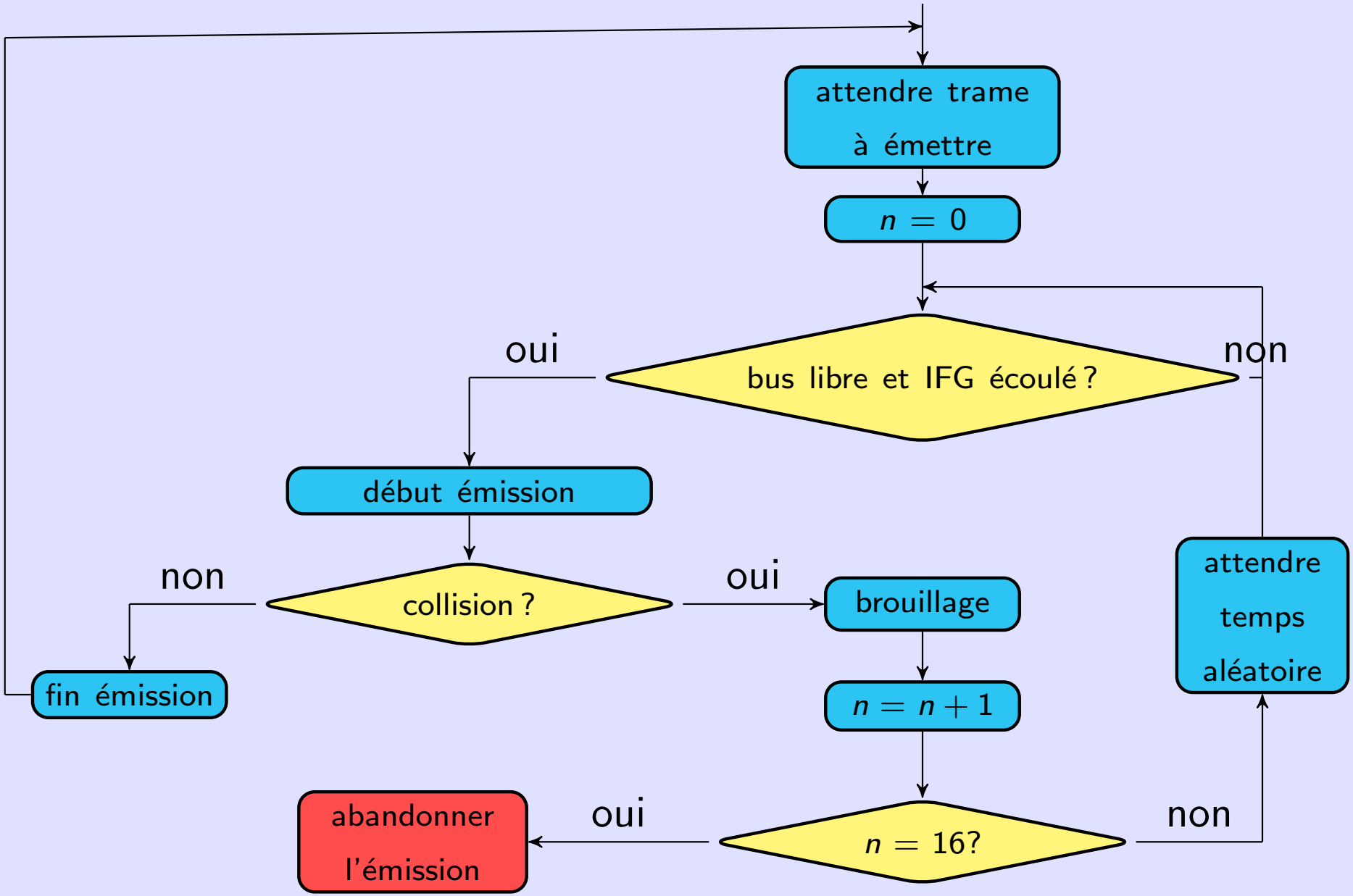
  On ne cherche pas à éviter les collisions, on les détecte puis les corrige.
  collision détectée en cas d'émission et de réception simultanées

L'approche CSMA/CA (Réseaux sans fil)

- **CA** = Collision Avoidance (évitement des collisions)

  On évite les collisions ($\Rightarrow$ pas de mécanisme de correction).

Principe du CSMA/CD



IFG

- **IFG** = Inter Frame Gap (Délai inter-trame)
- délai minimum de 9.6μs entre deux émissions ou réceptions
- utilité : marquer une séparation entre les trames et permettre aux interfaces de se préparer à une nouvelle trame

Séquence de brouillage

- brouillage = envoi d'une séquence de 4 octets incohérents
- utilité : s'assurer que la collision est détectée par toutes les machines du réseau

Délai aléatoire après collision

- Pour que les interfaces ne tentent pas de retransmettre au même moment, elles attendent un temps tiré aléatoirement.

Protocole de réception

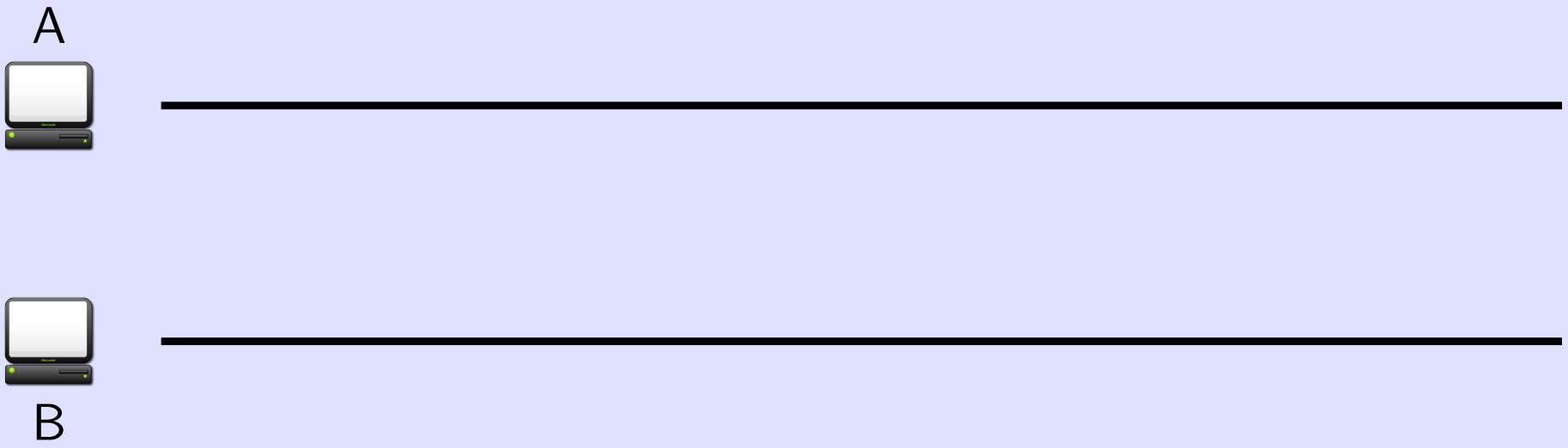
Protocole de réception exécuté par une interface Ethernet :

1. écouter sur le bus et attendre qu'une trame arrive
2. quand une trame est arrivée, vérifier :
   - 2.1 qu'elle a une longueur $\geq$ 72 octets

     (trame de moins de 72 octets = trame ayant subi une collision)
   - 2.2 et qu'elle est correcte (reste de la division des champs de la trame par le polynôme générateur = FCS)

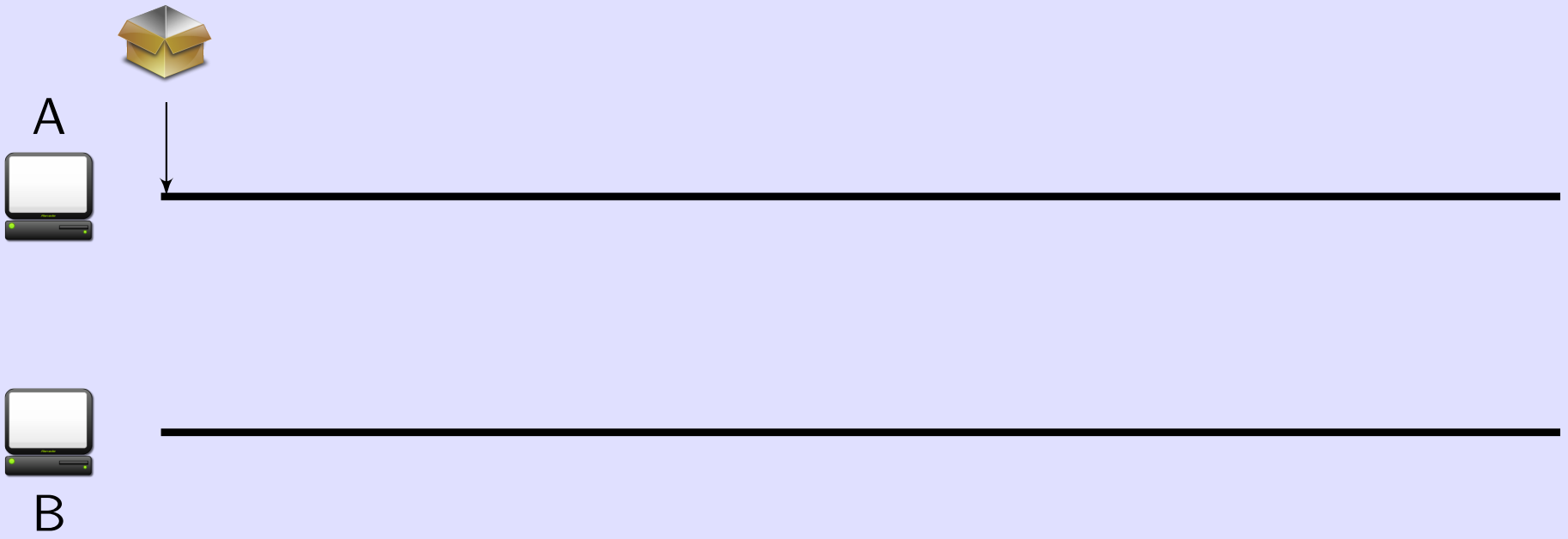
     (en cas de collision le brouillage garantit que la trame sera incorrecte)
3. si la trame est correcte, regarder son champ DA :
   - 3.1 si DA = mon adresse ou `FF:FF:FF:FF:FF:FF` $\Rightarrow$ délivrer le champ de Données à la couche supérieure (au système d'exploitation dans le cas d'un paquet IP)
   - 3.2 sinon, la trame ne m'est pas destinée $\Rightarrow$ l'ignorer

Exemple



d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

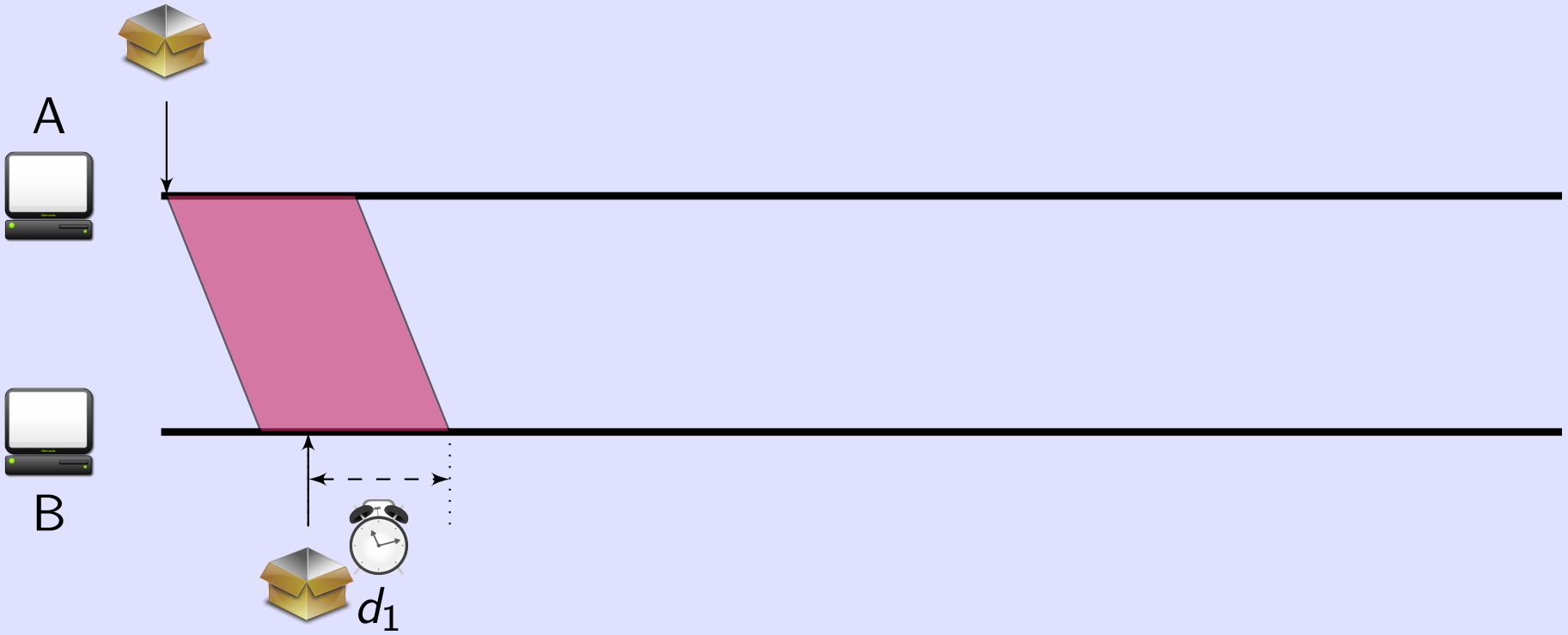
Exemple

d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple

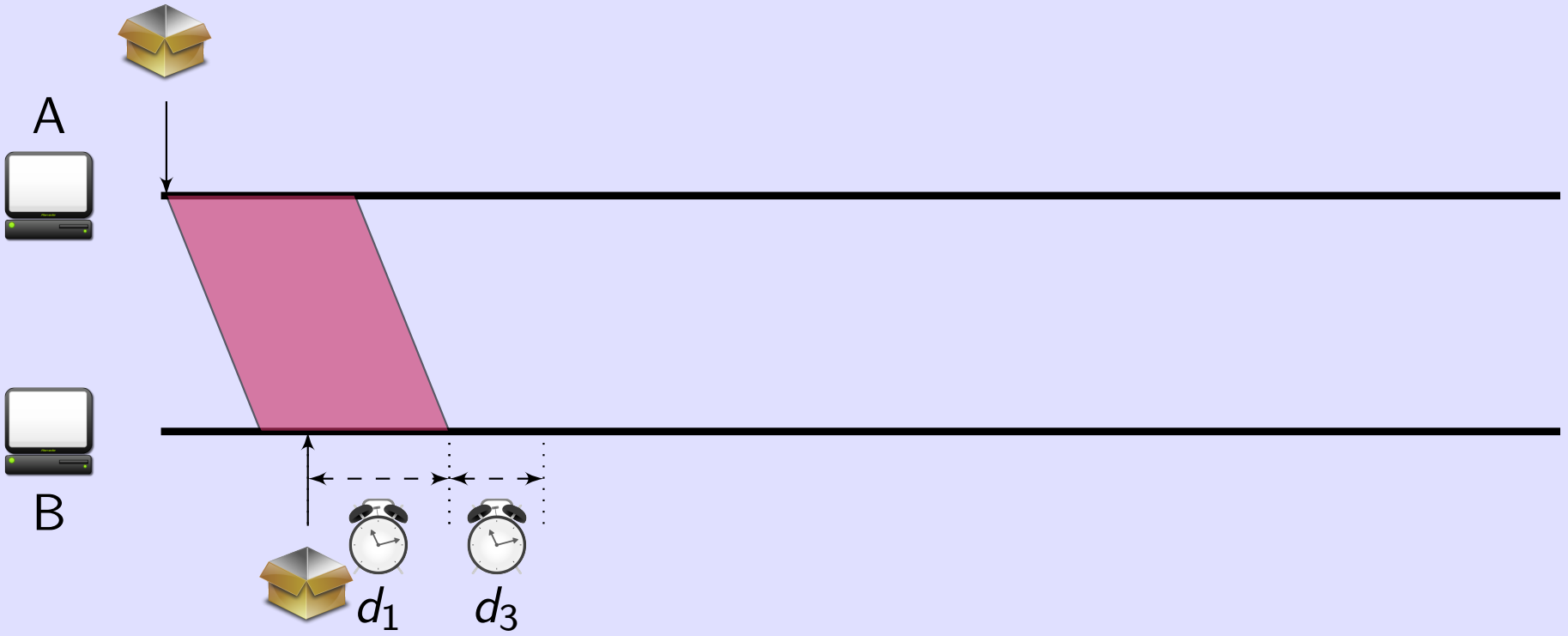
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



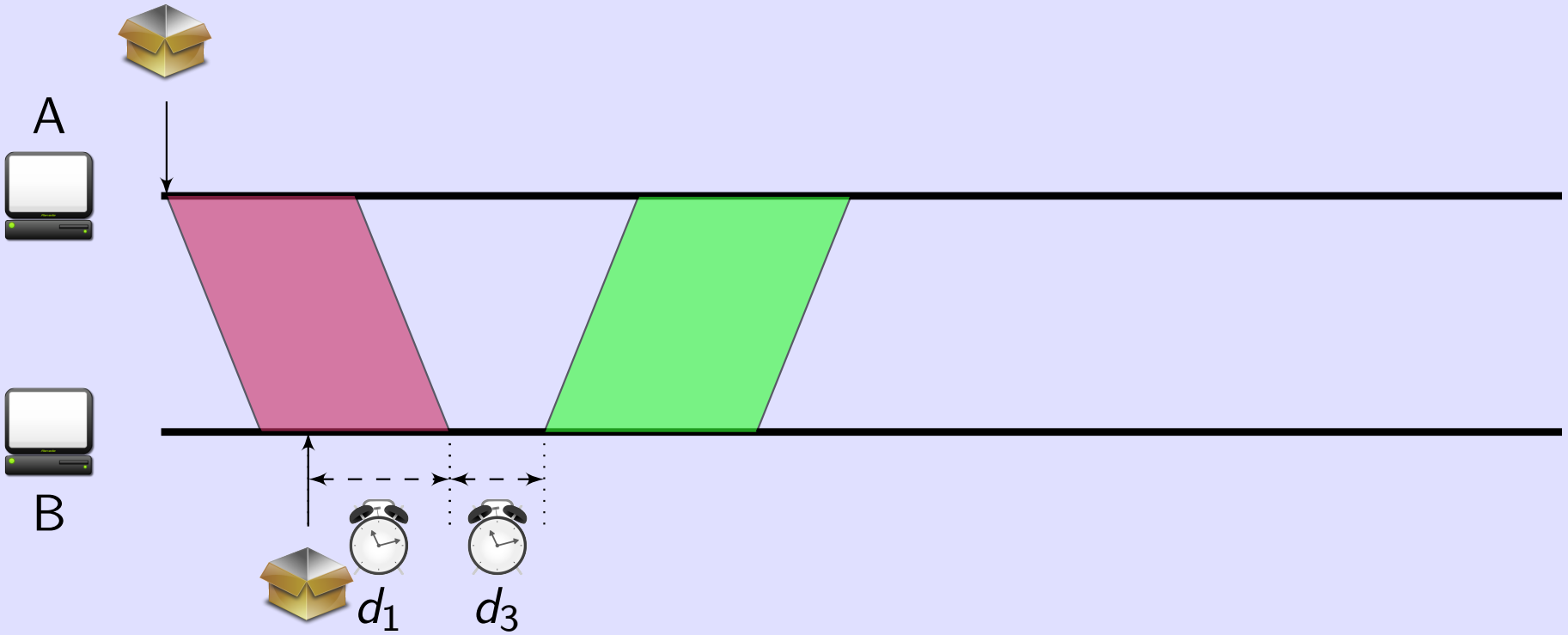
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



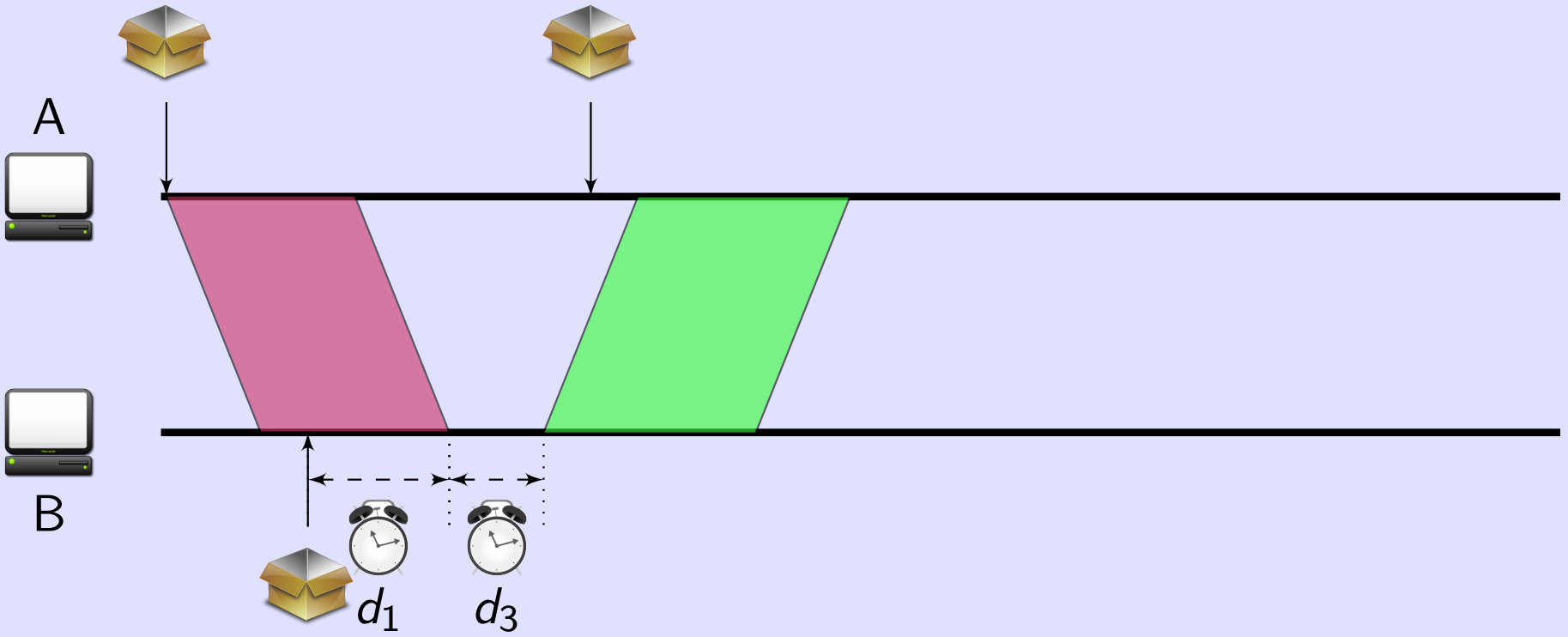
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



