

Universal Serial Bus (USB)



Pimp My Ride



Pimp My Serial Port



Pourquoi le USB?

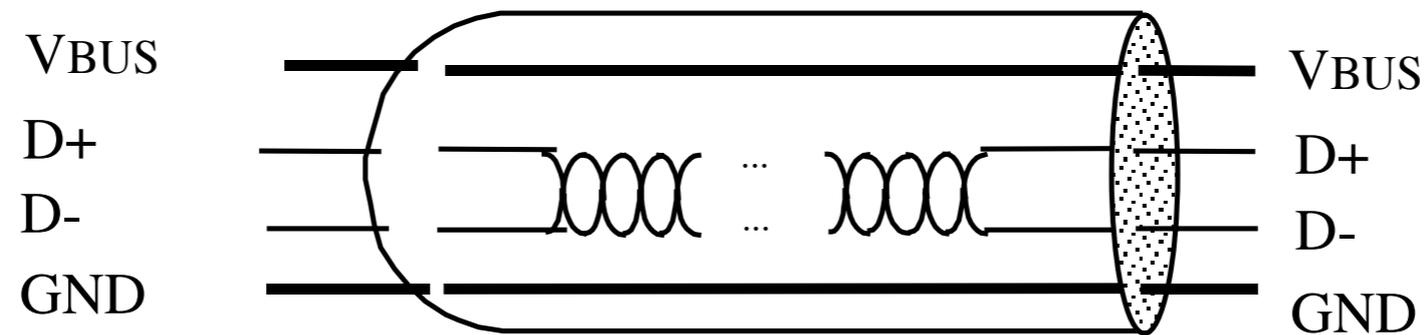
Circa 1997



Circa maintenant



Matériel et connecteurs



- Le câble USB est constitué de 4 fils:

Vbus est l'alimentation 5Vdc (entre 4.75V et 5.25V)

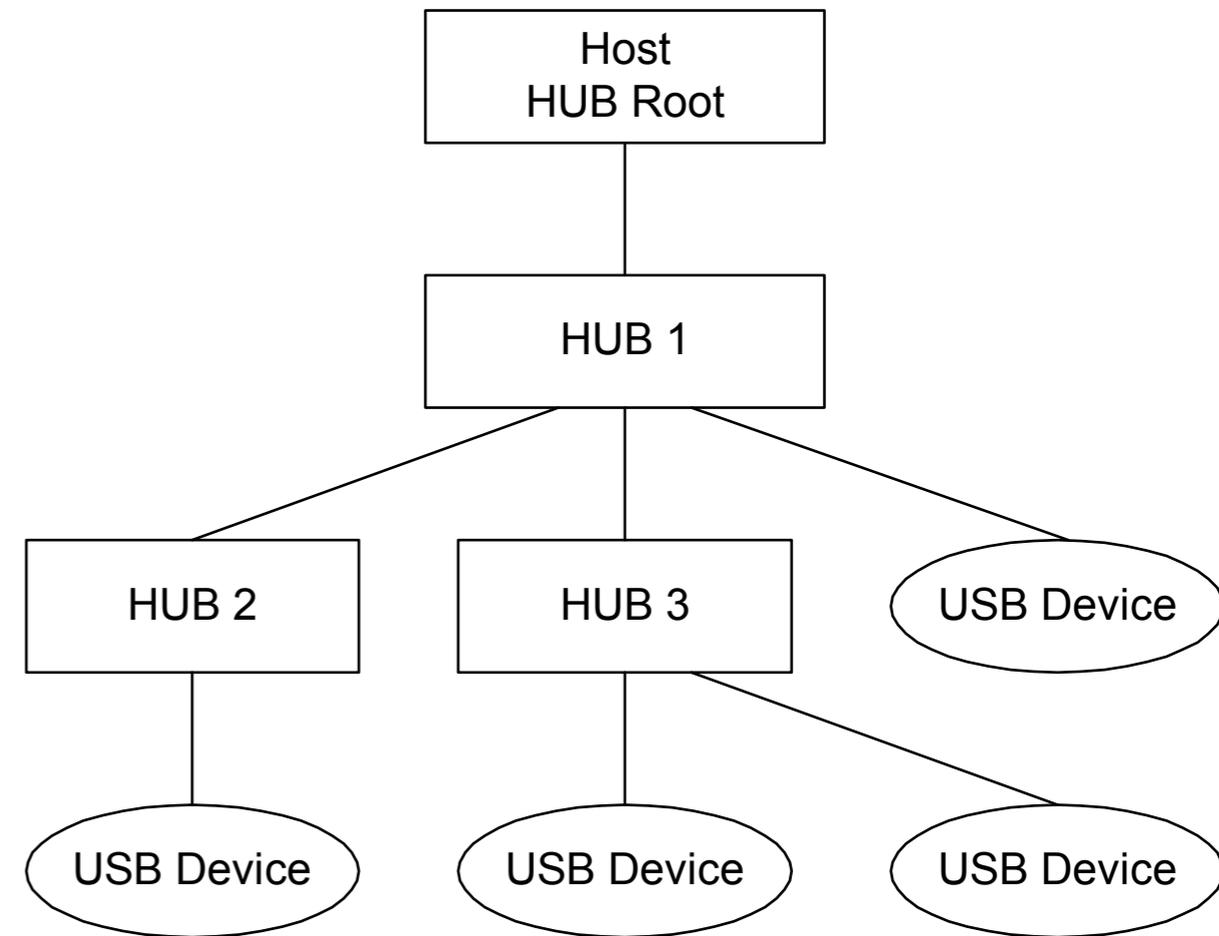
(peut alimenter les appareils branchés sur le bus!)

