

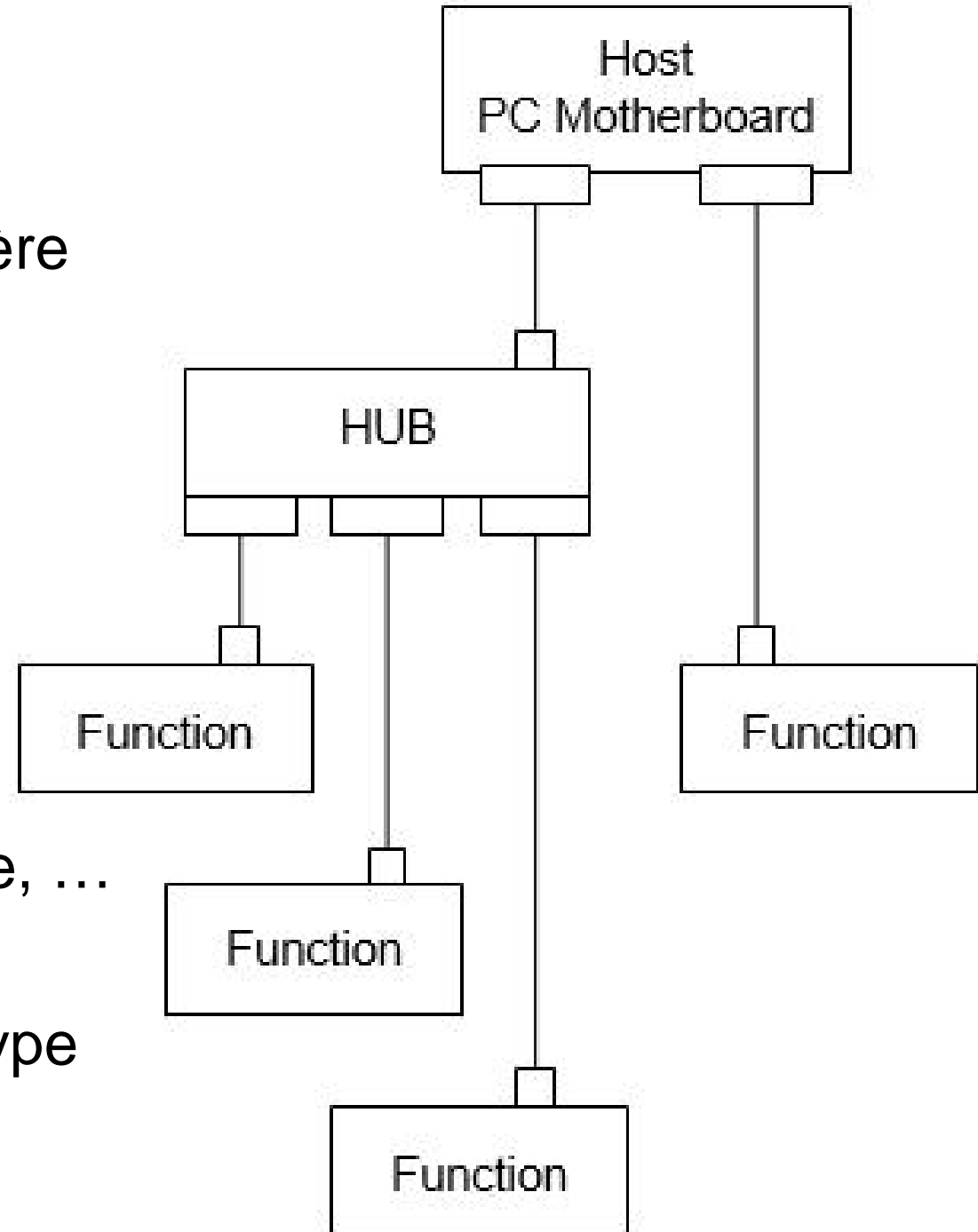
# Universal Serial Bus (USB)