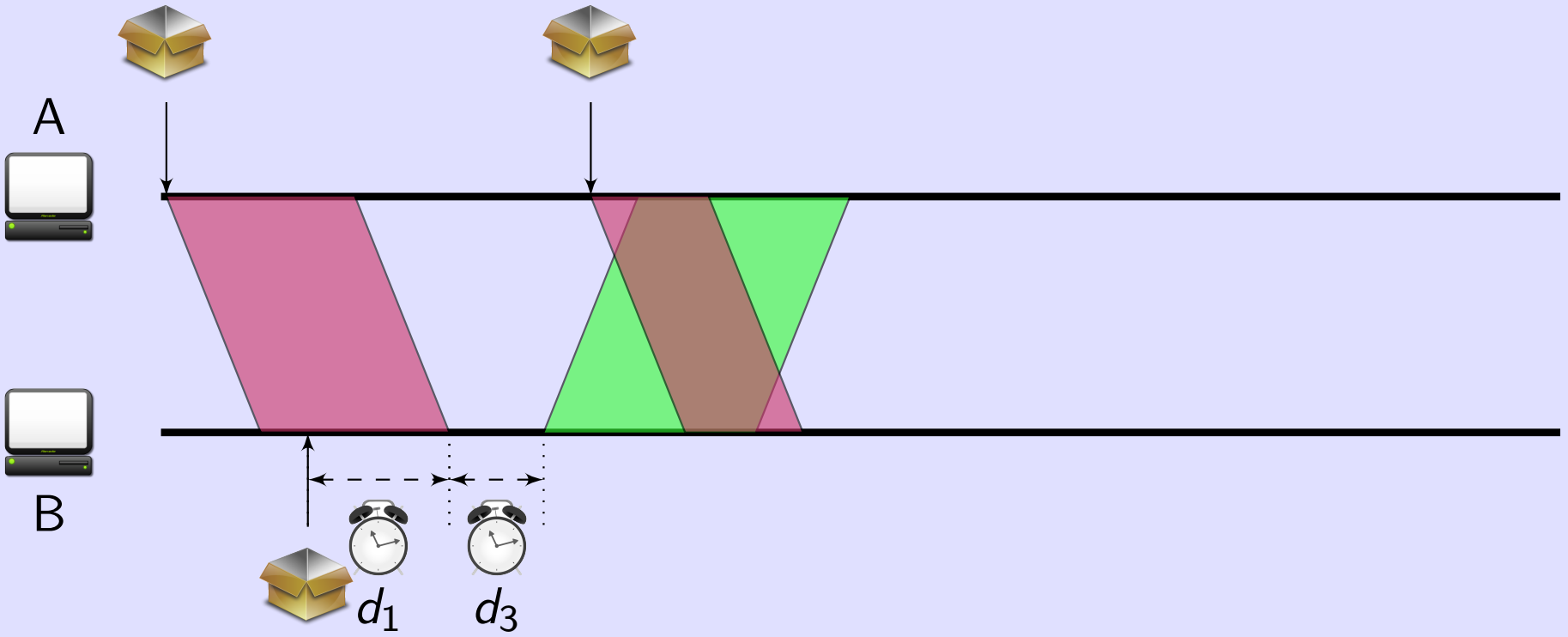
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



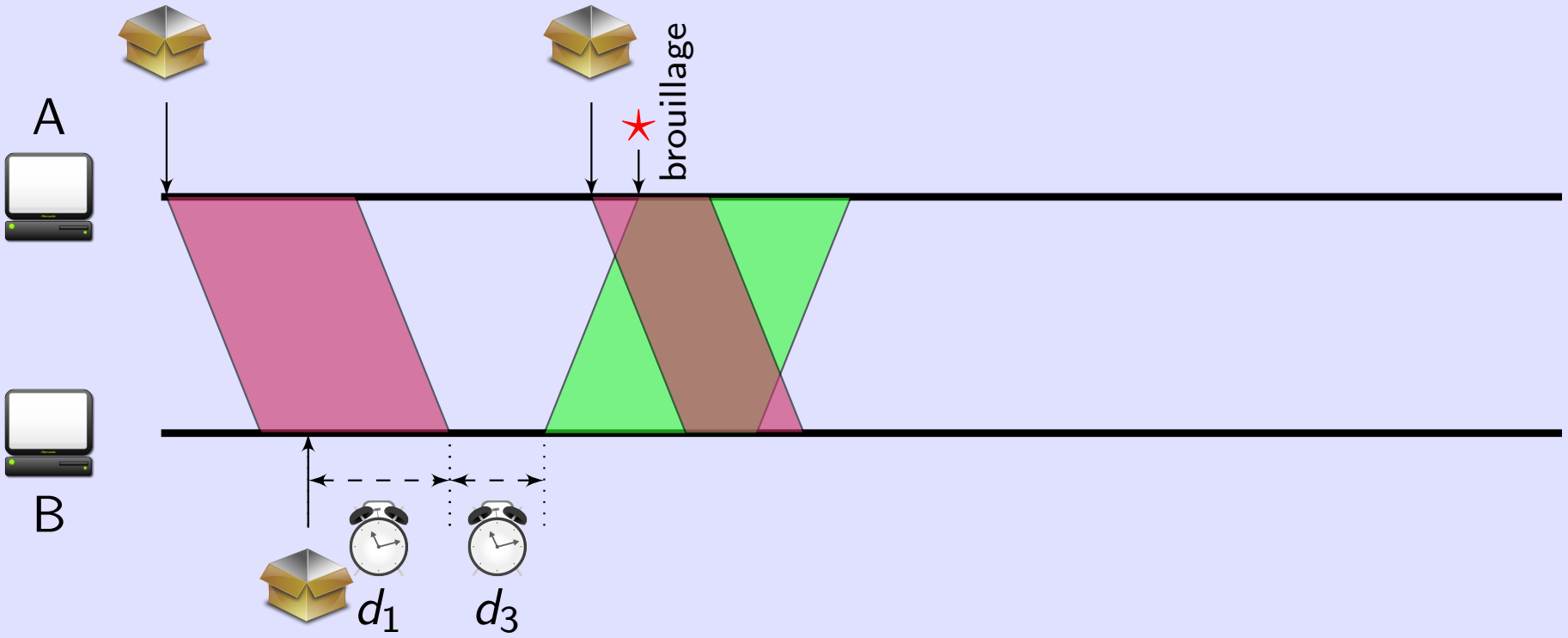
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



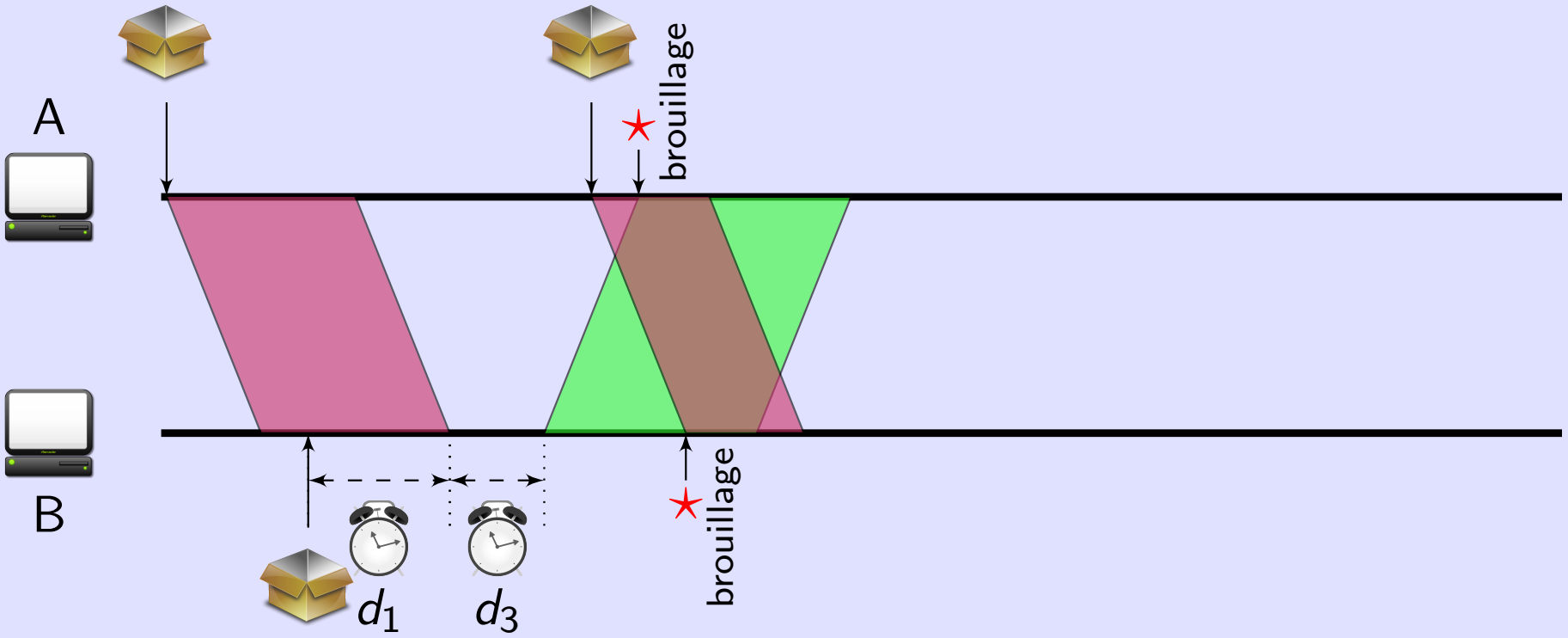
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
$\star$	détection d'une collision

Exemple



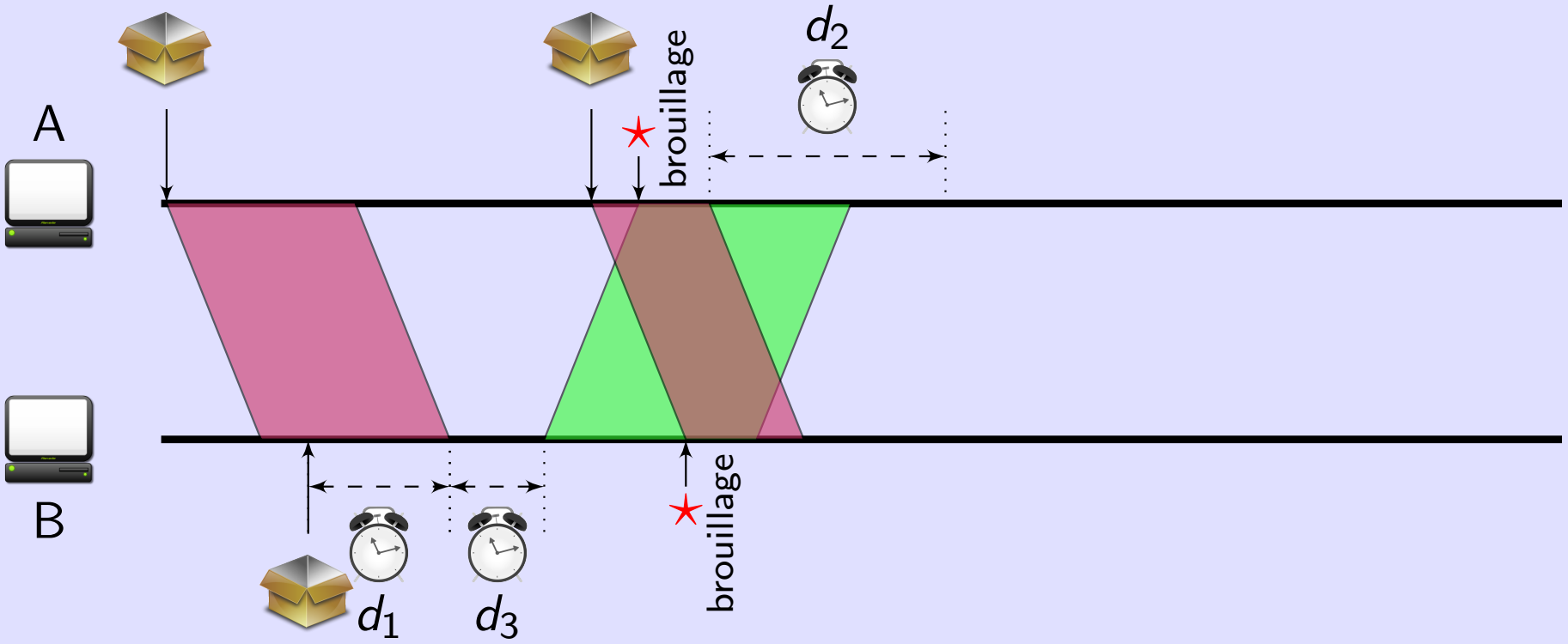
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
⋆	détection d'une collision

Exemple



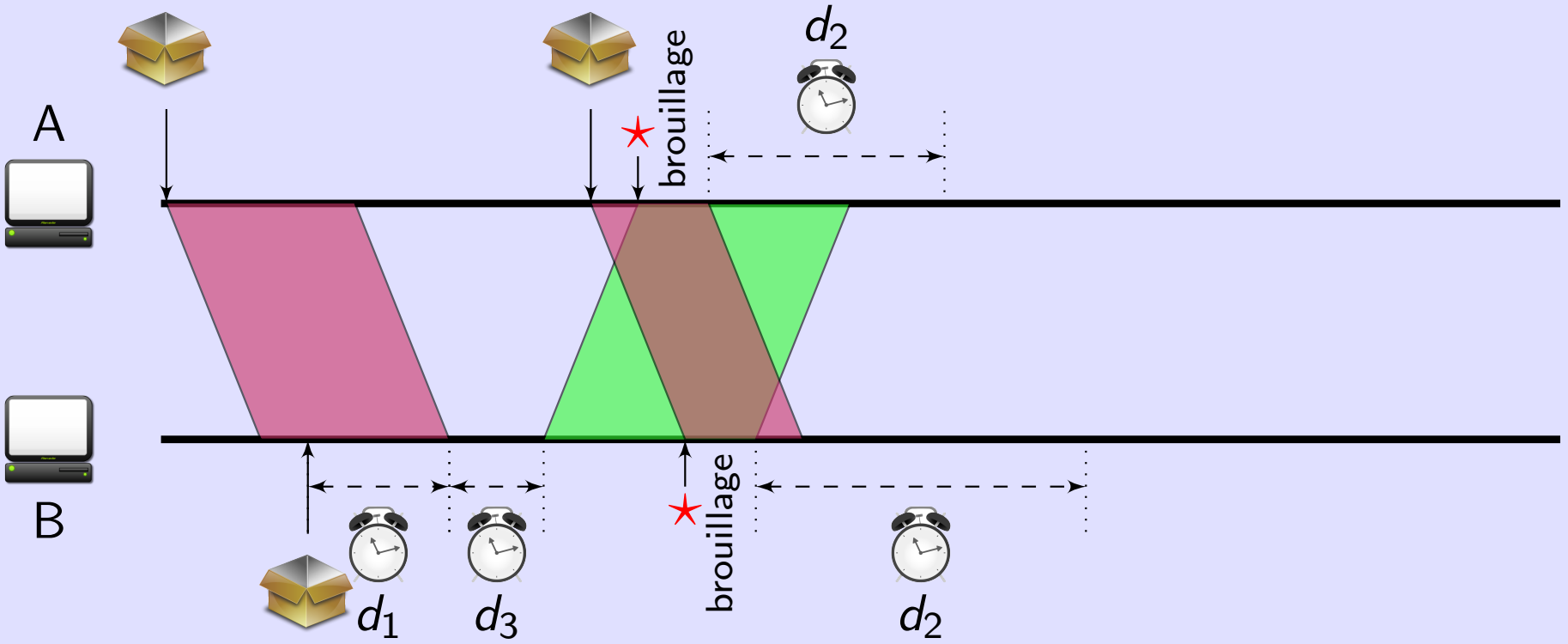
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

Exemple



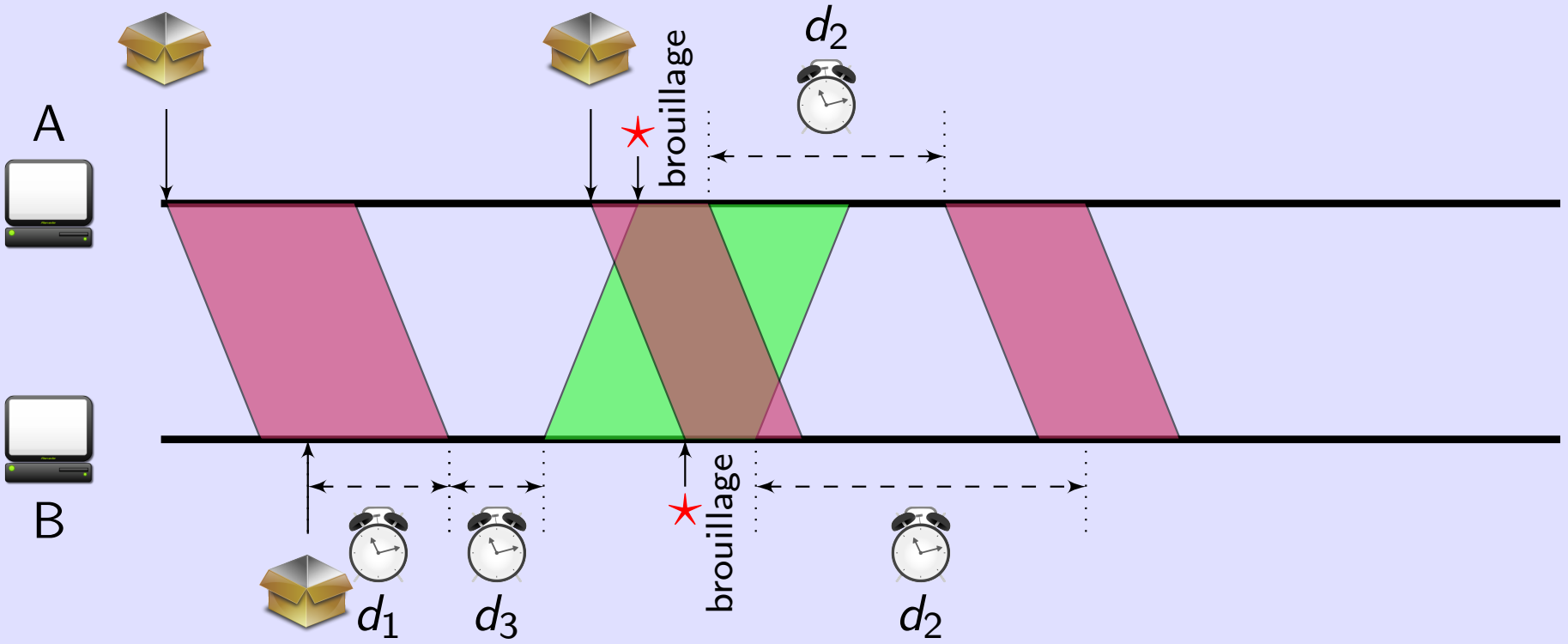
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

Exemple



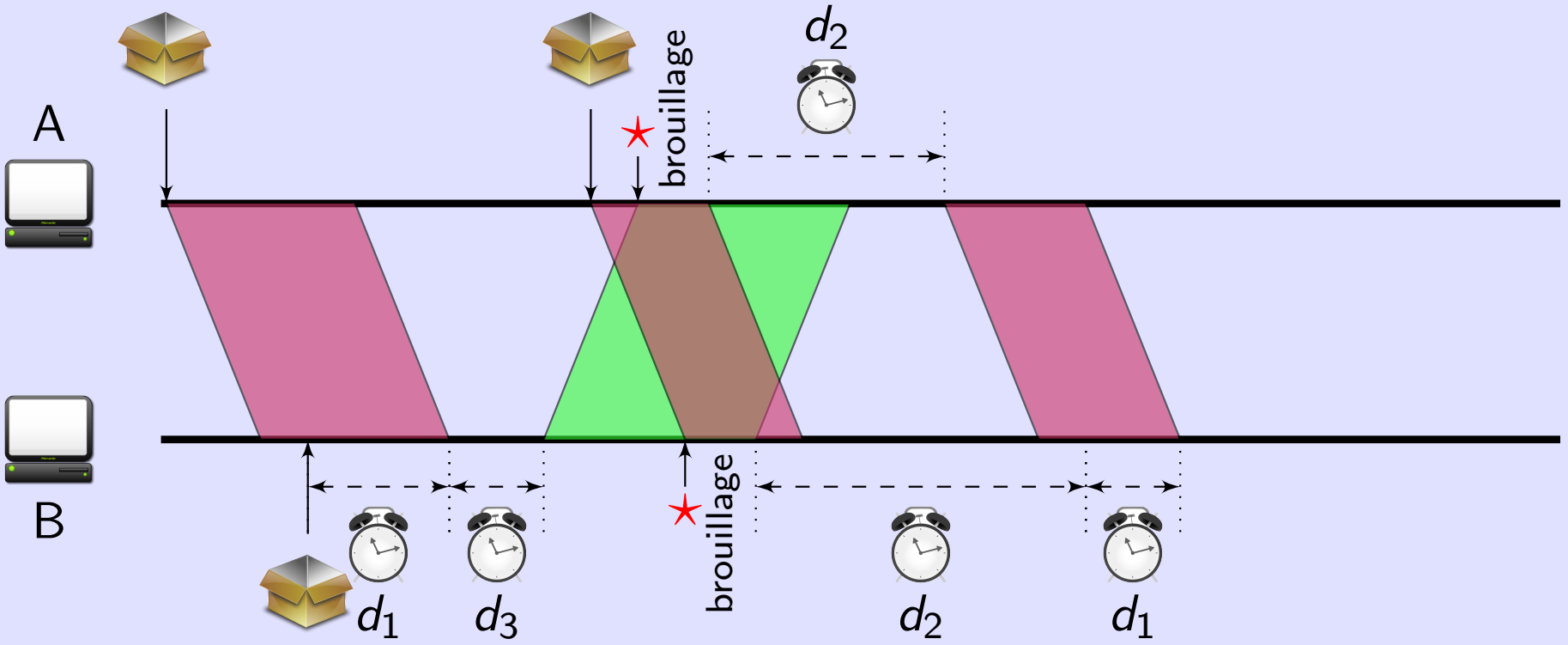
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