D- et **D+** servent au transport des données.

GND est la référence électrique

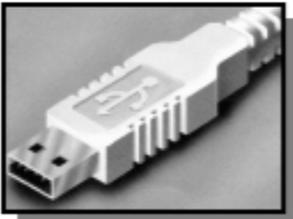
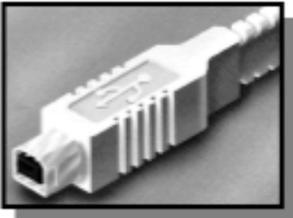
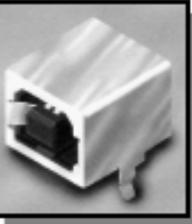
Topologie d'un réseau USB

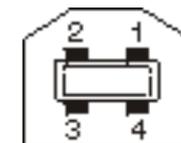
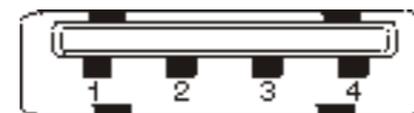
- Un réseau USB a une topologie en étoile.
- Le port USB est contrôlé entièrement par un contrôleur unique appelé hôte (*host*). Souvent l'ordinateur, il initie toutes les communications, et est le **maître absolu** du bus.
- Les “hubs” permettent de relier plusieurs appareils à un seul port USB.
 - Le rôle principal des hubs est de transférer les données de l'hôte aux périphériques.
 - Chaque hub contrôle ses ports afin de savoir si un appareil s'y connecte
 - Il peut y avoir 5 niveaux de hub en plus du hub racine.
- Il y a 127 appareils maximum dans un réseau USB (excluant l'hôte). Chaque appareil a son adresse.



Matériel—connecteurs

- Il y a deux types de connecteurs USB: A et B.
- Pour garantir la topologie étoile, les connecteurs:
 - A “pointent” toujours vers le haut, vers l’hôte
 - B “pointent” toujours vers le bas, vers les périphériques

| Series "A" Connectors | Series "B" Connectors |
|---|--|
| <ul style="list-style-type: none">◆ Series "A" plugs are always oriented upstream towards the <i>Host System</i>  <p>"A" Plugs (From the USB Device)</p>  <p>"A" Receptacles (Downstream Output from the USB Host or Hub)</p> | <ul style="list-style-type: none">◆ Series "B" plugs are always oriented downstream towards the <i>USB Device</i>  <p>"B" Plugs (From the Host System)</p>  <p>"B" Receptacles (Upstream Input to the USB Device or Hub)</p> |



Tiré de usb_20.pdf
(<http://www.usb.org/developers/>)
<http://www.networktechinc.com/technote.html>

Matériel — connecteurs



micro-B

mini-B

standard-A
(femelle)

standard-A
(mâle)

standard-B

RS-232: on « emballe » les données



Bit de départ

Bit d'arrêt

Bit de parité

USB: on « emballe » aussi les données!