- 1996 Spécification USB 1.0 (12 / 1.5 Mbit/s)
- 1997 Travail de diplôme avec du matériel prêté par Logitech qui émule une carte mère avec port USB
- 1997 Premier PC Compaq Presario compatible USB au labo
- 1998 Windows 98 supporte les premiers périphériques USB
- 1999 Mandat du groupe Swatch
- 2000 Spécification USB 2.0
- 2002 [Ellisys](#) (founded by 2 EIG students) presents [Tracker 110](#)
- 2008 Spécification USB 3.0
  
- <http://en.wikipedia.org/wiki/USB>

# 3 niveaux hierarchiques

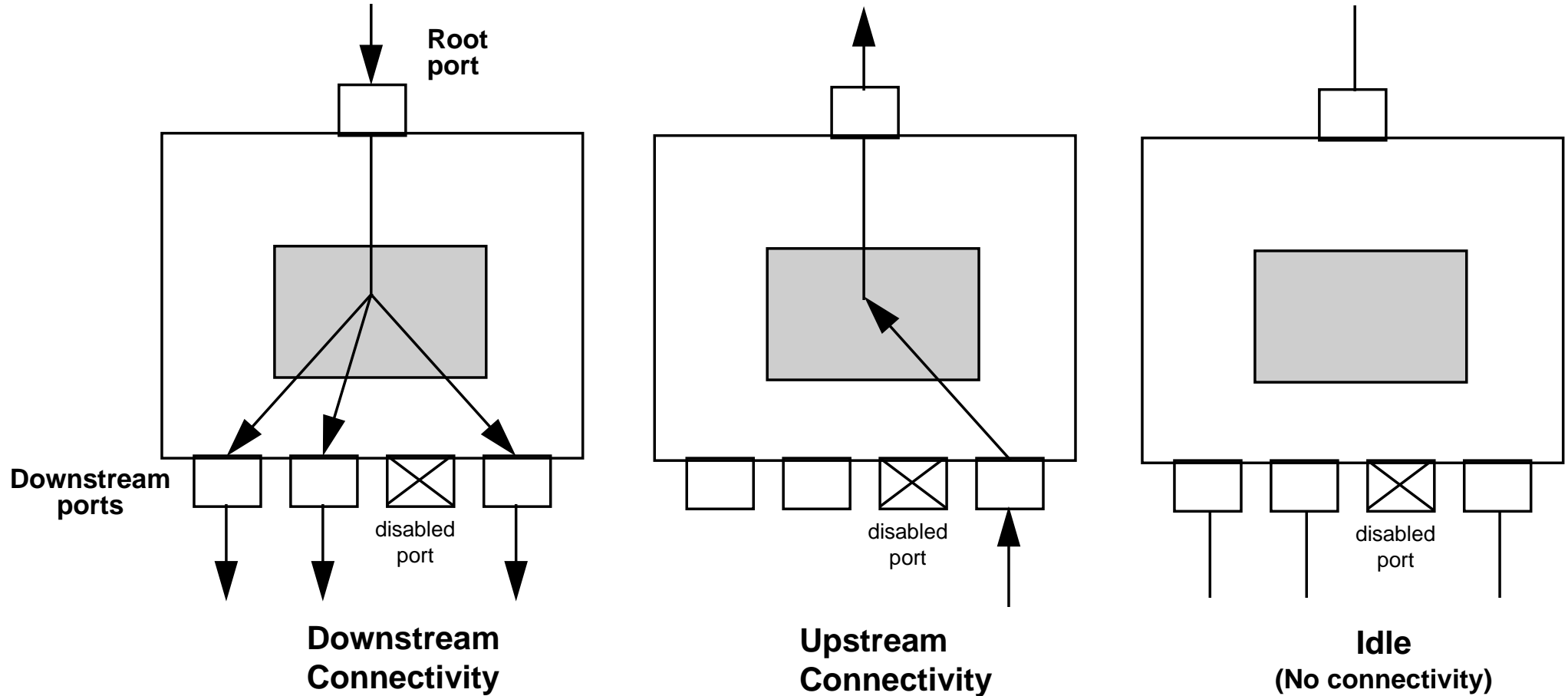
- **Host**  
Contrôleur USB sur la carte mère
- **Hub**  
Module d'interconnexion  
Analogie avec hub Ethernet
- **Function**  
Périphérique = souris, mémoire, ...
- Toutes les connexions sont de type point à point