Exemple



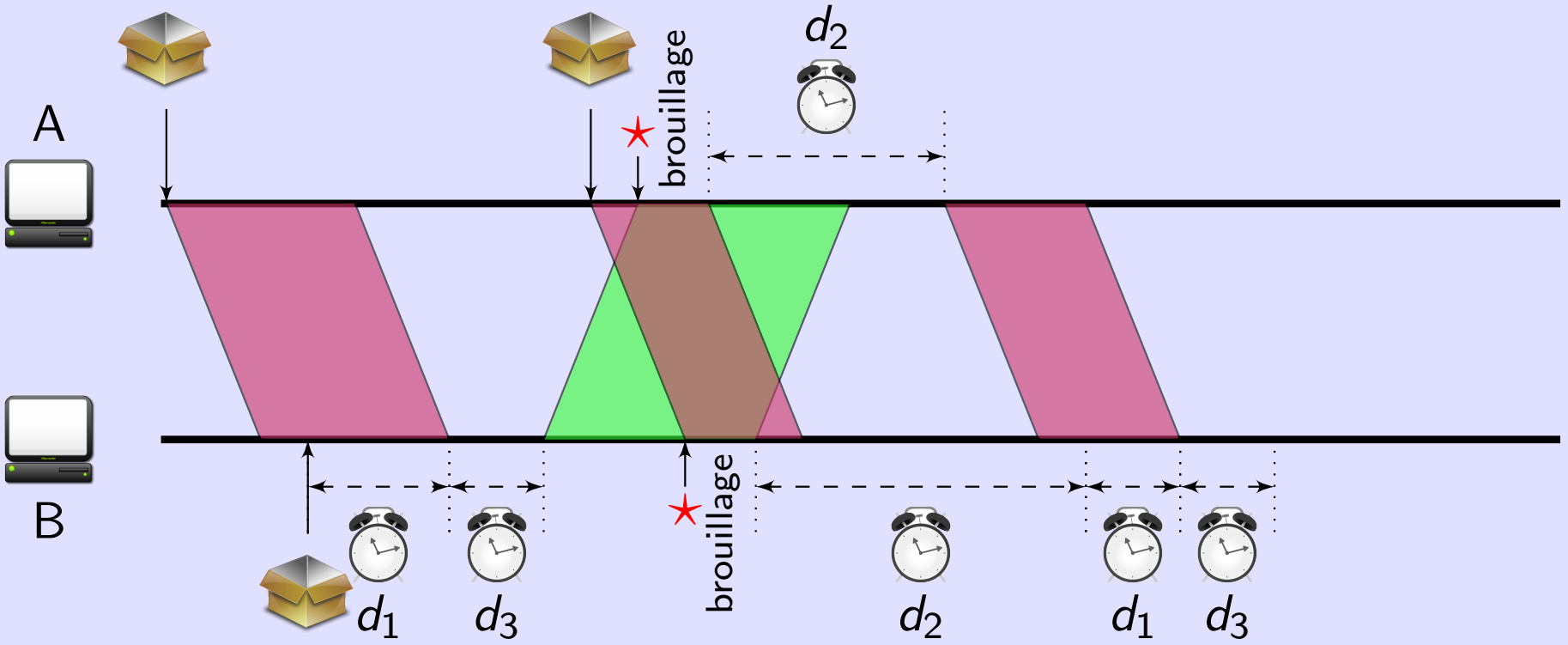
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

Exemple



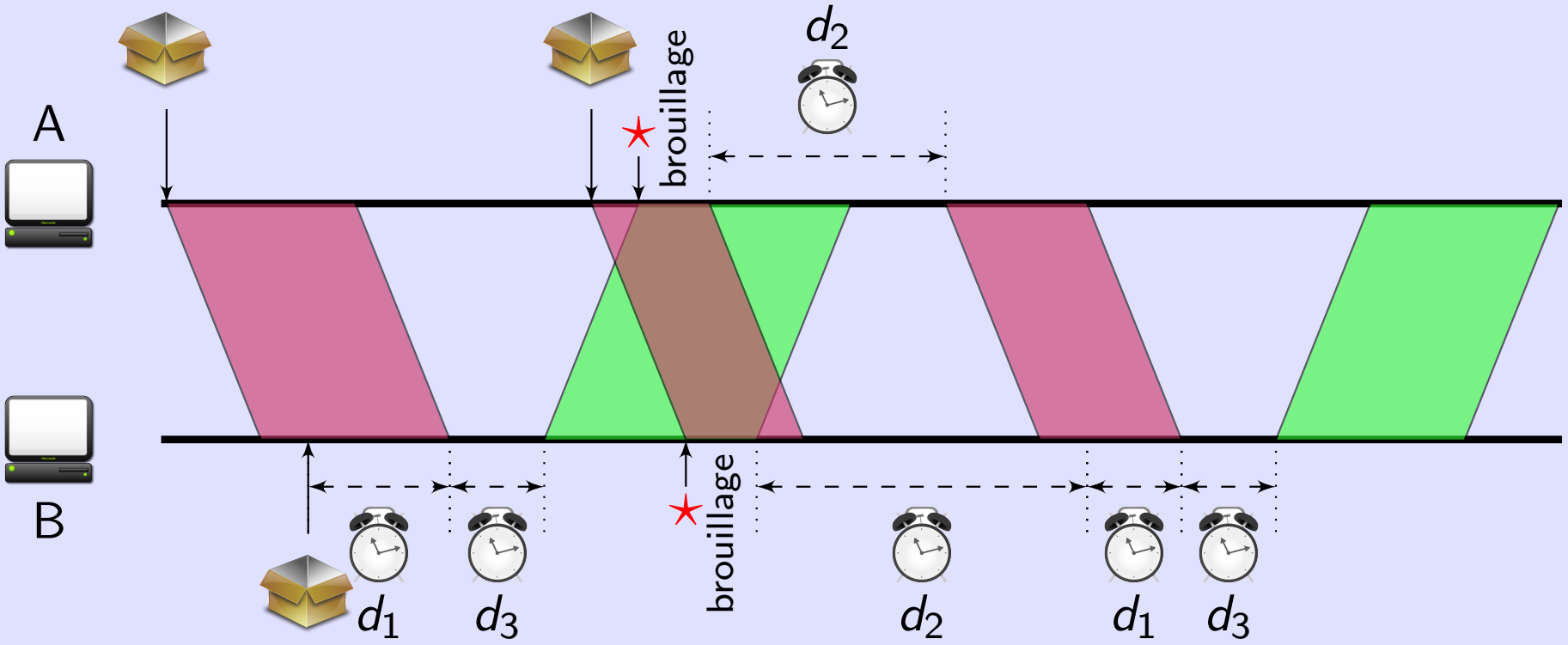
d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

Exemple



d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

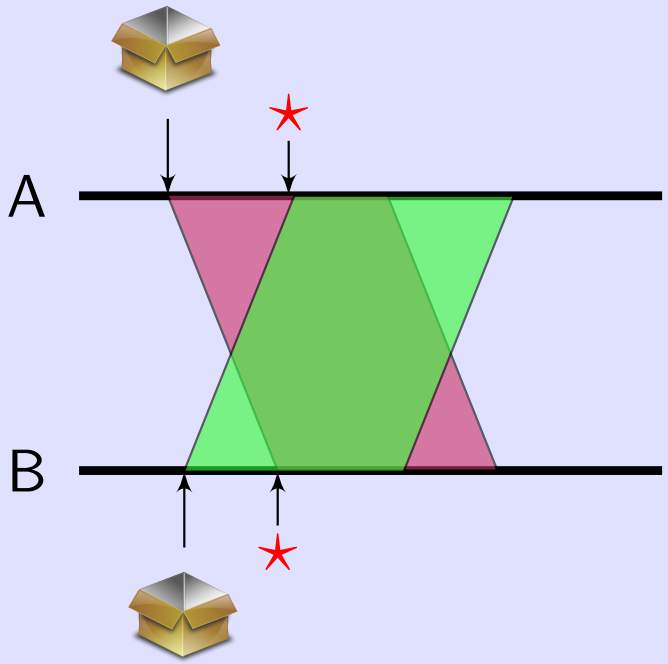
Exemple



d_1	attente de libération du support
d_2	attente d'une durée aléatoire
d_3	IFG = délai d'attente inter-trame
★	détection d'une collision

Scénarios de collision

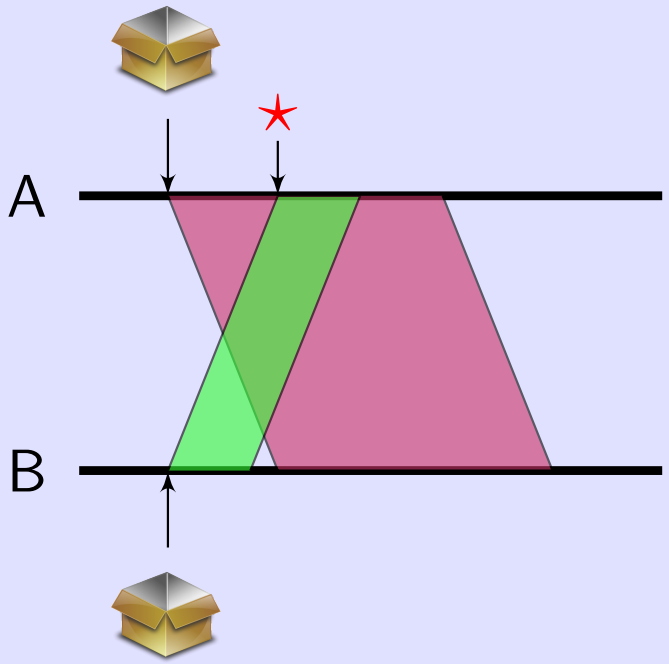
Scénario 1



- collision détectée par A et B
- A et B retentront leurs émissions

⇒ collision résolue

Scénario 2



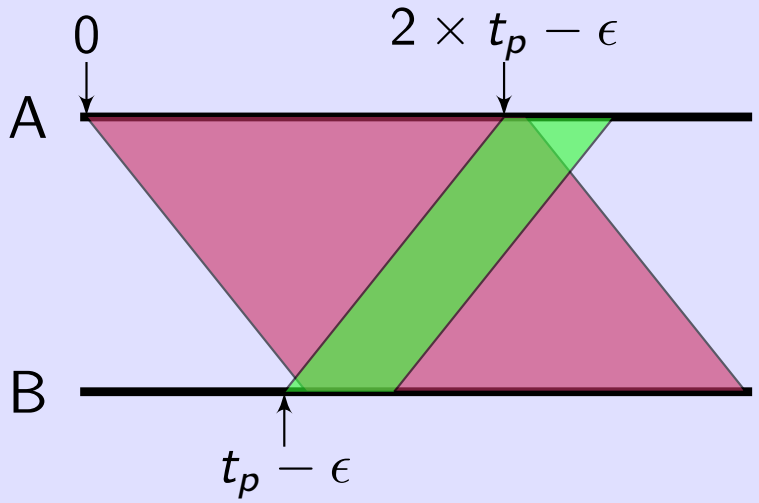
- collision détectée par A mais pas par B
- A retentra son émission mais pas B

⇒ **collision non résolue**

Condition suffisante pour détecter les collisions

- Supposons que :
  - A débute sa transmission à $t = 0$.
  - B débute sa transmission juste avant la réception du premier bit de la trame de A à $t_p - \epsilon$ (t_p = temps de propagation).
- A reçoit donc le premier bit de la trame de B à $2 \times t_p - \epsilon$.

$\Rightarrow$ Si A transmet pendant $2 \times t_p$ secondes ($> 2 \times t_p - \epsilon$), il détectera nécessairement la collision.



Taille minimale des trames

- On a vu que la transmission doit durer au moins $2 \times t_p$.
- Cela doit être vrai pour les deux machines les plus éloignées du réseau.
- Dans ce cas, on note $2 \times t_p =$ **DAR** (Délai Aller-Retour)
- temps de transmission minimal $\Leftrightarrow$ taille de trame minimale
- Cette taille T_{min} dépend :
  - du DAR ;
  - et du débit D.
- en Ethernet, $\mathbf{T_{min} = 72}$ octets choisi expérimentalement avec
  - $DAR = 51,2\mu s$ (DAR calculé pour un bus de 2500 mètres)
  - $D = 10$ Mbit/s

Bourrage des trames

- Pour que la trame ait une taille suffisante, l'émetteur rajoute des octets de **bourrage** si il n'y a pas assez de données à transmettre.
- Algorithme de bourrage de trame lors d'une transmission :
  - si le champ de données fait moins de 46 octets
    - on rajoute à la suite des données des octets 0 pour atteindre les 46 octets
  - sinon
    - on laisse le champ de données tel quel

Attente après une collision

- Si une station détecte une collision, elle attend pendant une durée décidée aléatoirement.
- Elle retente ensuite son émission (dès que le bus est libre).
- On utilise l'algorithme **BEB** (Binary Exponential Backoff) :
  - après la $n^{\text{ème}}$ collision sur une même trame on attend un temps $k \times DAR$ avant de retenter avec
    - k choisi aléatoirement dans l'intervalle $[0, 2^{\min(10,n)} - 1]$
    - $DAR = 51,2\mu s$
  - si $n = 16$ on abandonne la transmission
- principe du BEB :
  1. En augmentant progressivement le délai d'attente on
     1.1 décongestionne le réseau
     1.2 augmente les chances qu'un machine puisse transmettre sans collision
  2. Si peu de machines souhaitent émettre les délais d'attente restent courts.

Codification des supports

- On note un support sous la forme `DS-C`
  - `D` = débit en Mbit/s
  - `S` = codage du signal (voir module M3106)
    - `Base` = bande de base : signaux carrés
    - ou `Broad` = large bande : signaux sinusoïdaux
  - `C` = informations sur le câblage à utiliser
- Exemples :
  - `100Base-T` = débit de 100 Mbit/s sur paire torsadée
  - `1000Base-F` = débit de 1 Git/s sur fibre optique

La première famille de réseaux Ethernet : 10Base5

- 10Base5 = 10Mbit/s, codage en bande de base, et câbles coaxiaux de 500 mètres (au max.)
- On peut utiliser jusqu'à 4 **répéteurs** pour arriver à 2 500 mètres.
- Le bus est un **câble coaxial épais** (généralement jaune).

$\Rightarrow$ topologie physique et logique en bus

- Tous les 2,5 m. sur le câble : emplacement de raccordement d'une machine.

- Ce qui change :
  - le débit : 10Mbit/s (années 1980) → 100 Mbit/s (années 1990–2000) → 1–10 Gbit/s (années 2010)
  - le codage des signaux
  - le matériel utilisé (câble coaxial jaune → hubs → switchs)
  - les distances de câble autorisées
- Ce qui est commun à toutes les versions :
  - la structure des trames (préambule, SFD, DA, SA, ...)
  - le bourrage des trames
  - le protocole d'émission/réception (avec CSMA/CD et BEB)

Compatibilité avec les versions antérieures

- Avec les équipements modernes (type switch) il n'y a plus de collision. Ces équipements fonctionnent en full duplex.
- Pourquoi garder tout ce qui est relatif à la détection/résolution des collisions (CSMA/CD, bourrage des trames) ?
- Pour garder la compatibilité avec les équipements plus anciens (type hub) qui fonctionnent en half duplex.
- Ex : un switch doit pouvoir résoudre les collisions s'il est relié à un hub.

Principe et objectifs d'ARP

- Le système désigne les machines de son LAN par leurs adresses IP.
- Or pour transmettre une trame, l'adresse IP est inutile : on a besoin de connaître l'adresse MAC du destinataire.
- Exemple :
  - Je veux envoyer un paquet à la machine 1.2.3.4.
  - La table de routage me dit que ce paquet doit être envoyé au routeur 5.6.7.8.
  - Je dois donc encapsuler le paquet dans une trame destinée à 5.6.7.8.
    Problème : quelle est l'adresse MAC de 5.6.7.8 ?

$\Rightarrow$ besoin d'un mécanisme de correspondance adresse MAC $\leftrightarrow$ adresse IP

- Solution : le protocole **ARP** (Address Resolution Protocol)
- Quand utilise-t-on ARP ? Dès qu'on veut envoyer une trame à une machine dont on connaît l'adresse IP mais pas l'adresse MAC.
- Un message ARP est directement encapsulé dans une trame Ethernet :

Fonctionnement d'ARP

2 types de messages ARP : la requête et la réponse

La requête

- placée dans une trame diffusée à toutes les interfaces du LAN (adresse MAC destinataire dans l'en-tête Ethernet = `FF:FF:FF:FF:FF:FF`)
- contient l'adresse IP demandée
- signification : *quelle est l'adresse MAC de l'interface dont l'adresse IP est A.B.C.D ?*

La réponse

- transmise par la machine qui reconnaît son adresse IP dans la requête à la machine ayant émis la requête
- contient son adresse MAC et son adresse IP
- signification : *mon adresse IP est bien A.B.C.D et mon adresse physique est `AA:BB:CC:DD:EE:FF`*

La table ARP

- Tout envoi de paquet devrait théoriquement être précédé d'un échange ARP pour que l'émetteur connaisse l'adresse physique du destinataire.

$\Rightarrow$ génère beaucoup de trafic sur le réseau local

- Solution : chaque machine maintient une table ARP d'associations

(adresse IP, adresse MAC)

- Avant d'envoyer une requête ARP, une machine vérifie si l'adresse MAC recherchée n'est pas déjà présente dans la table.
- À la réception d'une réponse ARP, l'adresse MAC est ajoutée à la table.
- Pour voir la table ARP :

```
$ arp -n
Adresse          Type   AdresseMat          Indicateurs Iface
192.168.44.60    ether  68:5b:35:ab:4c:2d   C           eth0
192.168.44.254   ether  00:02:b3:bb:c0:29   C           eth0
```

(Indicateur C = entrée Complète : on a les deux adresses pour l'entrée.)

Caractéristiques des réseaux Token Ring

- **Token Ring** = anneau à jeton
- conçu par IBM en 1982
- normalisé par l'IEEE en 1985 : norme IEEE 802.5
- débit de 4, 16 ou 100 Mbit/s
- utilise le mode d'adressage MAC
- topologie logique en **anneau**/physique en **anneau** ou en **étoile**
- anneau **unidirectionnel** $\Leftrightarrow$ la circulation est à sens unique :
  - une machine reçoit des trames uniquement de la station précédente dans l'anneau
  - et elle envoie toujours à la station suivante

Principe général des anneaux à jeton

- Une trame spéciale, appelée **jeton**, circule sur l'anneau.
- Ce jeton est un droit à émettre.
- Il peut être pris par une machine ayant des données à émettre.
- A tout moment, **une seule** machine du réseau peut émettre.

$\Rightarrow$ 2 types de trames :

- Les trames d'**information** qui contiennent des données de la couche supérieure.
- Les jetons qui ne contiennent pas de données.

Réception du jeton par la machine m

- m passe le jeton à sa voisine si elle n'a pas d'information à transmettre
- sinon elle retire le jeton de l'anneau et émet sa trame d'information

Réception d'une trame d'information par la machine m

- le champ de données de la trame est passé à la couche supérieure si m est le destinataire
- si cette trame appartient à m elle la retire de l'anneau et
  - émet une nouvelle trame d'information si elle en a à émettre ;
  - ou passe le jeton à la machine suivante sinon.
- sinon elle retransmet la trame d'information à la machine suivante

$\Rightarrow$ Une trame d'information transmise par une machine m fera toujours le tour de l'anneau avant d'être retirée de la circulation par m.

Comparaison Ethernet-Token Ring

Ethernet (version initiale)

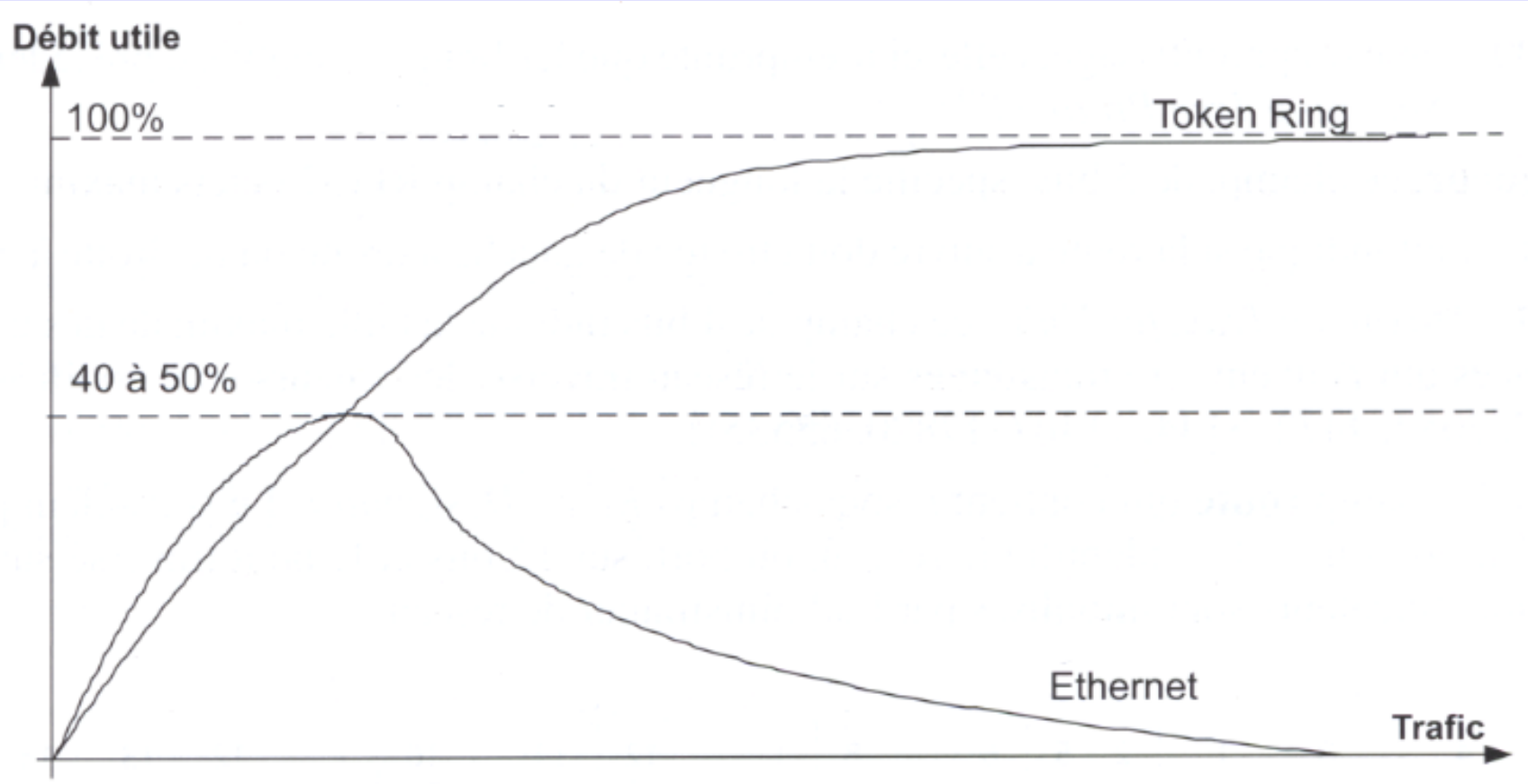
- excellentes performances s'il y a peu de trafic
- mais le débit utile s'effondre avec l'augmentation du trafic
  (augmentation du nombre de collisions)
- aucune garantie sur le temps d'attente avant l'émission (correcte) d'une trame

Token Ring

- système de priorités inexistant dans Ethernet
- débit optimal plus faible qu'avec Ethernet :
  - Une machine doit attendre le jeton avant de pouvoir émettre.
  - Le temps d'attente dépend du nombre de machines sur l'anneau et de la quantité d'informations qu'elles veulent émettre.
- Mais on a un débit garanti en fonction du nombre de machines.
  - Une machine est sûre de recevoir le jeton au bout d'un certain temps.