Plusieurs bits de départ

PID, adresse

CRC

USB: paquets

- Un paquet est divisé en plusieurs parties:
 - Transition (“SYNC”): 10101010 en succession rapide pour synchroniser les horloges
 - PID (*packet identifier*): indique le *type* de paquet
 - Données: données à transmettre
 - Adresse: appareil à qui s’adresse ce paquet
 - Correction d’erreur (*Cyclic Redundancy Check*, ou CRC)



↑
Bit de parité sur les stéroïdes

USB: paquets

- Il existe plusieurs types de paquets. Par exemple:
 - **IN**: indique le démarrage d'un transfert de données d'un périphérique vers l'hôte
 - **OUT**: indique le démarrage d'un transfert de données de l'hôte vers le périphérique
 - **DATA0**: contient les données à transférer
 - **ACK**: indique que les données ont été reçues sans erreur
 - **SETUP**: indique un envoi de données de configuration

| | | | | |
|------------|------------|---------|---------|-----|
| Transition | PID | Données | Adresse | CRC |
|------------|------------|---------|---------|-----|

Légende:

hôte vers
appareil

appareil vers
hôte

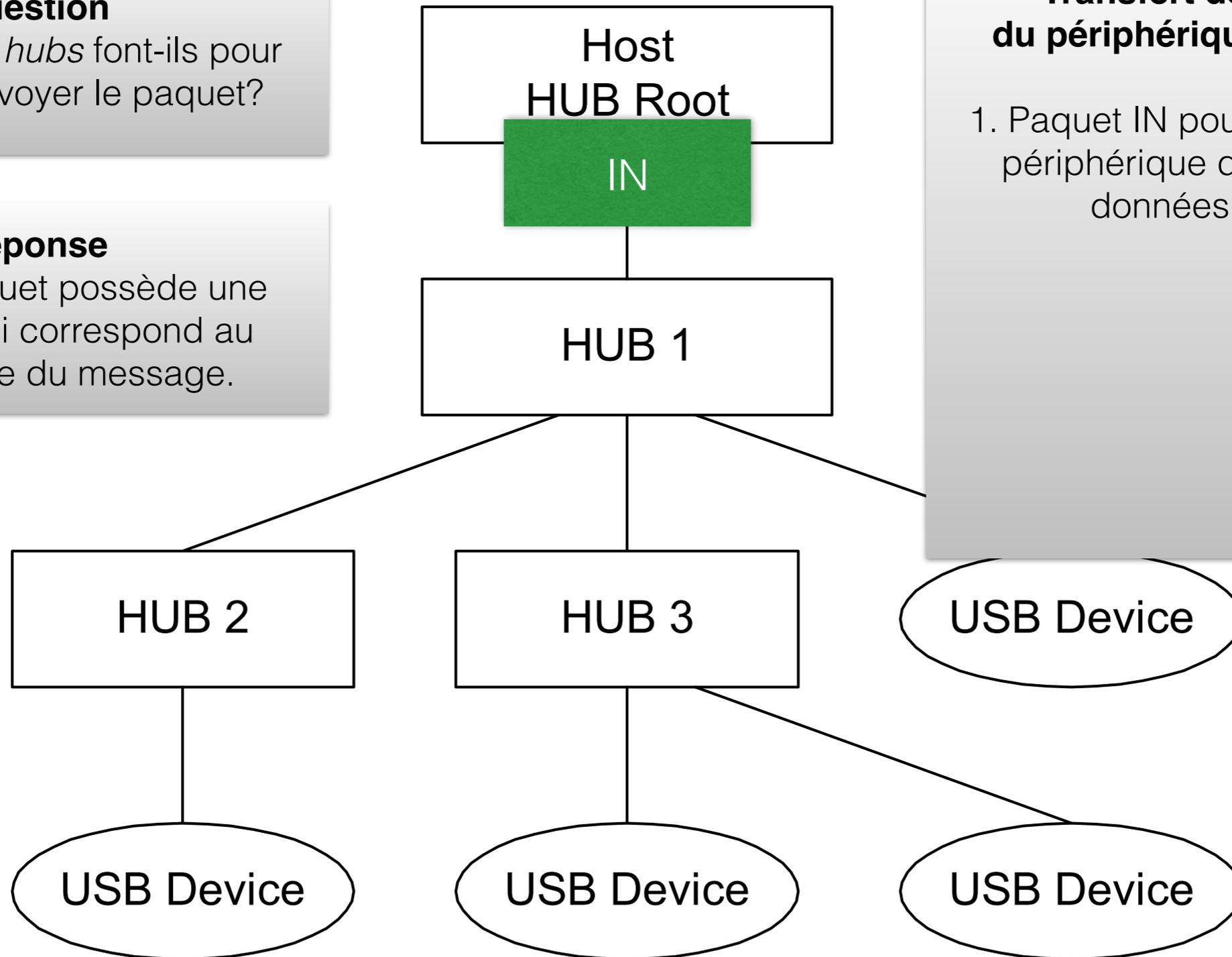
Exemple de transaction

Question

Comment les *hubs* font-ils pour savoir où envoyer le paquet?

Réponse

Chaque paquet possède une adresse, qui correspond au destinataire du message.