# ***Fonctionnement***

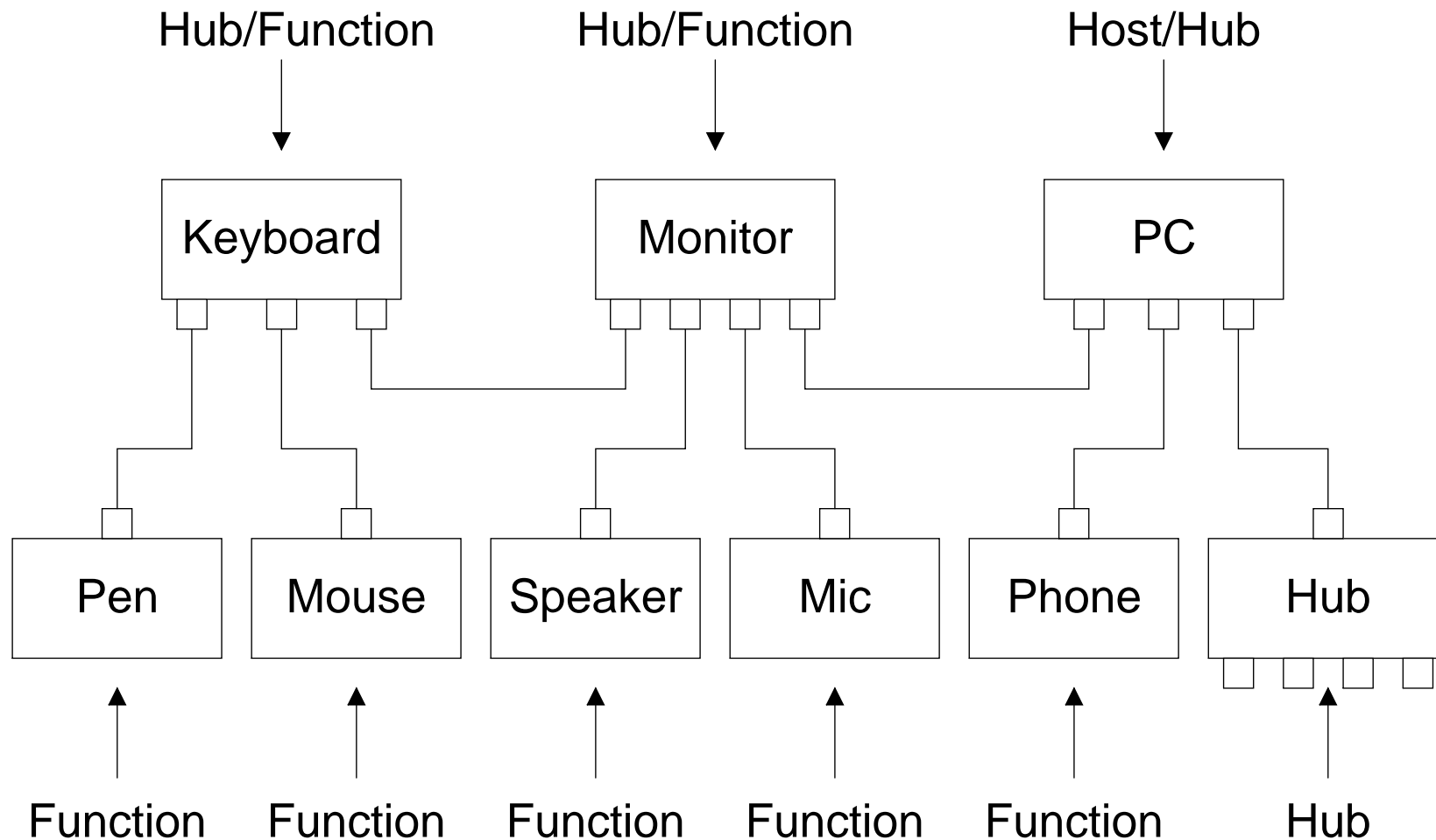
- Support de transmission partagé (shared medium)
- Topologie étoile produite à l'aide de hub (répéteur)
- Méthode d'allocation centralisée  
Host Controller = chef d'orchestre
- Une adresse par périphérique (max de 127)  
Analogie avec l'adresse ethernet = adresse physique
- Insertion et déconnexion dynamiques  
Attribution dynamique de l'adresse  
Raccordement du périphérique possible à tout instant
- Gestion de l'alimentation (power management)  
Téléalimentation possible