Courbe de performances

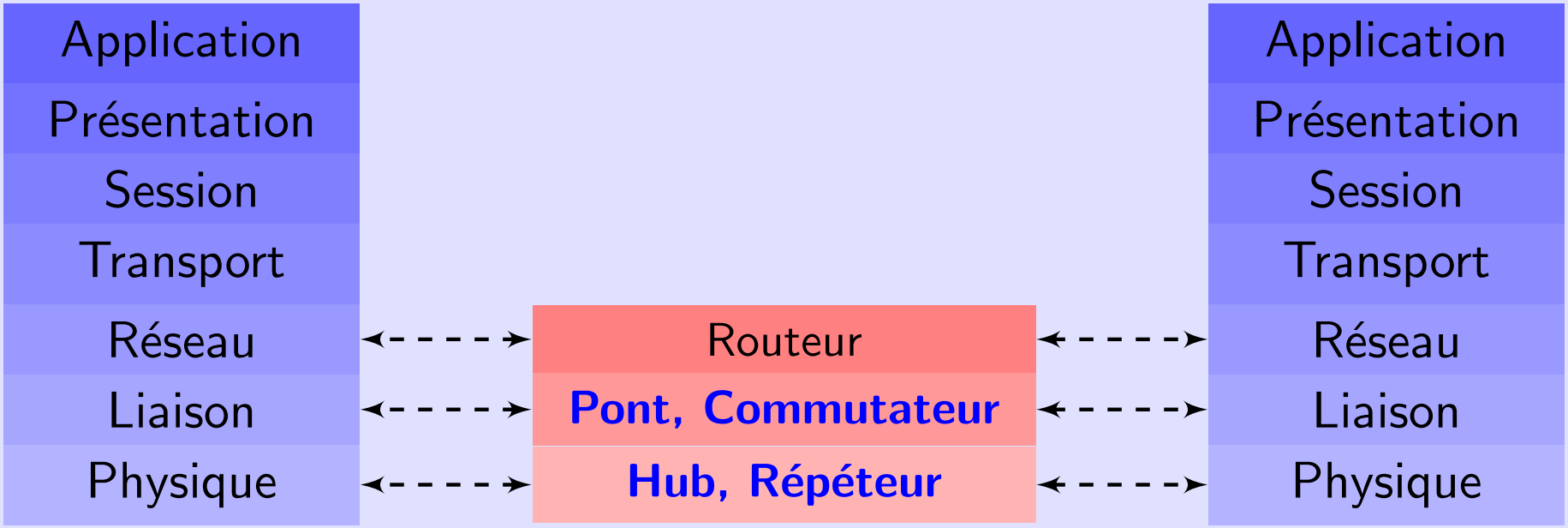
Débit utile en fonction du trafic avec Ethernet et Token Ring.



Plan

Equipements d'interconnexion

- Comment interconnecter des réseaux ?
- Plusieurs possibilités selon le niveau auquel on se situe.

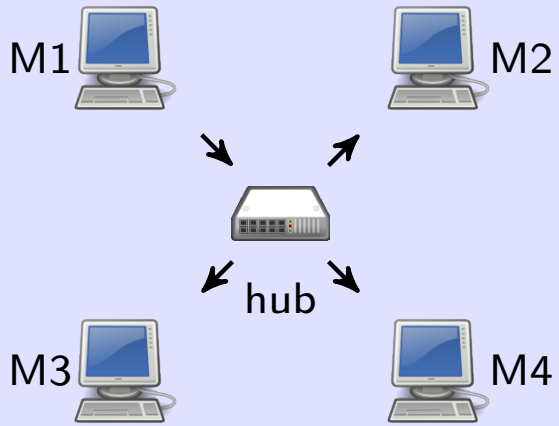


Dans ce cours : étude des ponts, commutateurs, hubs et répéteurs.

Au niveau physique

- Le **répéteur**
  - C'est un boîtier avec deux ports.
  - Il retransmet sur un port les signaux reçus sur l'autre port.

  $\Rightarrow$ régénère les signaux reçus qui se dégradent sur des longues distances
- Le **hub**
  - Il a plusieurs ports sur lesquels on peut brancher des cables Ethernet.
  - Un signal reçu sur un port est reproduit sur tous les autres ports.



- Ces équipements se situent au niveau physique car ils n'interprètent pas les bits reçus : il se contentent de les reproduire.

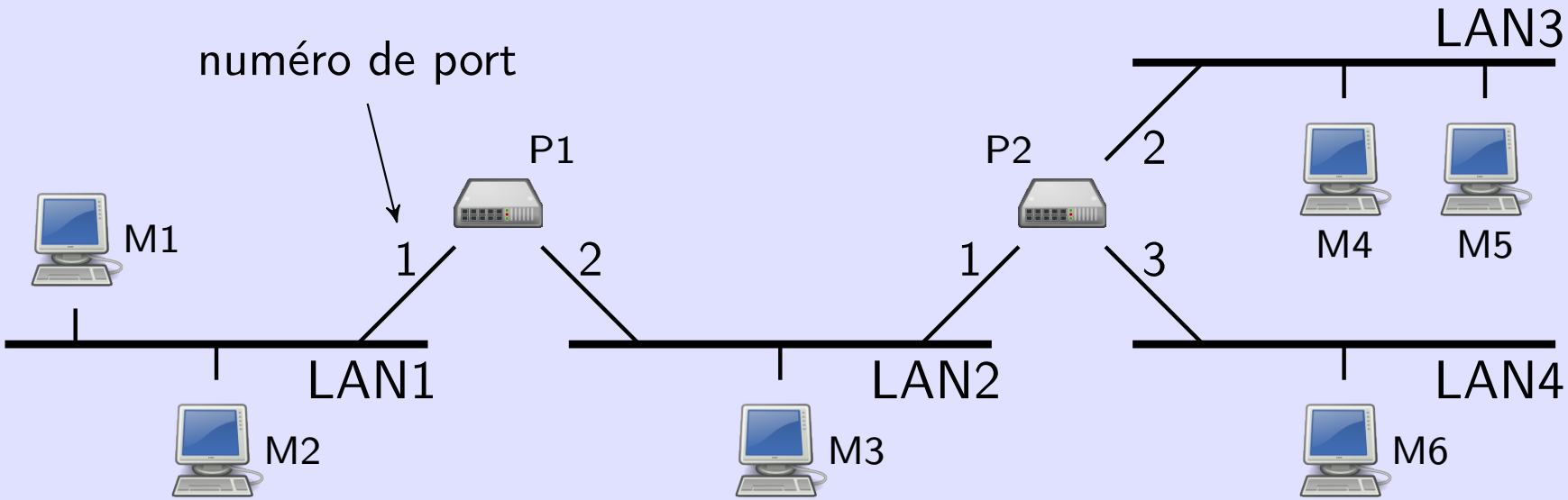
Différences entre ponts et commutateurs

- **Pont** est un terme général pour désigner un équipement qui relie
  - différents LANs
  - qui peuvent être de différents types (Ethernet, Wifi, ...).
- Un **commutateur** (ou **switch**) relie des LANs ou des équipements de même type.
- Dans ce cours, on s'intéresse uniquement aux réseaux Ethernet.

  donc : switch $\Leftrightarrow$ pont

Rôle des ponts

- Un pont Ethernet connecte des réseaux Ethernet.
- Exemple : 4 bus Ethernet connectés par 2 ponts P1 et P2.



- Rôle du pont :
  - attendre une trame
  - déterminer le(s) LAN(s) sur lequel(s) la trame doit être redirigée en utilisant une **table de commutation**
  - retransmettre la trame sur ce(s) LAN(s)
- **commutation** = action de mettre en relation deux correspondants (l'émetteur et le récepteur de la trame)

Caractéristiques des ponts

- Le pont se situe au niveau **2 - Liaison** du modèle OSI car :
  - Il a une adresse physique (MAC).
  - Il connaît les adresses MAC des machines qu'il relie.
  - Il exécute le protocole CSMA/CD (résolution des collisions).
  - Il comprend et interprète les trames qu'il reçoit.
- Un pont est un hub intelligent car :
  - il sait comment joindre la machine destinataire d'une trame
  - et il retransmet la trame uniquement sur le bon LAN
- Un pont est dit **transparent** :
  - Les machines voient tous les LANs connectés comme un unique LAN.
  - Les machines n'ont pas connaissance de l'existence des ponts.
- Un pont n'a pas de "carte" globale du réseau : l'acheminement d'une trame à son destinataire final se fait de **proche en proche**.

La table de commutation

- La table de commutation contient une liste d'enregistrements

  (adresse, port)

- Un enregistrement (d,p) est interprété de la façon suivante :

  Je sais que toute trame destinée à la machine dont l'adresse MAC est d doit être retransmise sur le port p.

- 2 possibilités à la réception sur un port p d'une trame destinée à d :
  1. d n'est pas dans la table ou $d =$ `FF:FF:FF:FF:FF:FF`
     $\Rightarrow$ le pont retransmet la trame sur tous ses ports à l'exception de p
  2. la table contient un enregistrement (d,p')
     $\Rightarrow$ le pont retransmet la trame sur le port p' uniquement (et si $p \neq p'$)

La table de commutation — Exemple

4 réseaux Ethernet connectés avec 2 ponts

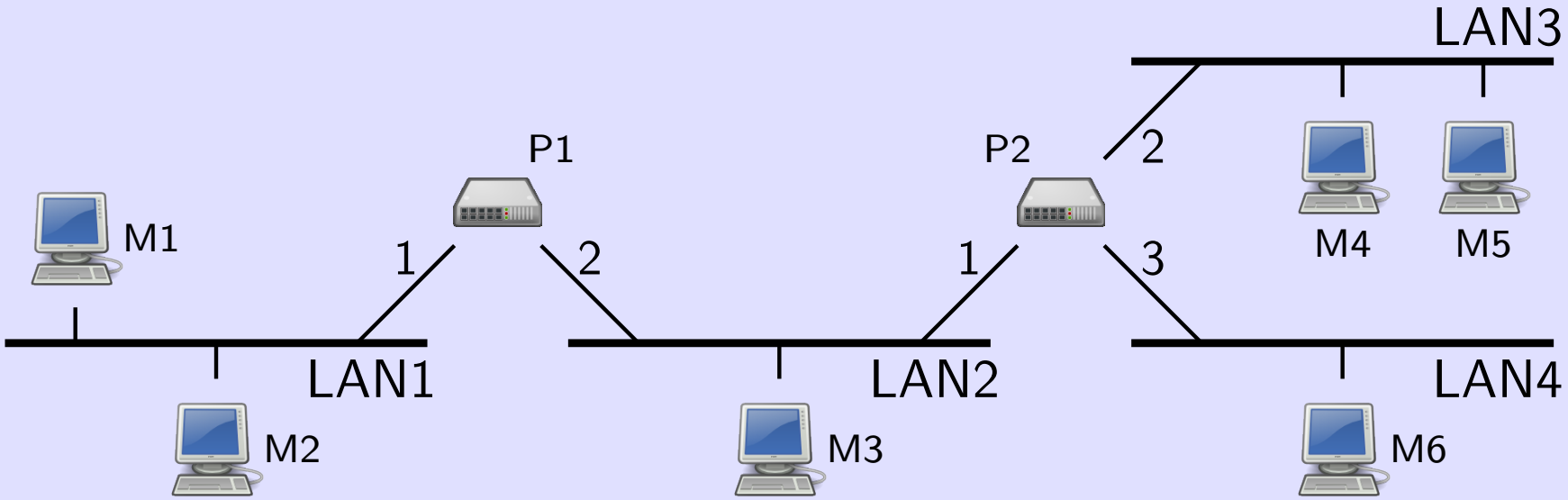


Table de P1	
adresse MAC	port
M1	1
M2	1
M3	2
M4	2
M5	2
M6	2

Table de P2	
adresse MAC	port
M1	1
M2	1
M3	1
M4	2
M5	2
M6	3

Remplissage manuel (ou statique)

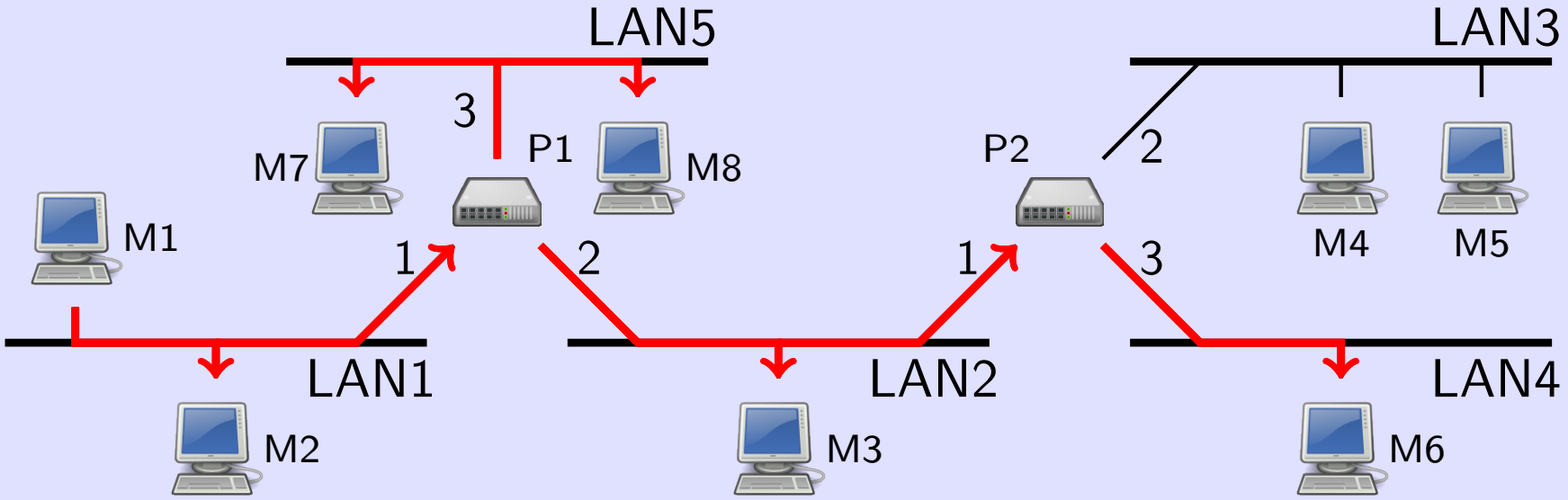
- L'administrateur du réseau crée un enregistrement par machine à l'installation du réseau.
- Plusieurs inconvénients :
  - fastidieux
  - erreur(s) possible(s) de l'administrateur
  - nécessité de modifier les tables quand on déplace une machine

Remplissage automatique (ou dynamique)

- Les tables sont initialement vides.
- À la réception d'une trame sur un port p, le pont
  - regarde l'adresse MAC de l'émetteur de la trame (champ SA)

  $\Rightarrow$ il sait que la machine d'adresse SA peut être jointe en utilisant le port p
  - supprime si besoin l'adresse SA de sa table de commutation
  - et y insère l'enregistrement (SA, p)

Exemple d'acheminement avec remplissage automatique



- Hypothèse :
  - **M1** transmet sur le LAN1 une trame destinée à **M6**.
  - La table de commutation de P1 est vide.
  - La table de commutation de P2 contient l'enregistrement (M6,3).

$\Rightarrow$ La trame est retransmise

  - par P1 sur tous ses ports $\neq$ du port de réception : 2 et 3
  - par P2 sur son port 3 uniquement

- État des tables de commutation après l'acheminement de la trame :
  - P1 : (M1, 1)
  - P2 : (M1, 1), (M6,3)

Algorithme de commutation

Traitement par un pont d'une trame reçue sur un port p :

extraire les champs DA et SA de la trame

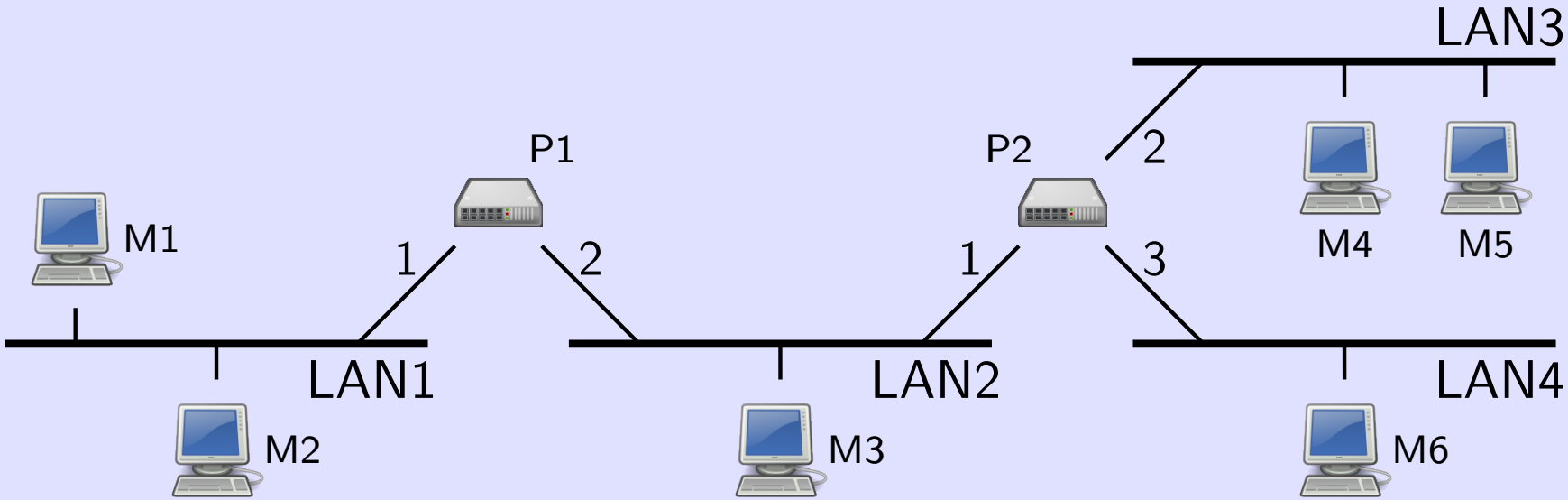
remplissage de la table

si il existe un enregistrement (adr, p') dans la table avec adr = SA
    supprimer cet enregistrement
insérer (SA, p) dans la table

retransmission de la trame

si il existe un enregistrement (adr, p') dans la table avec adr = DA
    si $p' \neq p$
        transmettre la trame sur le port p'
sinon
    pour tout port $p' \neq p$
        transmettre la trame sur le port p'

Intérêt des ponts



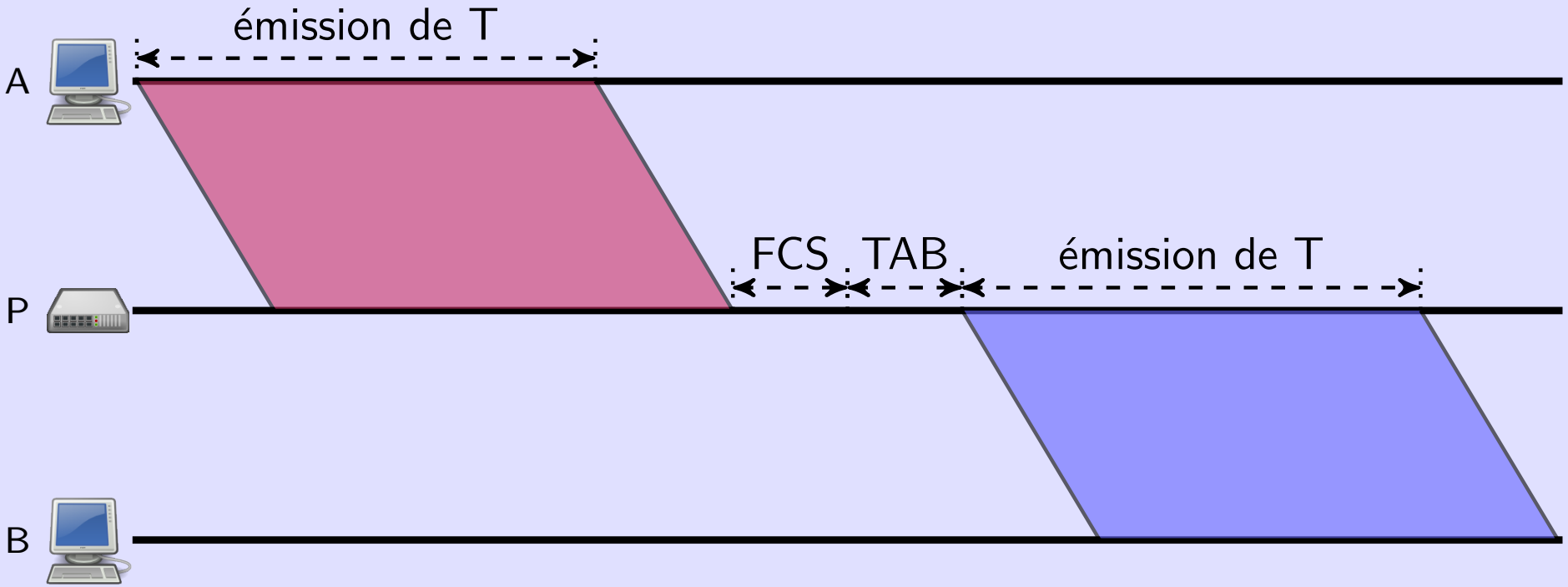
1. connecter des machines
2. s'affranchir des contraintes de distance d'Ethernet et augmenter la couverture du réseau
3. meilleures performances grâce à la réduction du **domaine de collision**
   - Exemple : M4 peut envoyer une trame à M5 pendant que M1 envoie une trame à M2. Pas de collision si M5 et M2 sont dans les tables de P2 et P1.
   - moins de collisions $\Rightarrow$ moins de retransmissions $\Rightarrow$ meilleur débit utile
4. confidentialité accrue : les machines ne voient plus toutes les trames
   - Exemple : M4 et M5 peuvent s'échanger des données confidentielles. Si elles sont dans la table de P2 aucune autre machine ne les recevra.