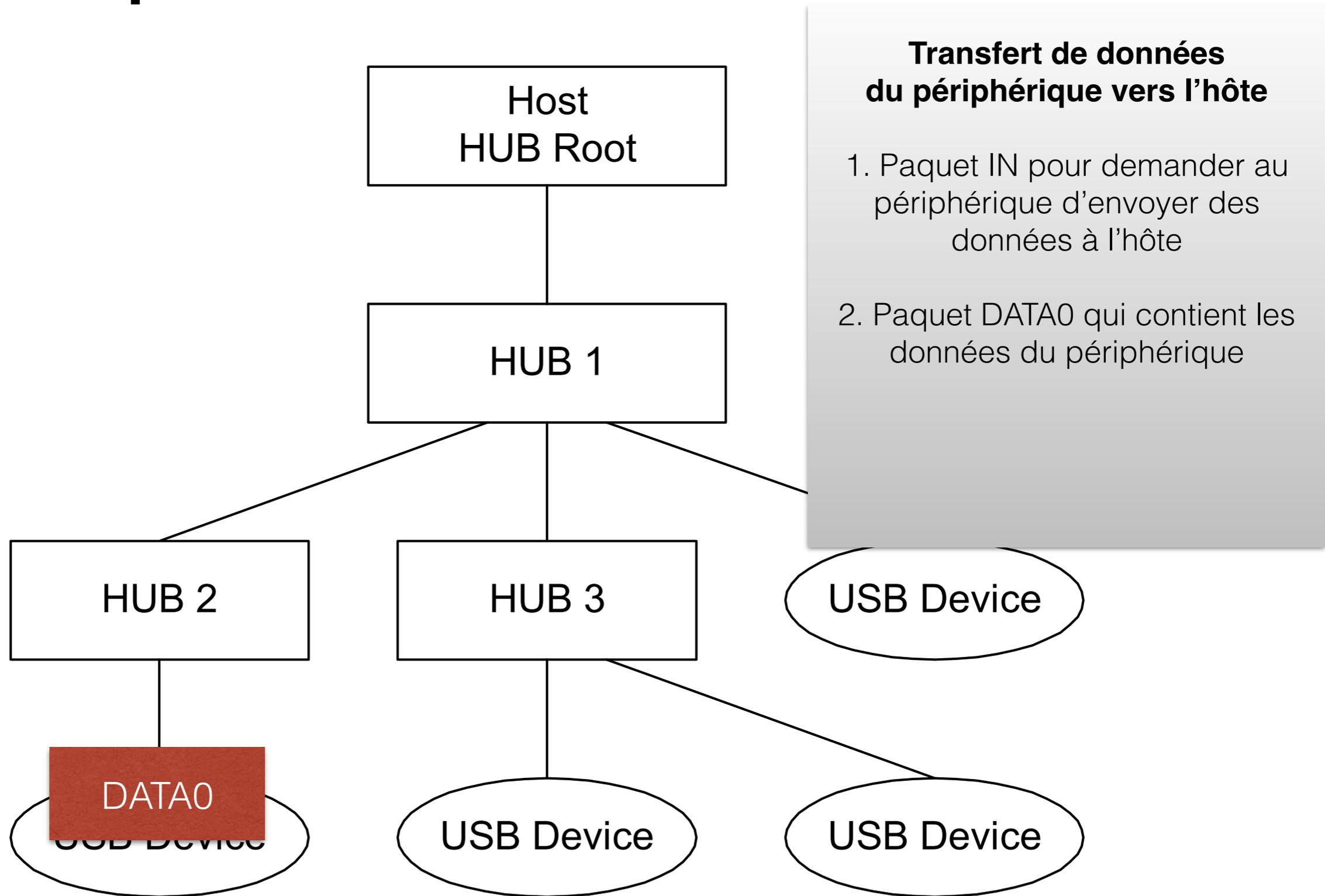


Légende:

hôte vers
appareil

appareil vers
hôte

Exemple de transaction

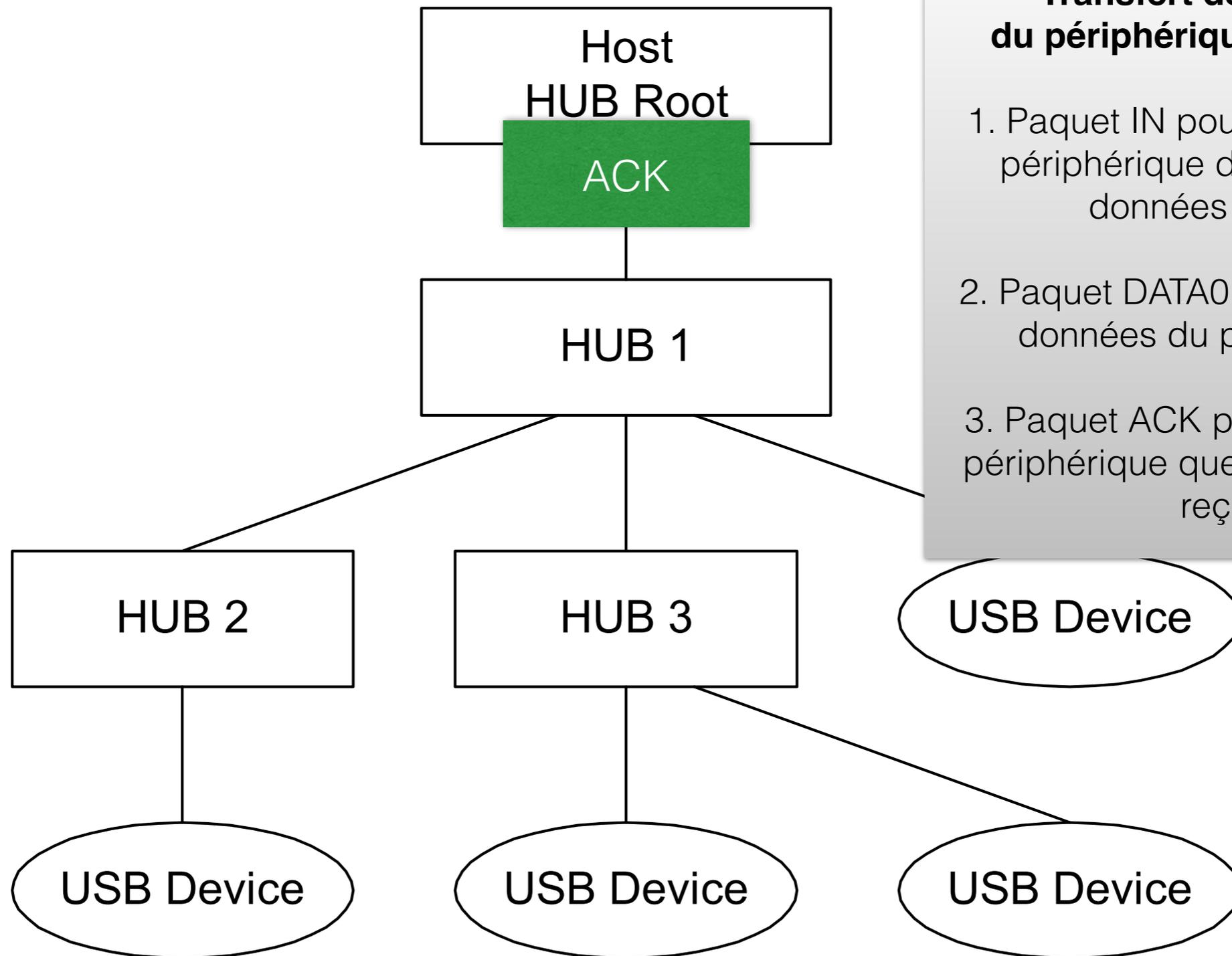


Légende:

hôte vers
appareil

appareil vers
hôte

Exemple de transaction



Transfert de données du périphérique vers l'hôte

1. Paquet IN pour demander au périphérique d'envoyer des données à l'hôte
2. Paquet DATA0 qui contient les données du périphérique
3. Paquet ACK pour indiquer au périphérique que tout a été bien reçu.

Légende:

hôte vers
appareil

appareil vers
hôte

Protocole de communication

- Exemples de transactions:

- Transfert de données, du périphérique vers l'hôte



- Transfert de données, de l'hôte vers le périphérique



- Configuration, de l'hôte vers le périphérique



↑
Token packet

↑
Data packet

↑
Handshake packet

Échange de données: transactions

- C'est l'hôte qui initie **tous** les transferts de données.
- La plupart des transactions nécessite l'envoi de 3 paquets:
 1. **Token packet**: l'hôte envoie un paquet décrivant le type, et la direction de la transaction.
 - Le paquet contient:
 - L'adresse de l'appareil USB
 - Le numéro de terminaison sur cet appareil
 2. **Data packet**: L'appareil USB correspondant s'active en fonction de l'adresse reçue. Un deuxième paquet est envoyé:
 - Le paquet contient les données correspondant à la transaction demandée,
 - Le paquet est envoyé selon la direction de la transaction (hôte vers appareil, ou appareil vers hôte),
 3. **Handshake packet**: l'appareil de destination indique si la transaction a été complétée avec succès ou non.

Transfert

- En plus des différents types de paquets, la norme USB définit:
 - différents types de transfert