# Flux montant & descendant



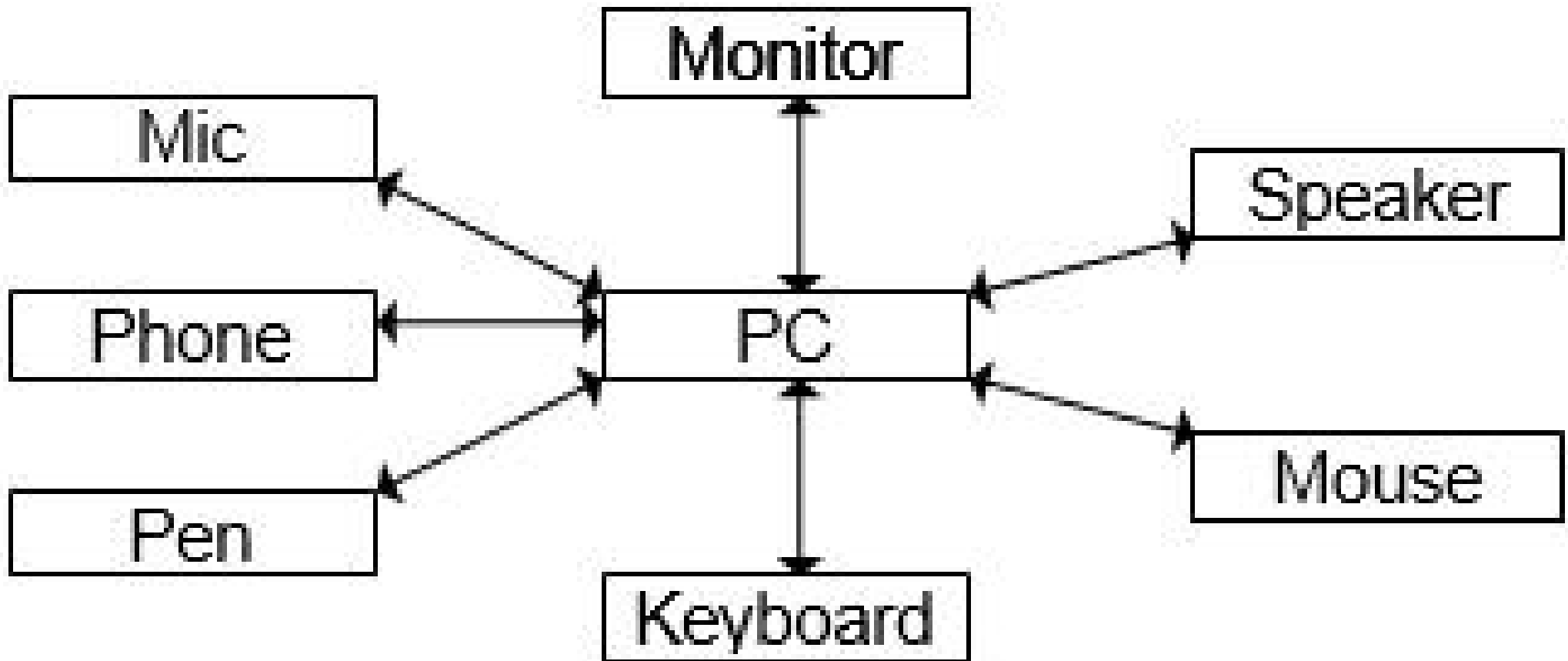
- Le flux descendant (downstream) est diffusé
- Le flux montant (upstream) n'est pas visible des autres périphériques