Stratégies de commutation

- À partir de quel instant un pont va-t-il relayer une trame reçue ?
- plusieurs stratégies possibles
- Nous allons en voir 2 :
  - **Store and Forward** — réception complète, contrôle d'erreur puis retransmission
  - **Fast Forward** — retransmission au plus tôt/dès que possible

Store and Forward

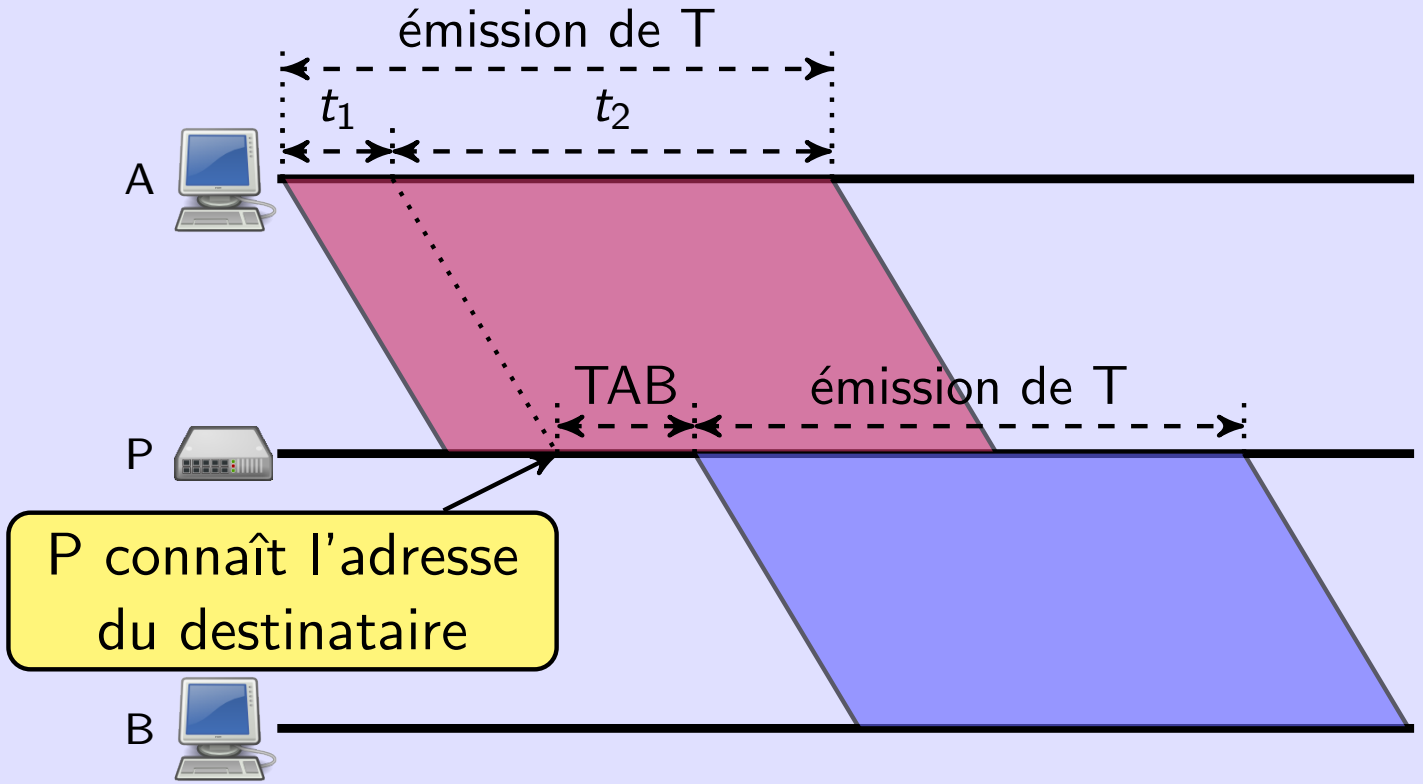
- retransmission de la trame T après réception de tous les bits, vérification du FCS et consultation de la table de commutation



FCS	vérification du champ FCS de la trame T
TAB	consultation de la table de commutation

Fast Forward

- retransmission de la trame T après réception de l'adresse de destination et consultation de la table de commutation



t_1	émission du préambule, SFD et champ DA de la trame T
t_2	émission du reste de la trame T
TAB	consultation de la table de commutation

Comparaison entre switchs et routeurs

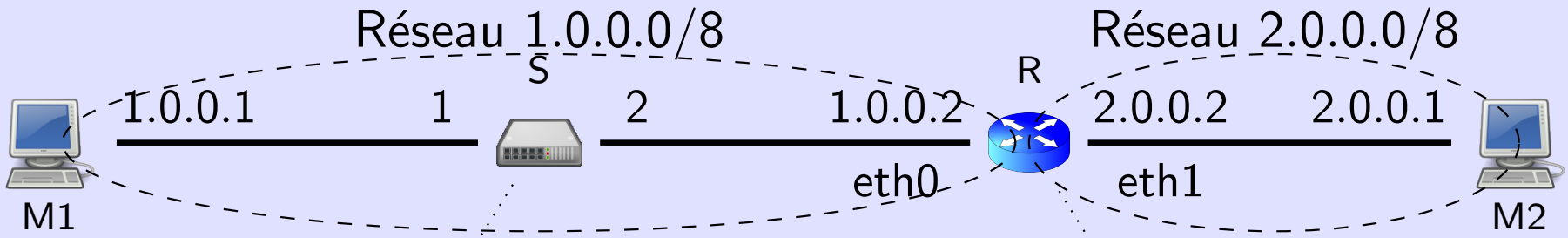
	Niveau OSI	PDUs traités	Adresses traitées	Action réalisée
Switch	Liaison	Trames Ethernet	MAC	Commutation
Routeur	Réseau	Paquets IP	IP	Routage

- Le switch interconnecte des machines d'un même réseau IP.

  Il utilise sa table de commutation pour savoir vers quel port rediriger une trame.
- Le routeur interconnecte des machines de réseaux IP différents.

  Il utilise sa table de routage pour savoir sur quelle interface et vers quel routeur (dans le cas d'une remise indirecte) rediriger un paquet.

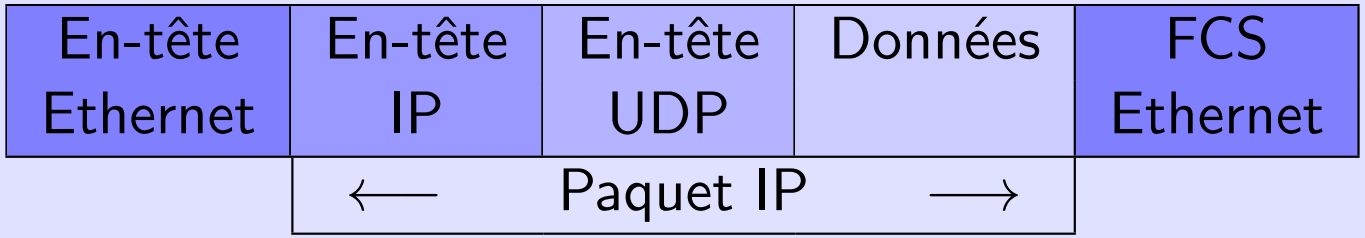
Exemple (1/2)



adresse	port
@MAC de M1	1
@MAC de R	2

dest.	masque	routeur	interface
1.0.0.0	255.0.0.0	—	eth0
2.0.0.0	255.0.0.0	—	eth1

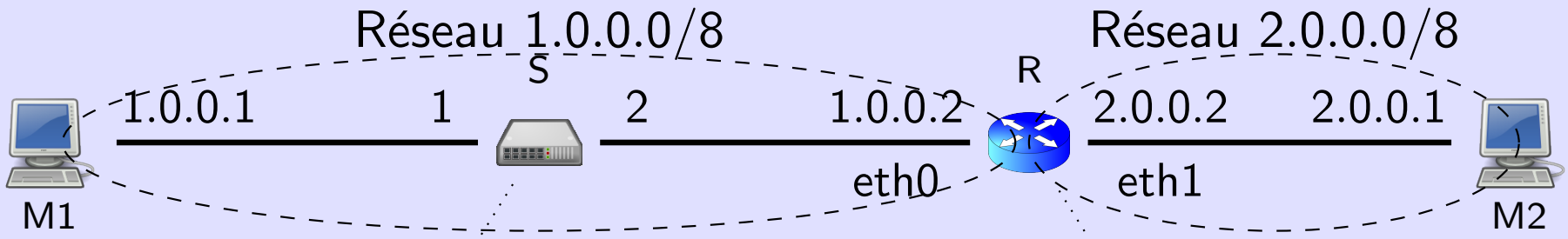
Hypothèse : M1 envoie des données à M2 avec le protocole de transport UDP.
Structure de la trame Ethernet envoyée par M1 au switch S :



adresses utilisées :

Dans l'en-tête Ethernet		Dans l'en-tête IP	
@MAC source	@MAC dest.	@IP source	@IP dest.
M1	R (eth0)	1.0.0.1	2.0.0.1

Exemple (2/2)



adresse	port
@MAC de M1	1
@MAC de R	2

dest.	masque	routeur	interface
1.0.0.0	255.0.0.0	—	eth0
2.0.0.0	255.0.0.0	—	eth1

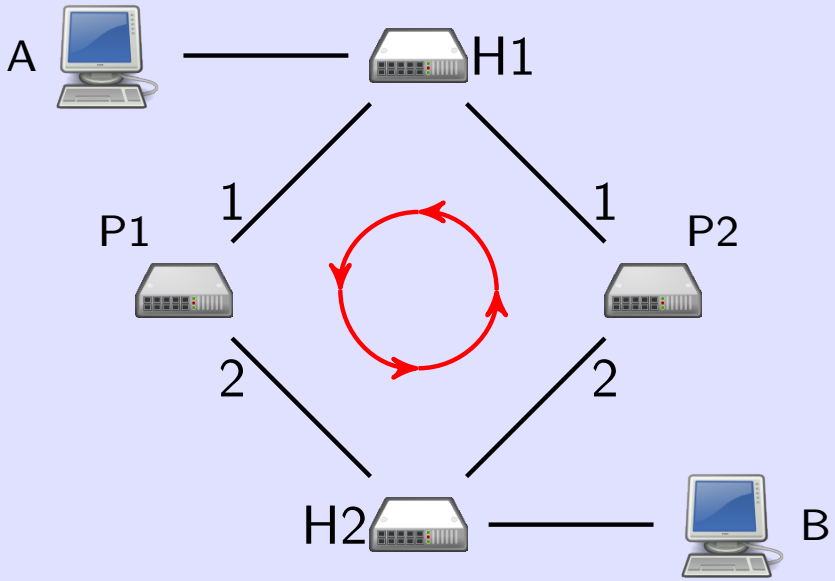
- S regarde l'adresse MAC de destination dans le PCI Ethernet : @MAC de R.
- consultation de la table de commutation : retransmettre trame sur port 2

⇒ S retransmet la trame à l'identique sur son port 2.

- R regarde l'adresse IP de destination dans le PCI IP : 2.0.0.1.
- consultation de la table de routage : retransmettre paquet sur eth1 en remise directe

⇒ R retransmet le paquet à l'identique sur eth1 en l'encapsulant dans une nouvelle trame Ethernet.

Problème de cycle

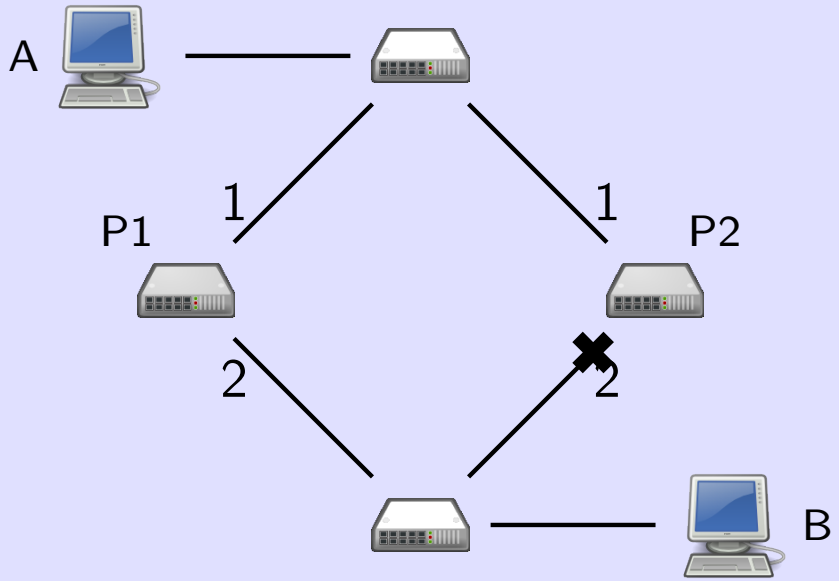


- Hypothèses
  - Les tables de commutation de P1 et P2 sont vides.
  - A envoie une trame à B.
- La trame est reçue par P1 et P2 depuis le hub H1. Ils vont ensuite se l'échanger sur le hub H2 puis sur le hub H1, . . .

$\Rightarrow$ La trame va circuler indéfiniment sur le réseau.

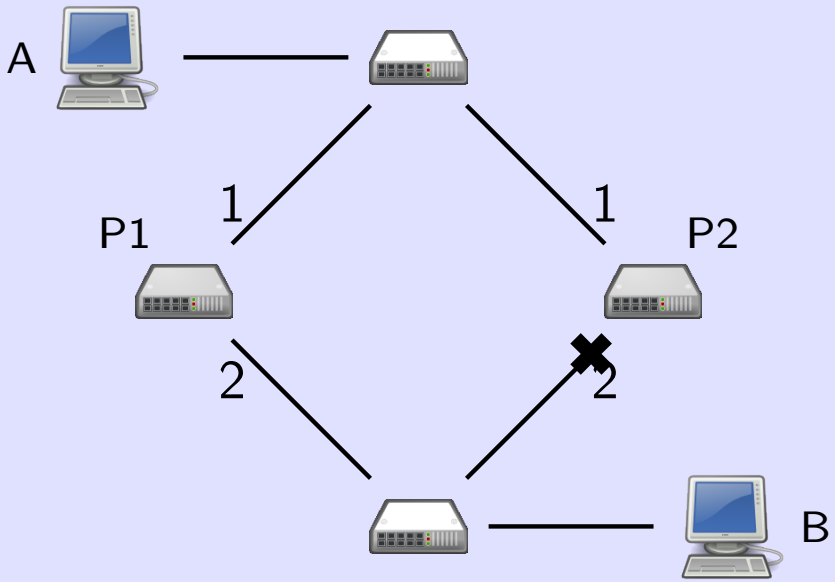
- Ce problème se pose lorsqu'il existe plusieurs chemins pour arriver d'un pont à un autre : un **cycle** (ou **boucle**).

Solution au problème de cycle



- éliminer les cycles de manière **logique** en **bloquant certains ports** à l'aide du protocole de l'**arbre couvrant**
  - protocole **STP** (Spanning Tree Protocol), norme IEEE 802.1D
  - STP est utilisé par les ponts uniquement.
- port bloqué $\Rightarrow$ $\begin{cases} \text{le pont ignore les trames reçues sur ce port} \\ \text{le pont ne retransmet pas de trames sur ce port} \end{cases}$
- Exemple : en bloquant le port 2 du pont P2 il n'y a plus de cycle.

Utilité des cycles

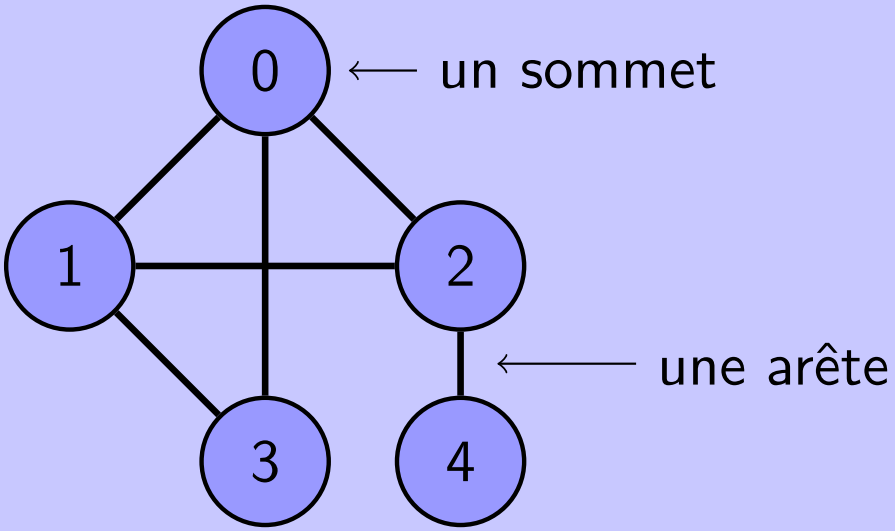


- Pourquoi créer des cycles dans le réseau ?

  pour avoir un réseau **résistant aux pannes**
- Si P1 tombe en panne :
  - P2 s'en apercevra car il ne recevra plus de trames de P1
  - il débloquera ensuite son port 2

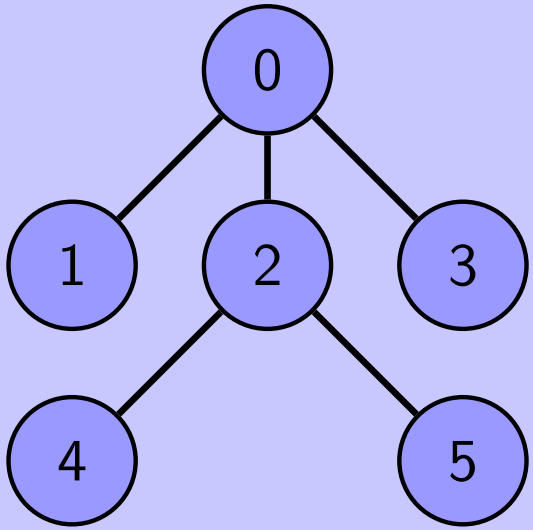
  $\Rightarrow$ Les machines resteront connectées.

Un graphe



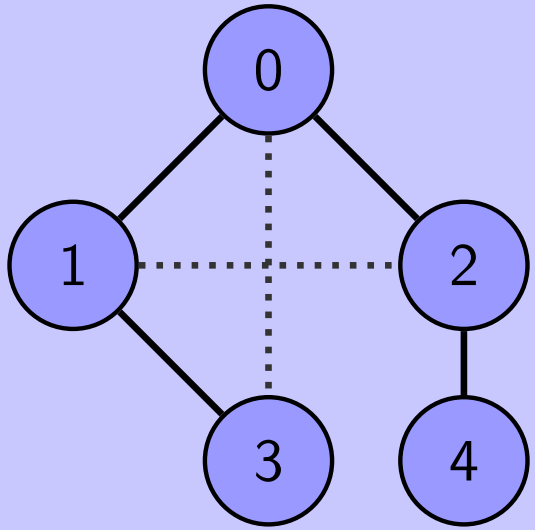
- 1 cycle = 1 chemin qui part d'un sommet et revient à ce sommet sans emprunter deux fois la même arête
- Exemples de cycles :
  - 0, 1, 3, 0
  - 0, 3, 1, 2, 0

Un arbre



- 1 arbre = 1 graphe sans cycle

Un arbre couvrant



- On supprime des arêtes du graphe pour obtenir un arbre.
- Tous les sommets doivent rester connectés.
- Pour un graphe, on peut avoir plusieurs arbres couvrants.