 - l'ordre (temporel) dans lequel ces transferts sont effectués

Types de transfert

- Le USB supporte 4 types de transfert:
 - **de contrôle**: sert à la configuration et à la commande d'un appareil. Il est effectué à partir de la terminaison 0.
 - **isochrone**: est un mode de transfert pour lequel les données sont transmises à vitesse constante, et garantie. Idéal pour les flux de données («streaming»).
 - **par interruption**: est utilisé par les appareils ayant peu de données à transmettre, mais ayant des données qui doivent être transmises rapidement (exemple: clavier ou souris). Ce ne sont pas de «vraies» interruptions: elles sont détectées par interrogation successive (*polling*) de provenance l'hôte. La fréquence de ces interrogations est donnée par les descripteurs de l'appareil.
 - **par bloc**: permet de transférer des volumes importants de données lorsqu'il n'y a pas de contraintes temporelles (exemple: imprimante).

Trames USB: ordre temporel

- Le **SOF** (ou *Start of Frame*) indique le départ d'une trame USB
- On commence tout d'abord par les transferts **isochrones**
 - comme leur taux de transfert doit être garanti, on les met en priorité.
- Ensuite, les « **interruptions** » (simulées)
- Finalement, les transactions restantes (de **contrôle** et en **bloc**) sont les moins prioritaires, on les garde donc pour la fin (s'il reste du temps).
 - Elles sont sélectionnées par tourniquet (chacun son tour).

| | | | |
|-----|------------|---------------|------------------------|
| SOF | Isochrones | Interruptions | de contrôle et en bloc |
|-----|------------|---------------|------------------------|