# Exemple de configuration physique



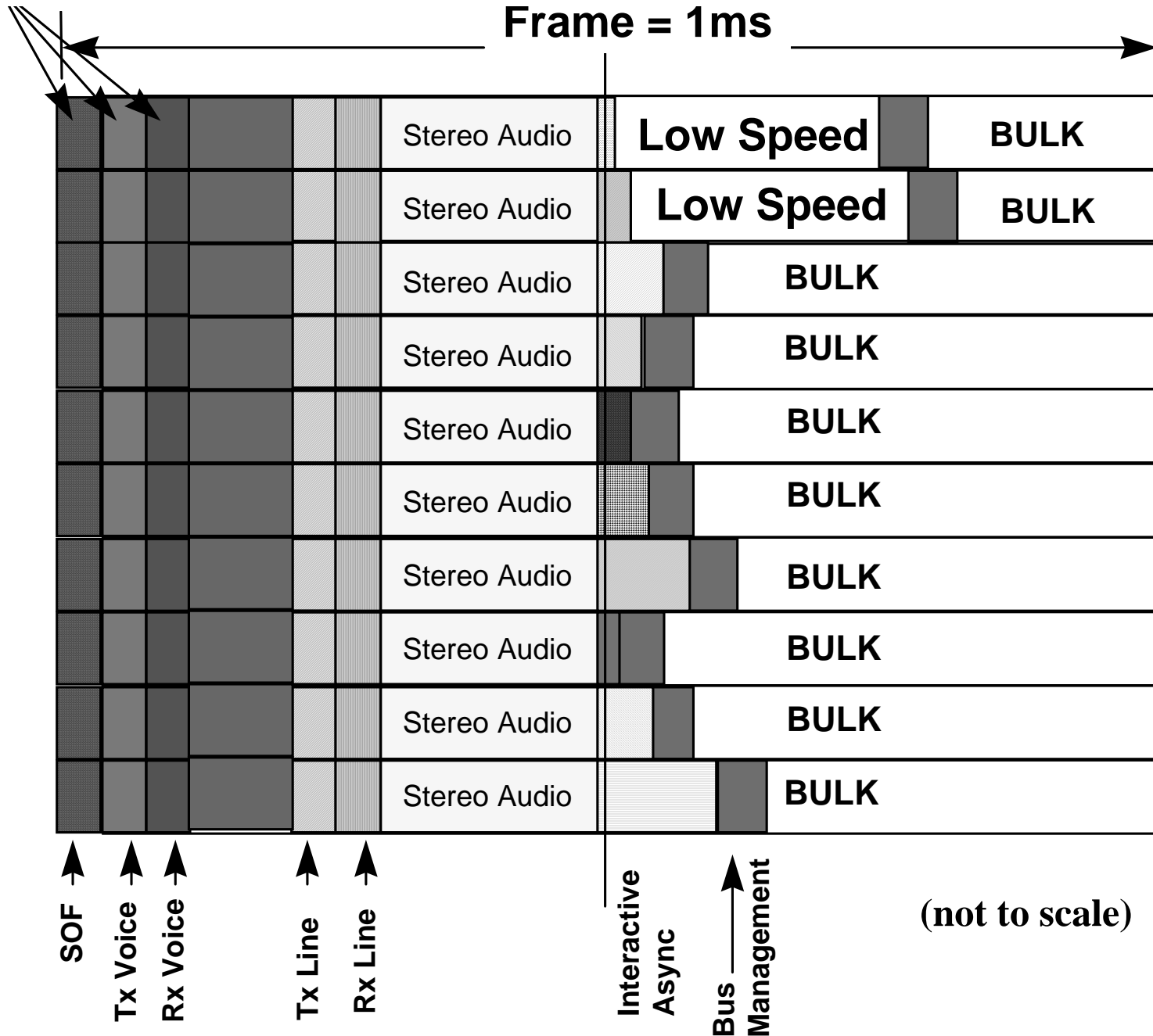
- Host + Hub dans PC
- Fonction + Hub dans Monitor et Keyboard
- Mise en cascade de hubs (6 au max)