Principe du protocole de l'arbre couvrant

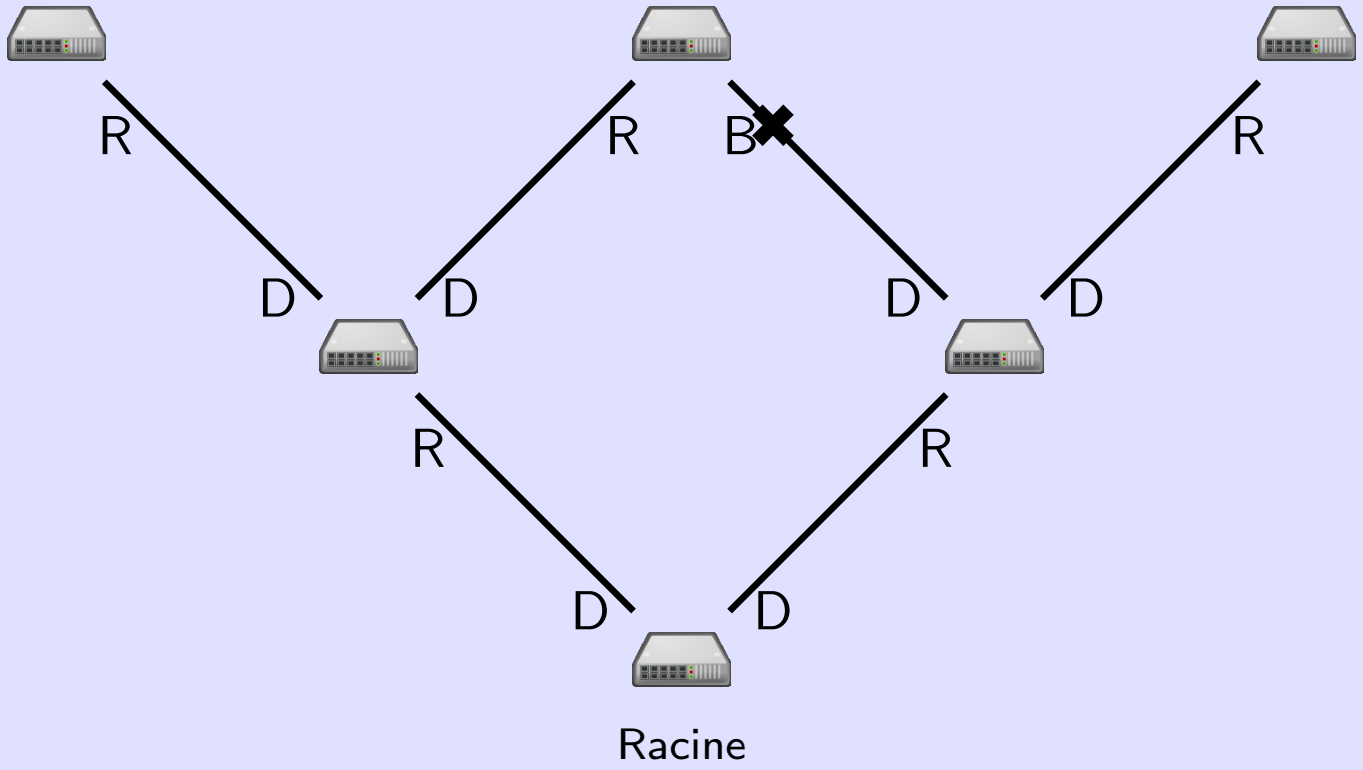
- Un **pont racine** va être désigné.
- C'est le pont ayant la plus petite adresse MAC (parmi tous les ponts).
- Au début, le pont racine n'est pas connu et chaque pont se considère comme le pont racine.
- Par l'échange de **BPDU**s (Bridge Protocol Data Unit), chaque pont va progressivement découvrir
  - l'adresse MAC du pont racine
  - sa **distance** au pont racine : le nombre de ponts à traverser sur le plus court chemin qui le mène au pont racine
- Tous les ports qui n'apparaissent pas sur les plus courts chemins calculés sont bloqués : ils ne font pas partie de l'arbre couvrant.
- Mais les ports bloqués sont toujours utilisés par le protocole STP : un pont continue à analyser les BPDUs qu'il reçoit sur un port bloqué.

Classification des ports

À l'issue du protocole un pont associera un des trois états suivants à chacun de ses ports :

- **bloqué** — Le pont ignore les trames qu'il reçoit sur ce port (sauf les BPDUs).
- **racine** — port utilisé pour joindre le pont racine sur le plus court chemin calculé
- **désigné** — port utilisé pour joindre un autre pont plus éloigné que lui du pont racine

Classification des ports — Exemple



D = port désigné R = port racine B = port bloqué

- En retirant les ports bloqués on obtient un arbre.
- Chaque pont ($\neq$ du pont racine) a exactement un port racine.
- En suivant le port racine d'un pont, on remonte à la racine.
- En suivant un port désigné depuis un pont, on s'éloigne de la racine.

Fonctionnement général du protocole

- Chaque pont a trois variables qui forment sa **configuration** :
  - **racine** — l'adresse MAC du pont qu'il considère (pour l'instant) comme le pont racine
  - **distance** — sa distance à racine
  - **portRacine** — port sur le plus court chemin qui le mène à racine
- Au début, tout pont se considère comme le pont racine et initialise
  - racine à son adresse MAC
  - distance à 0
  - et portRacine à -1 (un port inexistant)
- Les ponts s'échangent ensuite des BPDUs.
- La réception d'une BPDU peut éventuellement modifier la configuration d'une machine.
- Le protocole termine quand les configurations sont stables : l'envoi d'une BPDU ne peut plus changer la configuration d'une machine.
- À la terminaison du protocole, tous les ponts connaissent le pont racine et ont bloqué des ports qui créent des cycles.

Les trames BPDU

- **BPDU** = Bridge Protocol Data Unit
- Une BPDU n'est jamais retransmise par un pont.
  - La commutation n'a pas lieu dans le cas d'une BPDU.
- champ DA (adresse destinataire) d'une BPDU = `01:80:C2:00:00:00`
  - adresse spéciale réservée pour ce protocole
- Une BPDU contient quatre champs :
  - **R** adresse MAC du pont racine
  - **D** distance au pont racine
  - **E** adresse MAC du pont émetteur de la BPDU
  - **P** numéro du port utilisé par E pour émettre la BPDU

Envoi de BPDUs

- Soit un pont E avec une configuration (racine, distance, portRacine).
- Régulièrement (toutes les 2 sec.), E envoie sur chacun de ses ports p ($\neq$ de portRacine) la BPDU

  (racine, distance, E, p)
- En envoyant une BPDU, un pont informe les autres de la connaissance qu'il a du pont racine et de sa distance par rapport à celui-ci.
- L'envoi d'une BPDU (R, D, E, P) signifie :
  Je suis le pont E et j'ai émis cette BPDU depuis mon port P. Je pense que le pont racine est R et je suis à une distance D de R.

Comparaison de BPDUs

- Les ponts ont besoin de comparer des BPDUs reçues pour modifier leurs configurations.
- Soient deux BPDUs $B_1 = (R_1, D_1, E_1, P_1)$ et $B_2 = (R_2, D_2, E_2, P_2)$
- B_1 est **meilleure** que B_2 si et seulement si :

$$
\begin{array}{llllllll}
 & R_1 < R_2 & & & & & & \\
ou & R_1 = R_2 & et & D_1 < D_2 & & & & \\
ou & R_1 = R_2 & et & D_1 = D_2 & et & E_1 < E_2 & & \\
ou & R_1 = R_2 & et & D_1 = D_2 & et & E_1 = E_2 & et & P_1 < P_2
\end{array}
$$

$\Rightarrow$ On compare sur l'adresse du pont racine, puis à adresses égales sur la distance, puis ...

Comparaison de BPDUs — Exemples

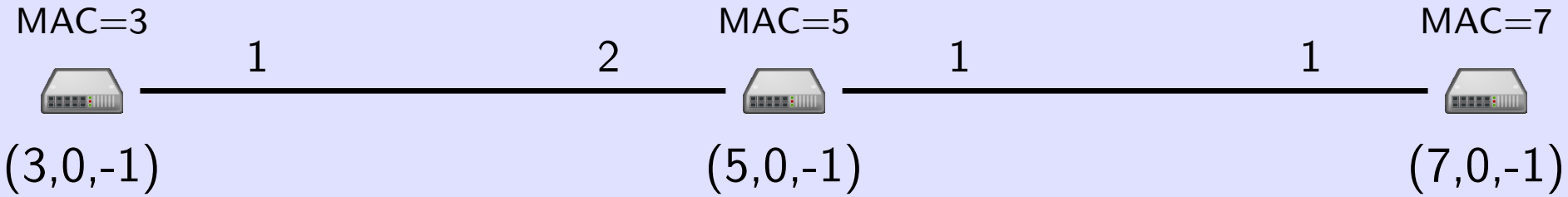
1. Je reçois les BPDUs $(\underline{\mathbf{5}}, 2, 7, 2)$ et $(\underline{\mathbf{3}}, 1, 6, 1)$.
   - 7 considère 5 comme le pont racine.
   - 6 considère 3 comme le pont racine.
   - $\Rightarrow$ 5 ne peut pas être le pont racine mais 3 l'est peut-être.
   - $\Rightarrow$ La BPDU envoyée par 6 est la meilleure.
2. Je reçois les BPDUs $(4, \underline{\mathbf{2}}, 7, 2)$ et $(4, \underline{\mathbf{1}}, 5, 1)$.
   - 5 et 7 considèrent tous les deux 4 comme le pont racine.
   - Mais le chemin de 5 pour rejoindre 4 est plus court (1 contre 2).
   - $\Rightarrow$ La BPDU envoyée par 5 est la meilleure.
3. Je reçois les BPDUs $(3, 2, \underline{\mathbf{7}}, 2)$ et $(3, 2, \underline{\mathbf{5}}, 1)$.
   - Les deux BPDUs indiquent le même pont racine avec la même distance.
   - $\Rightarrow$ La BPDU envoyée par 5 est la meilleure.
   - (Le choix est ici arbitraire : on prend 5, car $5 < 7$, mais on pourrait aussi prendre 7).

Changement de configuration

- Soit un pont
  - avec une configuration (racine, distance, portRacine)
  - recevant une BPDU (r, d, e, p) sur un port q
- Le pont change sa configuration si (r, d + 1) $<$ (racine, distance)
  autrement dit, si une des deux conditions est vérifiée :
  - r $<$ racine
    (Celui que je pense être le pont racine a une adresse plus grande que celui indiqué dans la BPDU.)
  - r $=$ racine ET d + 1 $<$ distance
    (Je connais le même pont racine que celui indiqué dans la BPDU, mais le chemin passant par e est plus court que celui que je connais.)
- La nouvelle configuration du pont devient alors
  - racine $=$ r
  - distance $=$ d + 1
  - portRacine $=$ q
- Le changement est visible dans les BPDUs envoyées ensuite par P.
- En cas de changement de configuration, le pont doit aussi débloquer les ports qu'ils avaient bloqués précédemment (s'il y en a).
  - (Le blocage de ces ports est valable uniquement pour l'ancienne configuration)

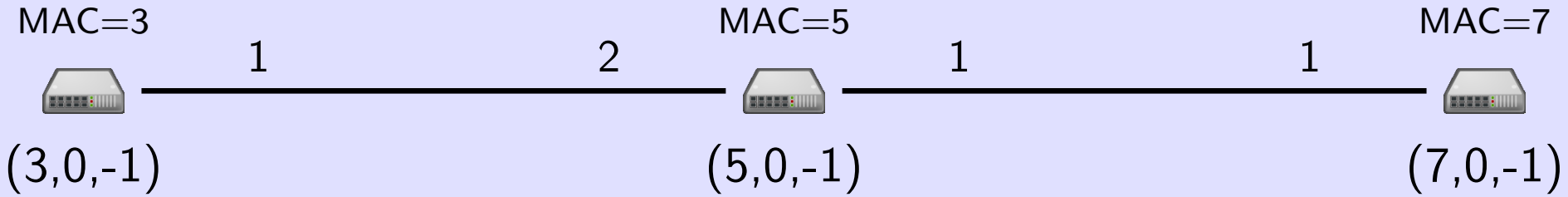
Changement de configuration — Exemple

Étape 0 : configuration initiale des ponts

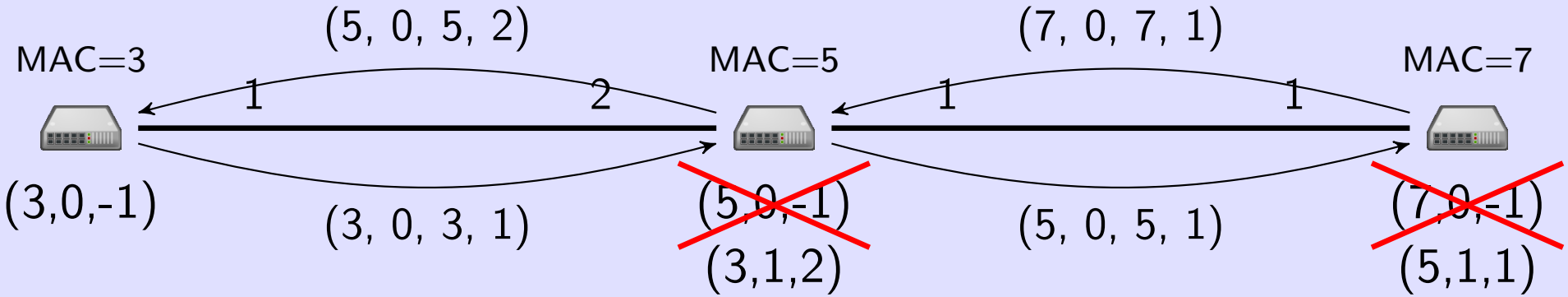


Changement de configuration — Exemple

Étape 0 : configuration initiale des ponts

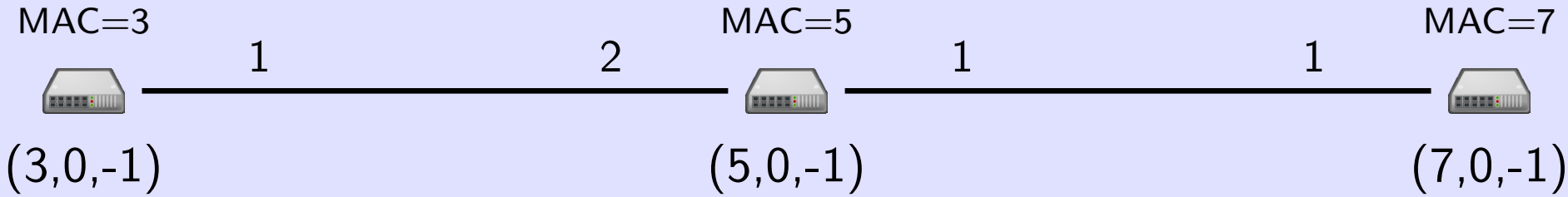


Étape 1 : 1er envoi de BPDUs

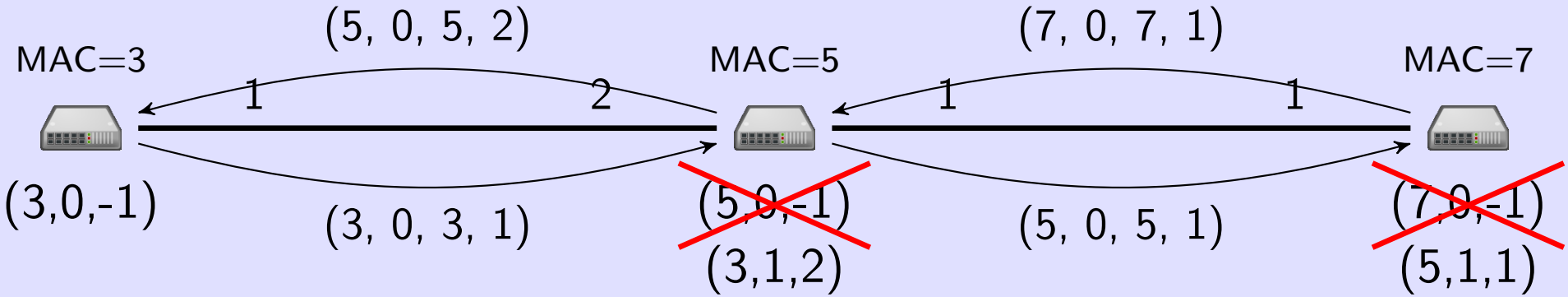


Changement de configuration — Exemple

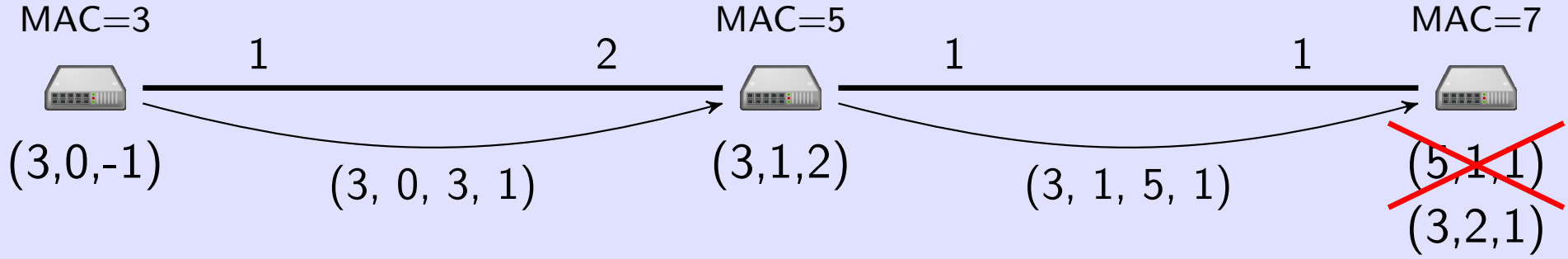
Étape 0 : configuration initiale des ponts



Étape 1 : 1er envoi de BPDUs



Étape 2 : 2ème envoi de BPDUs



Blocage de port

- Soit un pont o
  - avec une configuration (racine, distance, portRacine)
  - recevant une BPDU (r, d, e, p) sur un port q ($\neq$ de portRacine)
- Le pont bloque q si r = racine et (d, e) $<$ (distance, o)

  autrement dit, si une des deux conditions est vérifiée :

  Condition 1 r = racine ET d $<$ distance

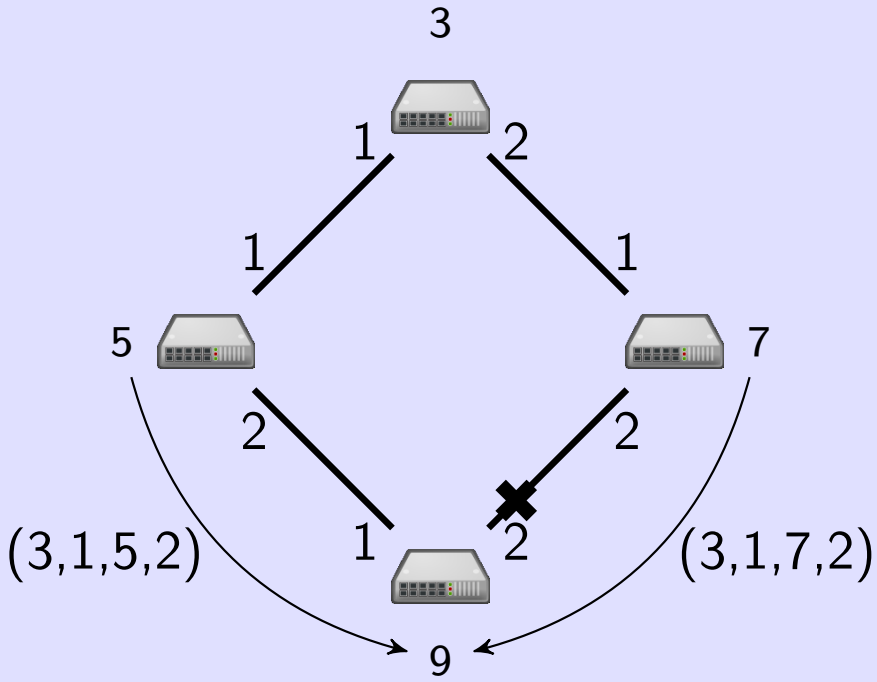
  Condition 2 r = racine ET d = distance ET e $<$ o

Blocage de port : condition 1

- Un pont a 2 chemins de même longueur le menant au pont racine.

⇒ Il bloque le port sur lequel il reçoit des BPDUs émises par le pont avec la plus grande adresse MAC.

- Exemple :



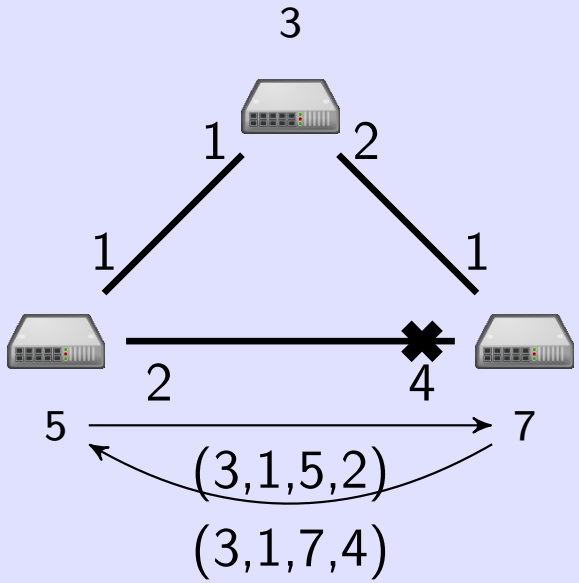
- 9 a la configuration (racine=3,distance=2,portRacine=1) et reçoit la BPDU (r=3,d=1,e=7,p=2) sur son port 2.
- On a bien r=racine ET d<distance ⇒ 7 bloque son port 2.

Blocage de port : condition 2

- Deux ponts sont connectés et à égale distance du pont racine.

⇒ C'est le pont ayant la plus grande adresse MAC qui bloque son port le connectant à l'autre.

- Exemple :



- 7 a la configuration (racine=3,distance=1,portRacine=1)

  et reçoit la BPDU (r=3,d=1,e=5,p=2) sur son port 4.
- On a bien r=racine ET d=distance ET e $< 7 \Rightarrow$ 7 bloque son port 4.