# Exemple de configuration physique



- Le chef d'orchestre (Host controller) contrôle l'accès des périphériques

# Méthode d'accès & Flux de données



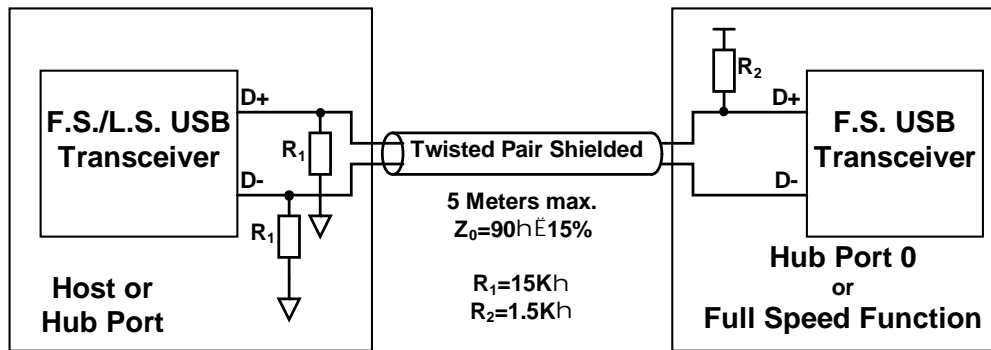
# Méthode d'accès & Flux de données (suite)

- **Host génère une marque (Start Of Frame) chaque ms**
- Ordre de priorité est défini pour garantir certains flux
  - 1 **Isochronous**      Flux garanti      1 par ms  
Exemple = streaming de 5 Mbit/s, flux vidéo
  - 2 **Interrupt**      Flux garanti      1 pour n ms  
Exemple = souris qui est interrogée chaque 20 ms
  - 3 **Control**      Flux non garanti  
Utile lors du branchement du périphérique → énumération
  - 4 **Bulk**      Flux non garanti  
Exemple = imprimante, mémoire flash