Résumé du fonctionnement de STP

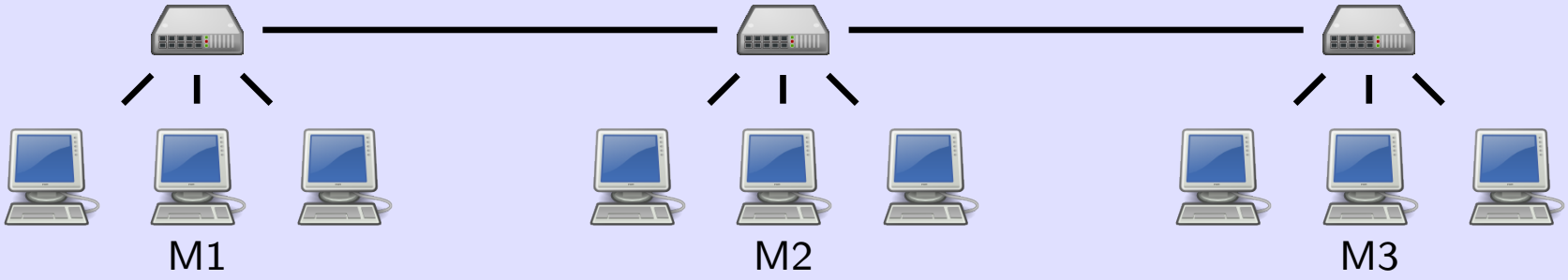
1. initialisation :
   - 1.1 racine = moi
   - 1.2 distance = 0
   - 1.3 portRacine = -1
2. envoyer la BPDU (racine, distance, moi, p) sur tout port p non bloqué et $\neq$ de portRacine
3. attendre des BPDUs sur tous les ports
4. soit (r, d, e, p) la meilleure BPDU reçue
5. si (r, d + 1) $<$ (racine, distance) alors
   - 5.1 débloquer les ports bloqués
   - 5.2 racine = r
   - 5.3 distance = d + 1
   - 5.4 portRacine = port de réception de (r, d, e, p)
6. bloquer tout port ($\neq$ de portRacine) sur lequel on a reçu une BPDU (r', d', e', p') telle que r' = racine et (d', e') $<$ (distance, moi)
7. si une configuration d'un pont a changé (point 5) alors aller en 2
8. sinon $\Rightarrow$ fin du protocole

Plan

Exemple introductif

Hypothèses :

- Soit le réseau suivant :



- M1, M2, M3
  1. s'échangent beaucoup de trames et en particulier des trames de diffusion
  2. et/ou s'échangent des données confidentielles que les autres machines ne doivent pas recevoir

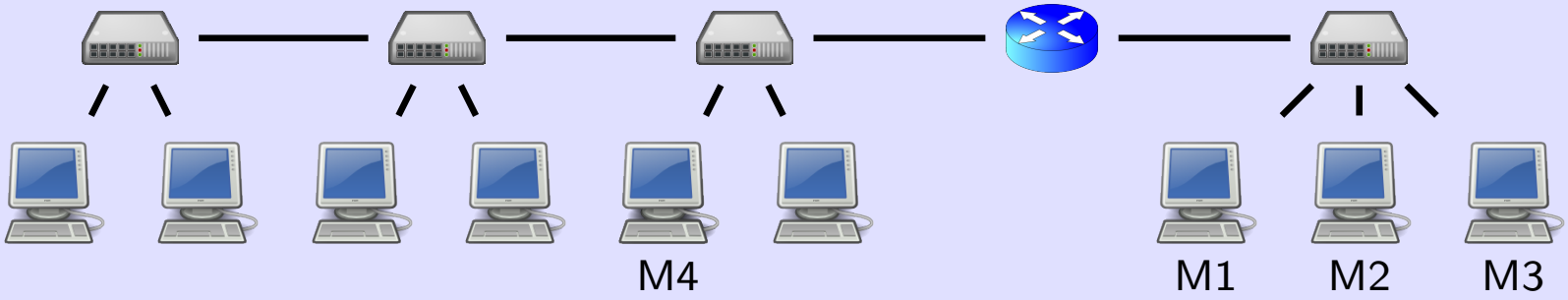
Objectifs :

1. éviter que M1, M2 et M3 n'encombrent le réseau
2. faire en sorte qu'une autre machine ne puisse pas espionner le trafic entre M1, M2 et M3

2 solutions possibles :

- Solution 1 : les mettre sur un réseau séparé
- Solution 2 : les répartir sur des **réseaux locaux virtuels**.

Solution 1 : création d'un réseau séparé



- Le routeur fonctionne comme une barrière entre les deux réseaux.
- Impossible pour M4 de capturer une trame émise par M1, M2 ou M3 (même une trame de diffusion).

$\Rightarrow$ répond bien à nos objectifs

Inconvénients :

- achat d'équipements supplémentaires (switchs, câbles, routeur, ...)
- répartition géographique rigide : M1, M2 et M3 ne peuvent pas être déplacées.

Solution 2 : répartition en réseaux locaux virtuels

- pas de changement de l'architecture physique du réseau
- création d'un réseau virtuel contenant uniquement M1, M2 et M3
- D'un point de vue logique, les deux solutions sont équivalentes
  ($\Leftrightarrow$ revient à créer deux réseaux étanches)

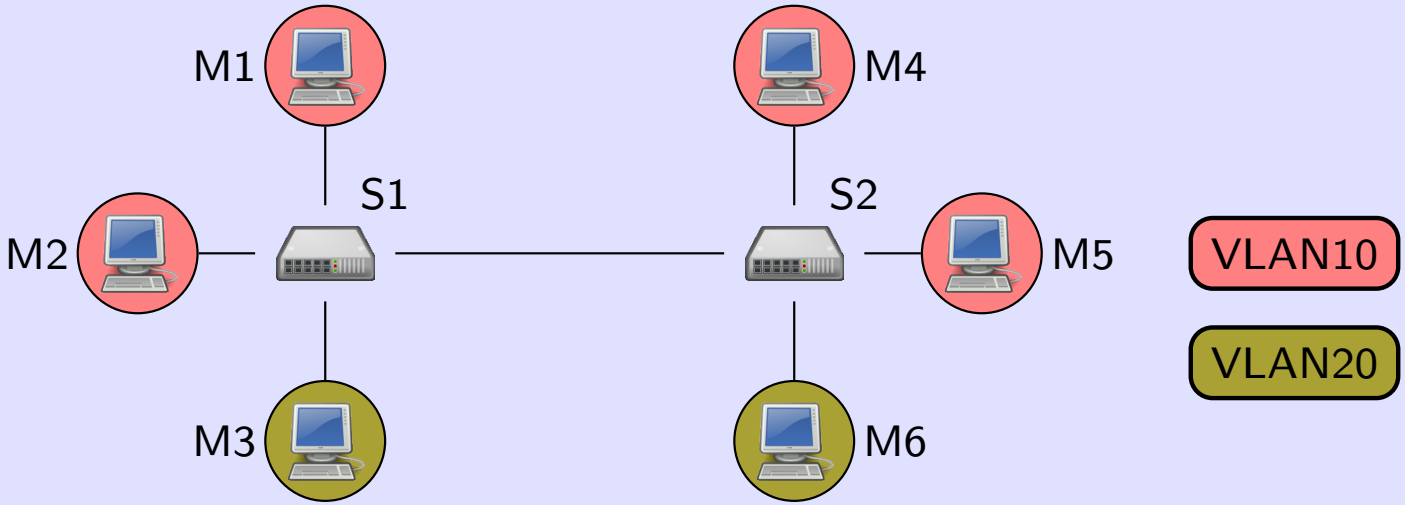
  mais la solution 2 n'a pas les inconvénients de la solution 1.
- réseaux locaux virtuels $\approx$ moyen d'avoir plusieurs réseaux logiques sur un seul réseau physique

Principe des réseaux locaux virtuels

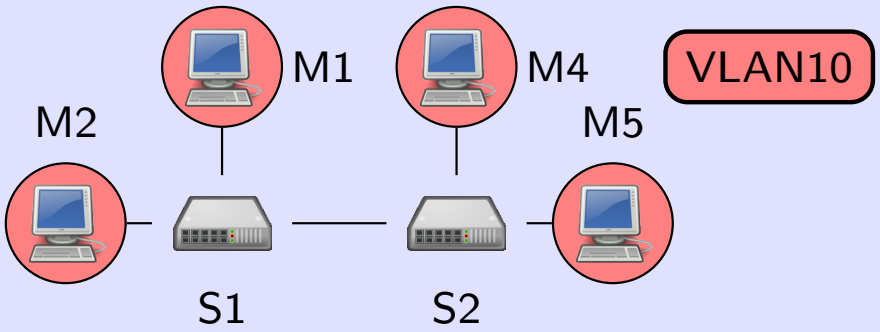
- On a un ensemble de réseaux locaux virtuels (ou VLANs, Virtual LANs).
- Chaque VLAN a un **identifiant**.
- Une machine est rattachée à un VLAN.
- La décomposition en VLANs implique un **découpage** du réseau :
  - Une machine ne peut communiquer qu'avec les machines du même VLAN.
- Le découpage en VLANs est à la charge de l'administrateur du réseau.
  - nécessite un travail de configuration des switchs (voir TPs 3 et 4) : créer des VLANs, attribuer des VLANs aux machines, ...
- Ce découpage logique se superpose à l'organisation physique.
  - Exemple de découpage : attribuer le VLAN10 au département R&T, le VLAN20 au département Informatique, ...

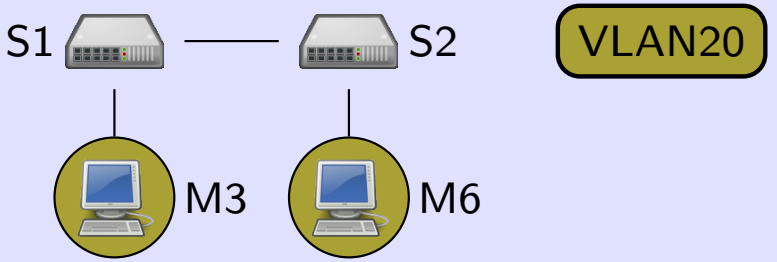
Vue physique/Vue logique — Exemple

- Vue physique : 6 machines connectées par deux switchs



- Vue logique : 2 réseaux logiques séparés





Avantages et inconvénients de la virtualisation

Avantages :

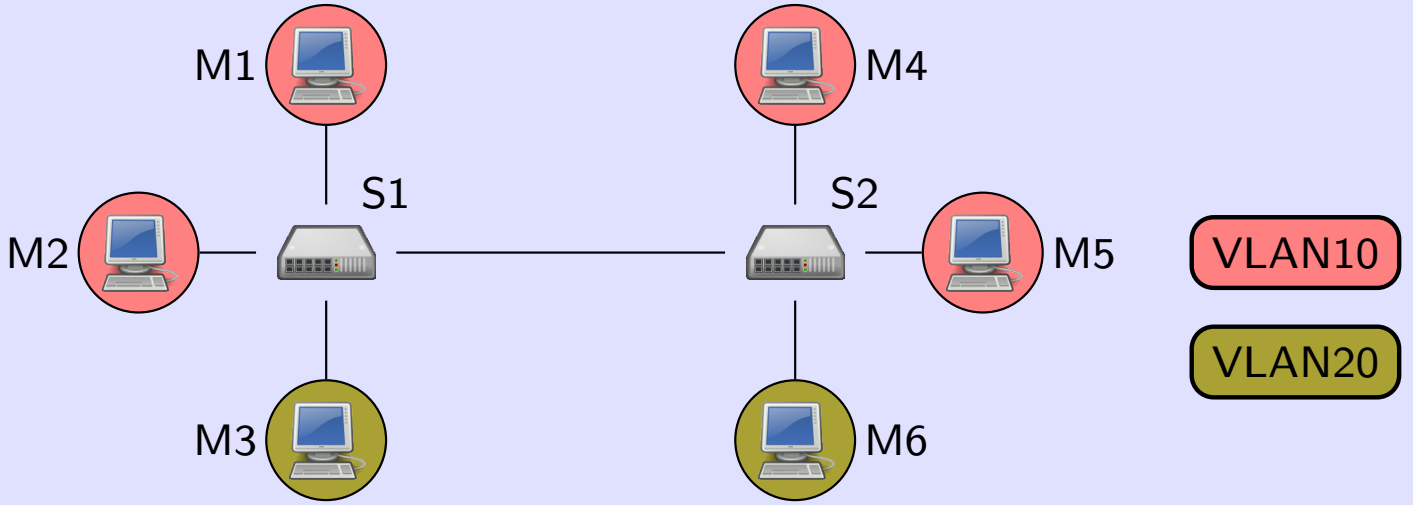
- Une trame de diffusion n'est reçue que par les machines du même VLAN.
- Une machine ne peut voir que les trames circulant sur son VLAN.
  - $\Rightarrow$ résout les contraintes de confidentialité
- L'architecture physique ne doit plus refléter l'architecture logique.
  - $\Rightarrow$ facilite la mobilité des machines
    - Exemple : une machine du département R&T peut être déplacée n'importe où dans l'IUT, elle restera toujours dans le VLAN R&T.

Inconvénient :

- Le réseau est découpé : deux machines sur des VLANs différents ne peuvent plus s'échanger de trames.

La norme IEEE 802.1Q

- Elle définit le fonctionnement des équipements nécessaires à la gestion des VLANs.
- Principale difficulté pour gérer les VLANs : pas d'information sur les VLANs dans les trames Ethernet.
- Exemple :



  - M1 envoie une trame de diffusion sur le VLAN 10
  - S1 retransmet cette trame à S2
  - Comment S2 sait-il que la trame circule sur le VLAN 10 (et donc ne doit pas être retransmise à M6) ?
- Solution choisie par la norme : rajouter dans l'en-tête Ethernet une **étiquette** qui donne des informations sur le VLAN.

Structure de la trame Ethernet IEEE 802.3 (hors préambule et SFD)

DA	SA	DL/EType	Données + Bourrage	FCS
6 o.	6 o.	2 o.	46–1 500 o.	4 o.

Structure de la trame Ethernet étiquetée IEEE 802.1Q

DA	SA	**Id de prot.**	**Pri.**	**DEI**	**VLAN-id**	DL/EType	Données + Bourrage	FCS
6 o.	6 o.	**2 o.**	**3 bits**	**1 bit**	**12 bits**	2 o.	46–1 500 o.	4 o.

← étiquette de 4 octets →

- étiquette = 4 champs insérés entre les champs SA et DL/EType
- champs Pri. et DEI : pas de rapport avec les VLANs

Détail des champs de l'étiquette

- Id. de prot. — 2 octets
  - Ce champ a toujours la valeur 0x8100.
  - mêmes taille et position que le champ DL/Etype en Ethernet standard

  $\Rightarrow$ Si un switch reçoit une trame avec DL/Etype = 0x8100 il sait que c'est une trame étiquetée. Sinon c'est une trame Ethernet standard.
- Pri. — 3 bits
  - priorité de la trame
  - 3 bits $\Rightarrow$ 8 niveaux de priorité
  - Les switchs retransmettent les trames par ordre de priorité décroissante.
- DEI — 1 bit
  - DEI = Drop Eligible Indicator
  - utilisé pour le contrôle de la congestion
  - DEI = 1 $\Rightarrow$ la trame peut être ignorée en cas de congestion du réseau
- VLAN-id — 12 bits
  - Il identifie le VLAN sur lequel la trame circule.
  - 12 bits $\Rightarrow 2^{12} = 4\,096$ VLANs au maximum

Liaisons trunk et standard

En présence de VLANs on distingue deux types de liaison :

Les liaisons trunk (trunk = tronc)

- Ce sont les liaisons sur lesquelles circulent des trames étiquetées.
- On peut faire passer sur ces liaisons des trames de différents VLANs.
- Ce sont généralement les liaisons switch↔switch.
  - Mais pas seulement : un routeur peut par exemple être connecté à un switch par une liaison trunk (voir diapo 119).

Les liaisons standard (non trunk)

- Ce sont les liaisons sur lesquelles circulent des trames Ethernet standard (sans étiquette).
- Les trames circulant sur une liaison standard appartiennent toutes au même VLAN.
- Ce sont généralement les liaisons switch↔machine.

Niveaux des VLANs

- Pour une liaison standard, un switch doit associer le port correspondant à un VLAN. 3 niveaux d'association possibles :
  - Niveau **1** : VLAN par **port** (statique)
    - L'administrateur attribue un VLAN à chaque port.
    - $\Leftrightarrow$ colorer les ports du switch avec des couleurs différentes (une couleur/VLAN)
    - Si une machine est déplacée, il faut reconfigurer les ports.
  - Niveau **2** : VLAN par **adresse MAC** (dynamique)
    - L'administrateur associe dans une table un VLAN à chaque adresse MAC.
    - Dès qu'une machine envoie une trame au switch auquel elle est reliée, il consulte cette table et associe le VLAN au port de réception.
    - Si une machine est déplacée, la reconfiguration est automatique.
  - Niveau **3** : VLAN par **adresse IP** (dynamique)
    - Comme le niveau 2 mais sur l'adresse IP.

Dans ce module on s'intéressera uniquement au VLAN par port.

Quand va-t-on ajouter/retirer l'étiquette ?

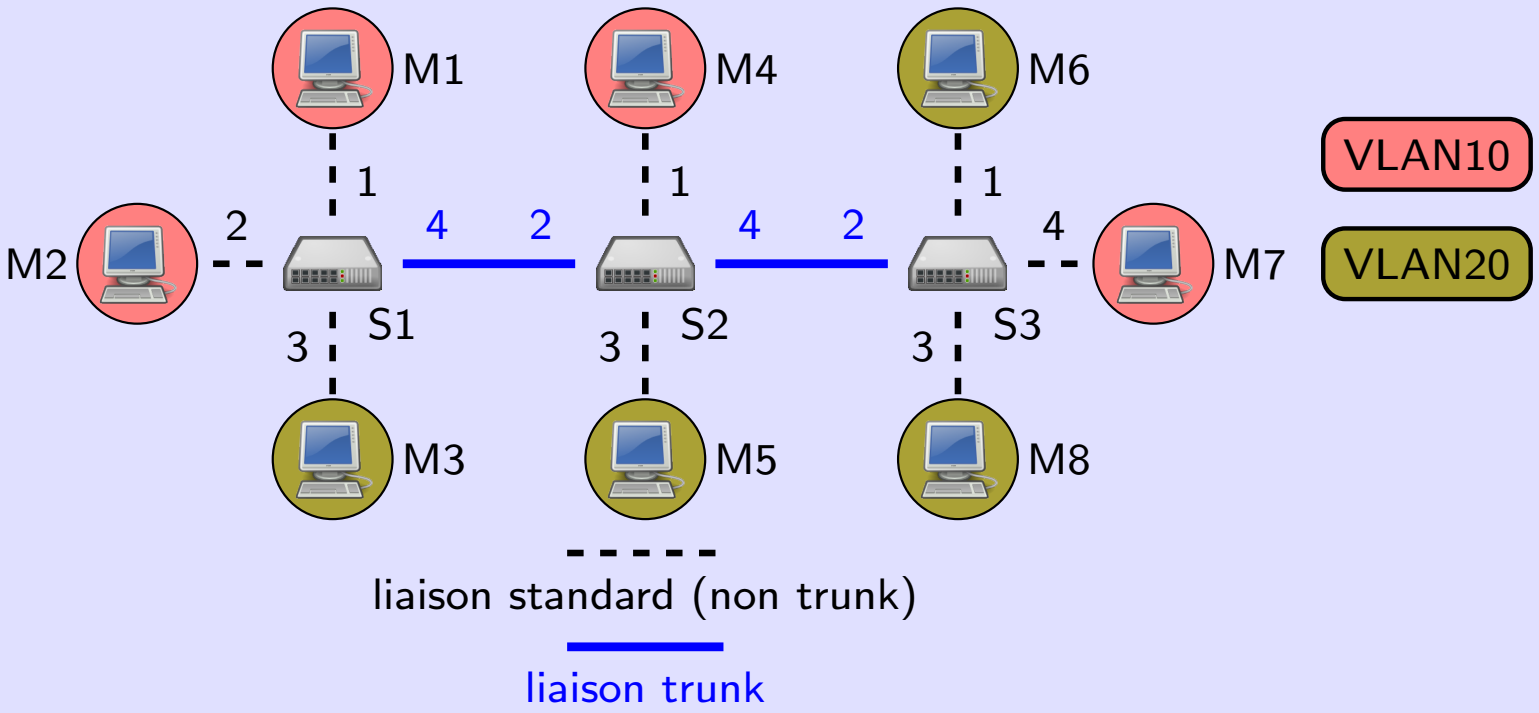
Insertion de l'étiquette

- par un switch qui reçoit une trame d'une machine sur une liaison standard
- Le switch détermine le VLAN associé au port de réception.
- Si la trame doit être redirigée sur une liaison trunk il rajoute, avant la redirection, l'étiquette contenant l'identifiant de du VLAN.

Retrait de l'étiquette

- par un switch avant redirection sur une liaison standard

Étiquetage des trames Ethernet — Exemple



- M3 envoie une trame à M6.
- S1 reçoit la trame. Il sait que son port 3 est associé au VLAN20.
- La trame doit être redirigée sur la liaison trunk S1↔S2.

⇒ Il insère dans la trame une étiquette avec VLAN-id = 20.

- La trame étiquetée va de S1 à S2 puis de S2 à S3.
- Avant de la délivrer à M6, S3 retire l'étiquette.

Étiquetage des trames par les machines

- Certaines machines “comprennent” les trames étiquetées.
- Une telle machine peut ajouter et retirer elle-même l'étiquette.
- Avantage : elle peut alors appartenir à plusieurs VLANs.
- Pour indiquer le VLAN sur lequel elle envoie une trame elle doit rajouter dans cette trame une étiquette avec l'identifiant du VLAN.
- La liaison qui la lie alors au switch est une liaison trunk.
- Voir TP 4.

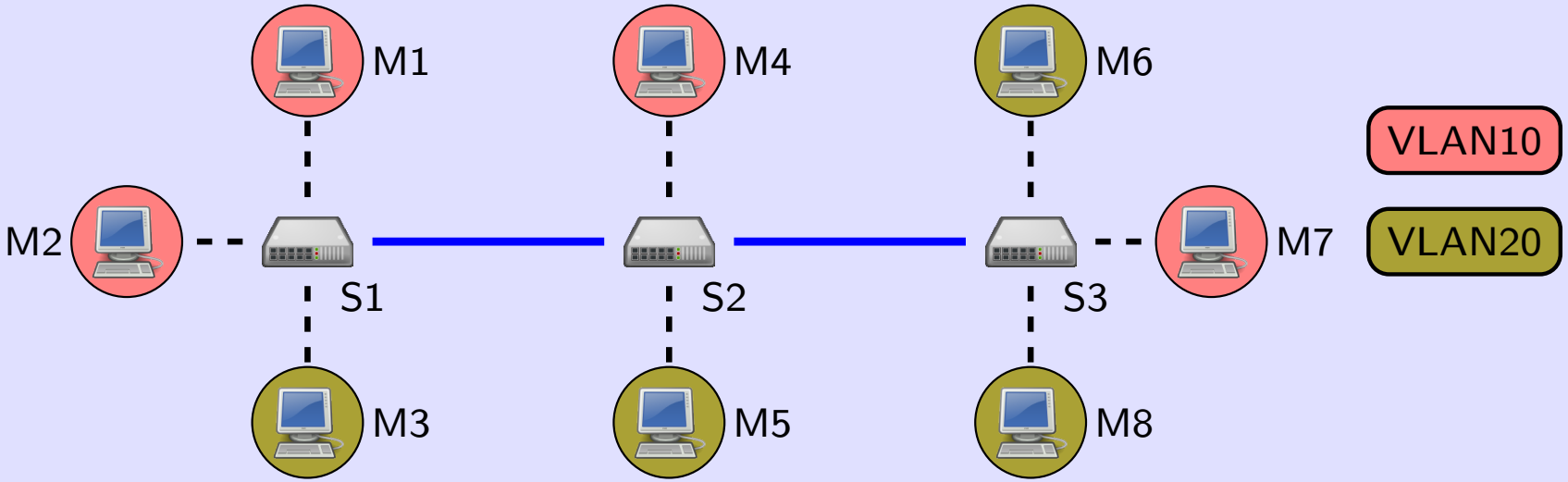
- En présence de VLANs, un switch ne doit jamais retransmettre une trame du VLAN V sur une liaison standard associée à un VLAN $V' \neq V$.

$\Rightarrow$ Avant de retransmettre une trame sur une liaison standard, il doit contrôler que le VLAN associé à la liaison est bien celui de la trame.

- Si ce n'est pas le cas, la trame n'est pas retransmise.
- Cela vaut aussi pour les trames de diffusion.

Énoncé

On considère le réseau suivant :



On capture une trame qui circule de S1 vers S2.
Voici les 48 premiers octets de la trame (hors champs Préambule et SFD) :

ff	ff	ff	ff	ff	ff	f8	b1	56	44	23	9a	81	00	e0	14
08	00	45	00	03	e8	00	01	00	00	20	11	6d	68	0a	00
14	04	ff	ff	ff	ff	27	10	4e	20	00	82	47	ee	44	65

Analyse de la trame — En-tête Ethernet

ff	**ff**	**ff**	**ff**	**ff**	**ff**	**f8**	**b1**	**56**	**44**	**23**	**9a**	**81**	**00**	**e0**	**14**
08	**00**	45	00	03	e8	00	01	00	00	20	11	6d	68	0a	00
14	04	ff	ff	ff	ff	27	10	4e	20	00	82	47	ee	44	65

- ff ff ff ff ff ff = champ Ethernet DA = adresse MAC de destination
- f8 b1 56 44 23 9a = champ Ethernet SA = adresse MAC source
- 81 00 = 2 octets à la place du DL/Etype Ethernet indiquant que la trame contient une étiquette de VLAN
  - $\Rightarrow$ 81 00 e0 14 = étiquette de VLAN
- e0 14_{16} = $\underbrace{111}_{PRI}\ \underbrace{0}_{DEI}\ \underbrace{0000\ 00010100}_{VLAN-id}$
  - PRI = Niveau de priorité de la trame = 7
  - DEI = Drop Eligible Indicator = 0
  - VLAN-id = 20
- 08 00 = champ Ethernet DL/Etype = code du protocole IP
  - La trame Ethernet encapsule un paquet IP.

ff	ff	ff	ff	ff	ff	f8	b1	56	44	23	9a	81	00	e0	14
08	00	**45**	00	**03**	**e8**	00	01	00	00	20	**11**	6d	68	**0a**	**00**
14	**04**	**ff**	**ff**	**ff**	**ff**	**27**	**10**	**4e**	**20**	00	82	47	ee	**44**	**65**

En-tête IP :

- 45 ⇒ version d'IP utilisée = 4 et longueur de l'en-tête IP = 5 × 4 octets = 20
- 03 e8 = longueur totale du paquet IP = 1 000 octets
- 11 = code du protocole encapsulé dans le paquet IP = code de UDP
  - Le paquet IP encapsule un datagramme UDP.
- 0a 00 14 04 = adresse IP source = 10.0.20.4
- ff ff ff ff = adresse IP destination = 255.255.255.255 = adresse de diffusion du réseau IP local

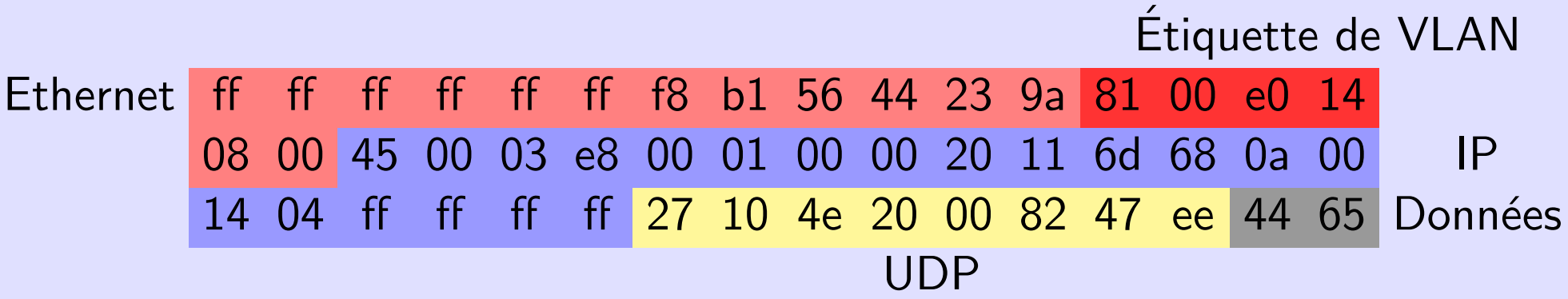
En-tête UDP :

- 27 10 = port UDP source = 10 000
- 4e 20 = port UDP destination = 20 000

Données : 44 65 ...

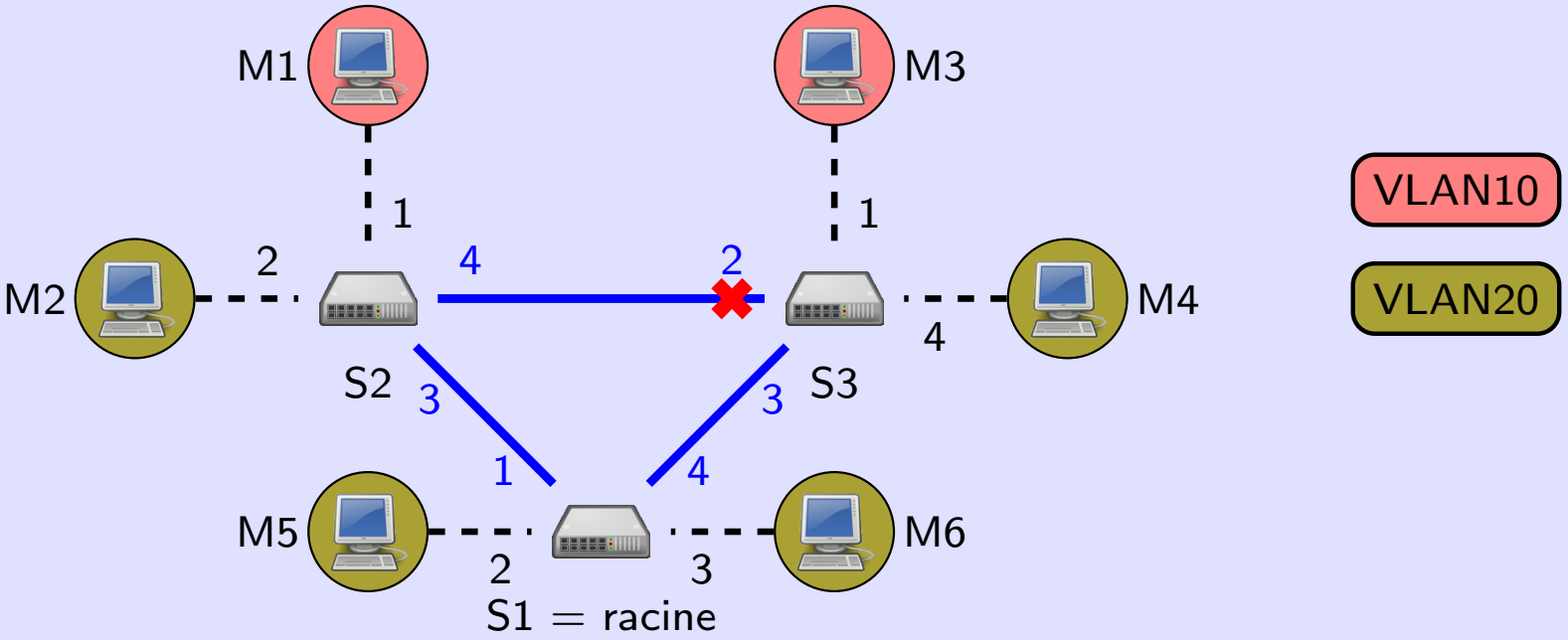
Analyse de la trame — Résumé

- Structure de la trame :



- La trame a été émise par la machine d'adresse MAC f8 b1 56 44 23 9a et d'adresse IP 10.0.20.4.
- Elle est destinée à toutes les machines (adr. Ethernet et IP de destination = adresses de diffusion) du VLAN20 se trouvant sur le réseau.
- Cette trame transporte des données émises
  - par un processus s'exécutant sur 10.0.20.4 et utilisant le port 10 000
  - et destinées à un processus utilisant le port 20 000.
- Le protocole de transport utilisé est UDP.
- Les données envoyées par le processus sont 44 65 ...

Utilisation de STP en présence de VLANs



- On suppose que les adresses MAC sont ordonnées ainsi : S1<S2<S3.

⇒ S3 bloque son port qui le mène à S2.

- Les trames échangées par M1 et M3 devront ensuite passer par S1.
- Or au niveau logique, il n'y a de cycle pour le VLAN 10.

⇒ Le blocage d'un port pour tous les VLANs n'est pas optimal.

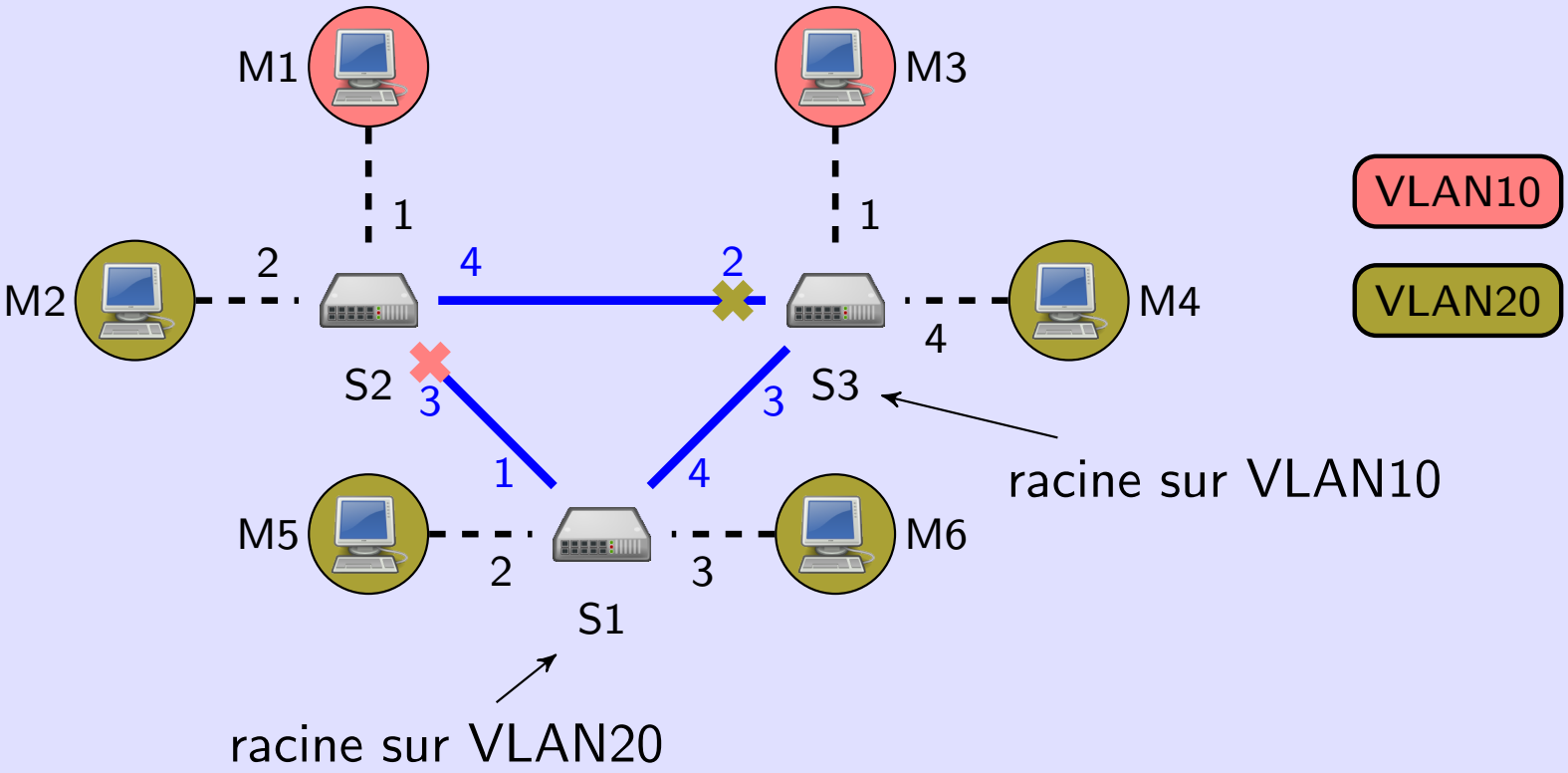
Le protocole MSTP

- La diapo. précédente montre que STP marche mais n'est pas toujours adaptée en présence de VLANs.
- Solution choisie : développement du protocole MSTP.
- norme IEEE 802.1s
- MSTP = Multiple Spanning Tree Protocol
- Principe : un arbre couvrant est construit pour chaque VLAN.
  - (Si il y a beaucoup de VLANs on peut les grouper, et on a alors un arbre couvrant par groupe.)

$\Rightarrow$ On a autant d'exécutions du protocole de l'arbre couvrant que de VLANs.

- Un port peut être bloqué pour un VLAN mais ouvert pour d'autres.
- Les BPDUs échangées avec le protocole MSTP contiennent un identifiant de VLAN indiquant le VLAN sur lequel la BPDU circule.
- Sinon, le principe du protocole est le même.

Le protocole MSTP — Exemple



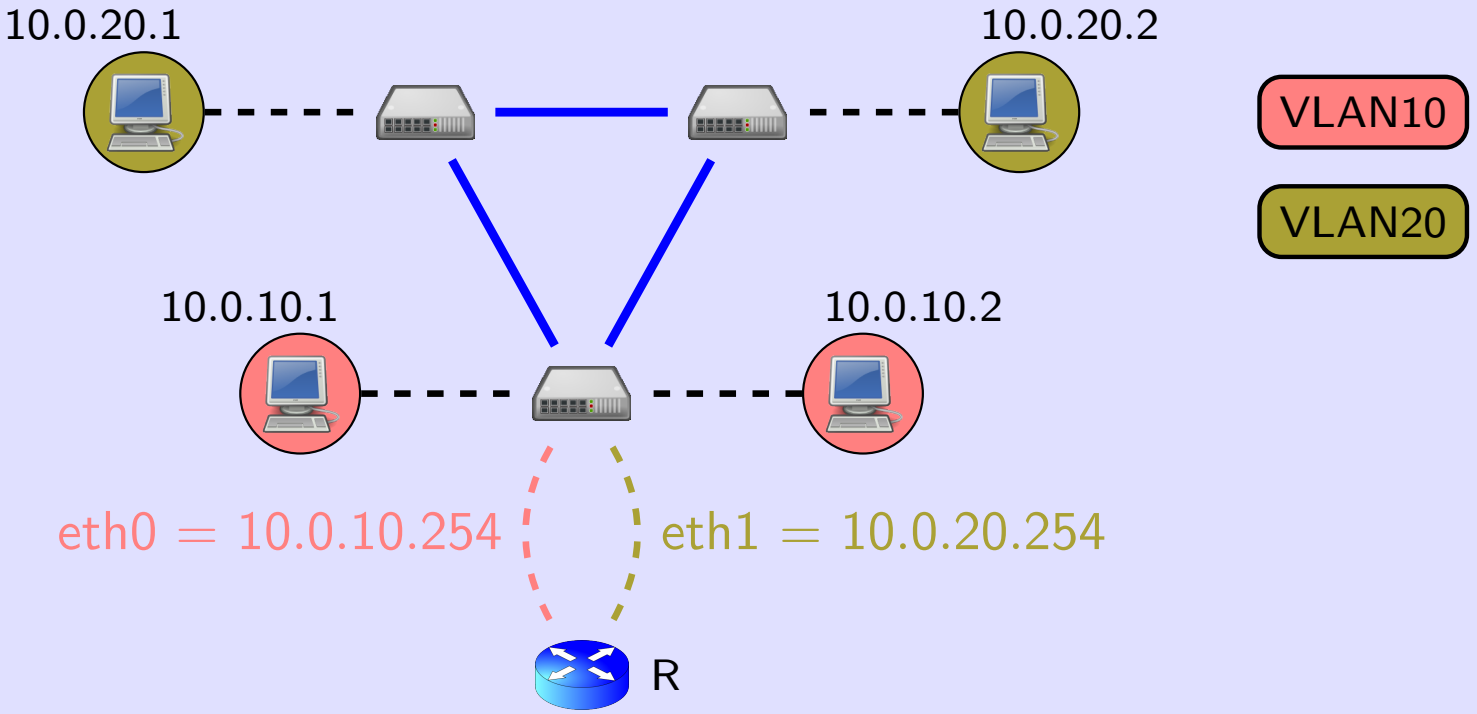
- On peut modifier les priorités des switchs sur les VLANs pour avoir :
  - S3 = racine sur VLAN10
  - S1 = racine sur VLAN20
- MSTP permet ensuite d'arriver à la configuration suivante :
  - S2 bloque son port 3 pour les trames du VLAN10
  - S3 bloque son port 2 pour les trames du VLAN20

$\Rightarrow$ M1 peut bien envoyer des trames à M3 sans passer par S1.

Trafic inter-VLANs

- On a vu que deux machines sur des VLANs différents ne peuvent plus s'échanger de trames.
- Comment peut-on faire pour que ces machines puissent tout de même s'échanger des données ?
- Cela ne peut être fait qu'au niveau supérieur : au niveau IP.
- Une solution :
  - On associe chaque VLAN à un réseau IP.
  - On rajoute un routeur qui est sur plusieurs VLANs.
- 2 mises en œuvre possibles (voir TP 4) :
  1. Le routeur a une interface physique par VLAN.
  2. Le routeur a
     - une seule interface physique ;
     - et plusieurs interfaces virtuelles (une par VLAN) associées à son interface physique.

Routage inter-VLANs — Solution 1



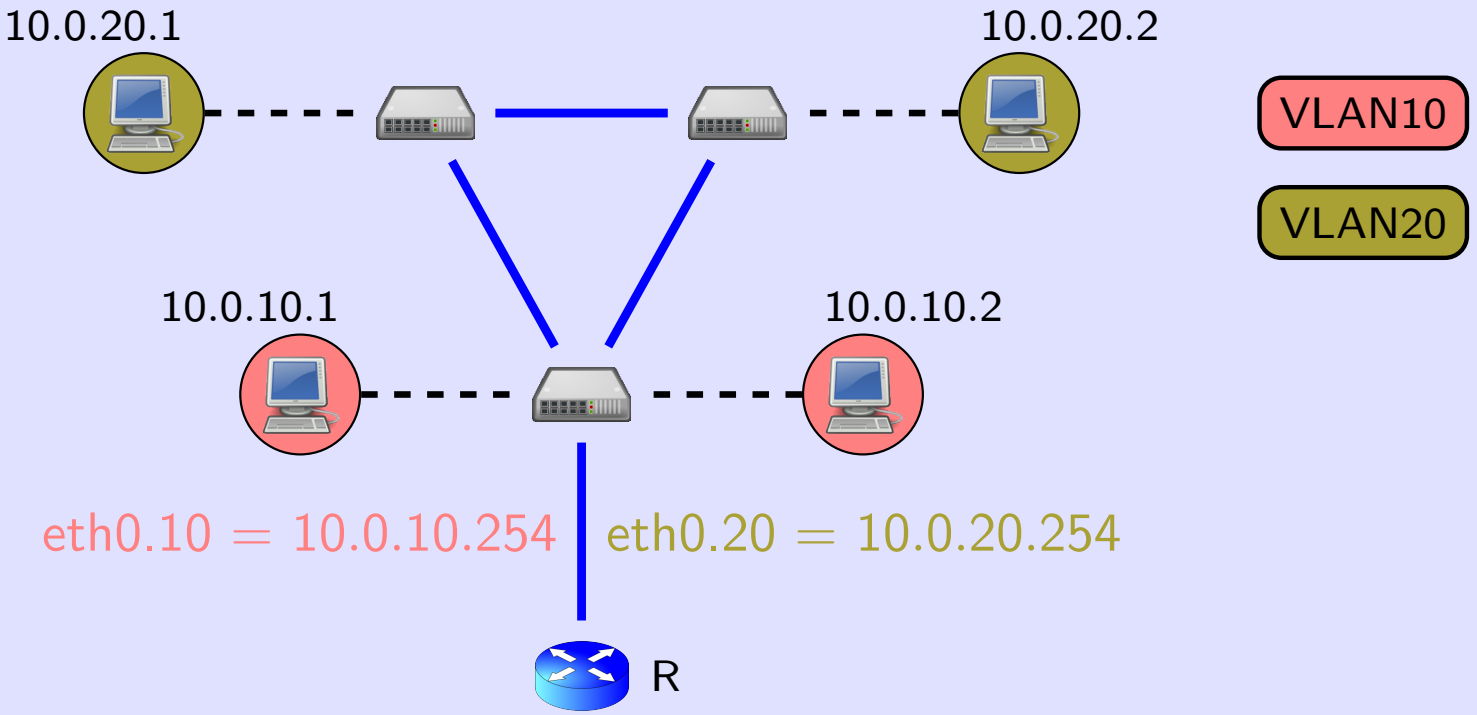
Configuration

- VLAN10 ⇔ réseau 10.0.10.0/24
- VLAN20 ⇔ réseau 10.0.20.0/24
- eth0 de R sur le VLAN10
- eth1 de R sur le VLAN20

2 trames pour envoyer un paquet de 10.0.20.1 à 10.0.10.2 :

1. 10.0.20.1 → 10.0.20.254 sur le VLAN20
2. 10.0.10.254 → 10.0.10.2 sur le VLAN10

Routage inter-VLANs — Solution 2



Configuration

- VLAN10 ⇔ réseau 10.0.10.0/24
- VLAN20 ⇔ réseau 10.0.20.0/24
- La liaison du routeur au switch est une liaison trunk.
- R a une interface physique (eth0) à laquelle sont rattachées :
  - eth0.10 : interface virtuelle pour le VLAN10
  - eth0.20 : interface virtuelle pour le VLAN20