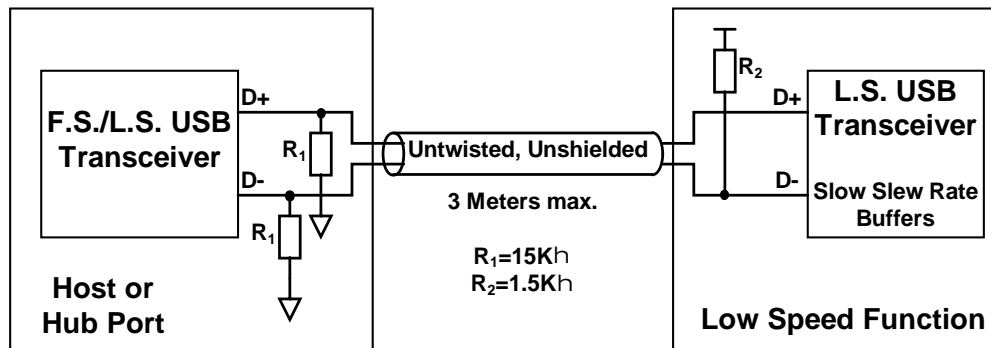


# Full & Low Speed

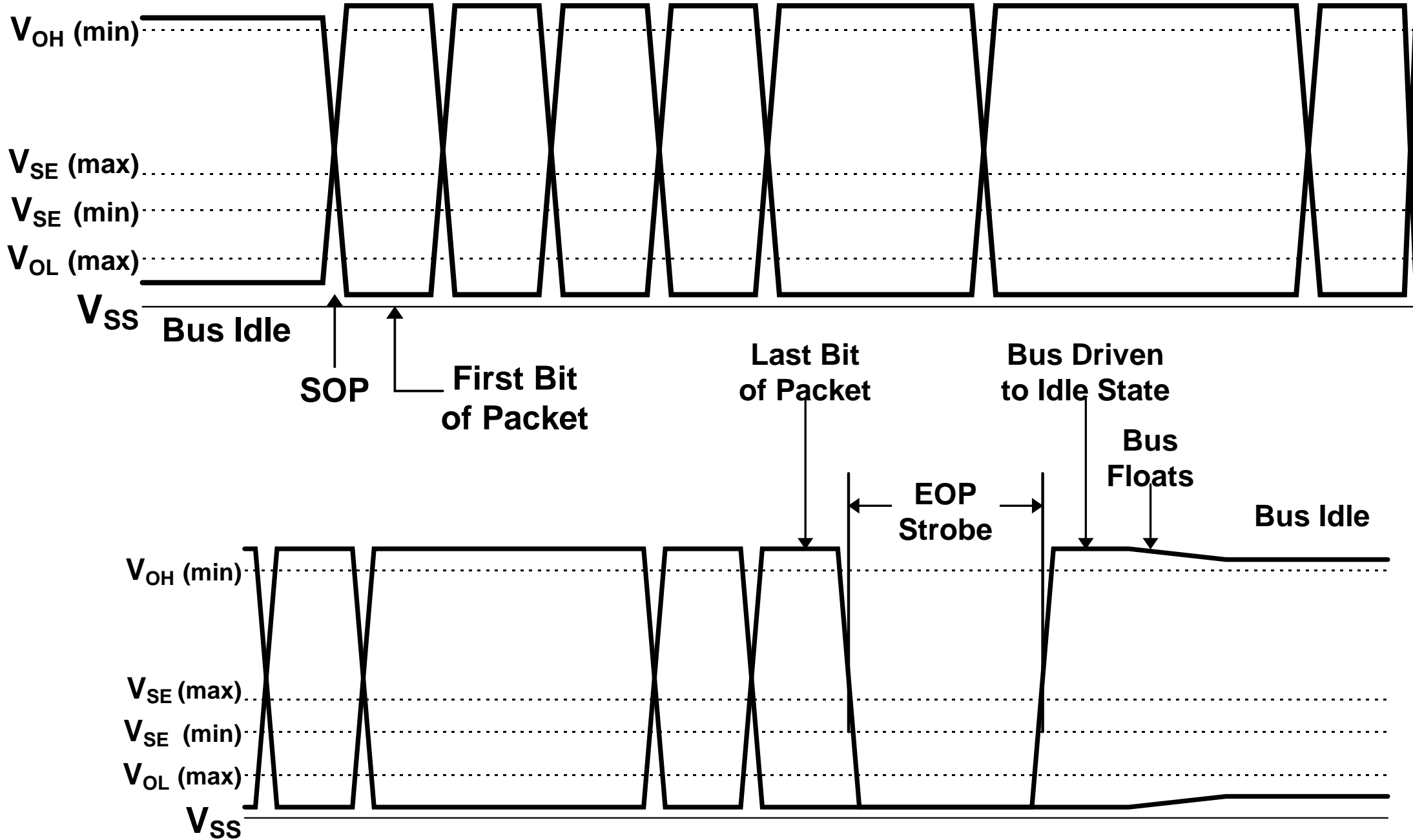
- La spécification 1.0 définit 2 débits possibles
- **Full Speed = 12 Mbit/s** → majorité des périphériques en 2000



- **Low Speed = 1.5 Mbit/s** → souris, ...



# Signal électrique

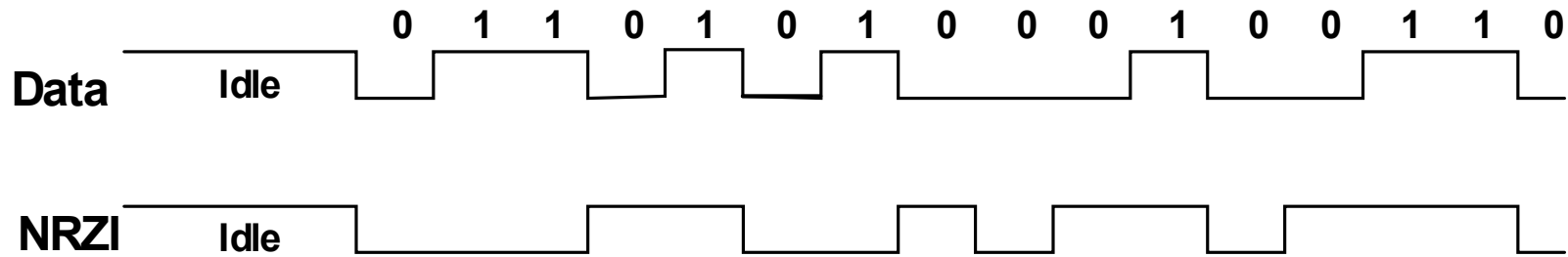


# Signal électrique (suite)

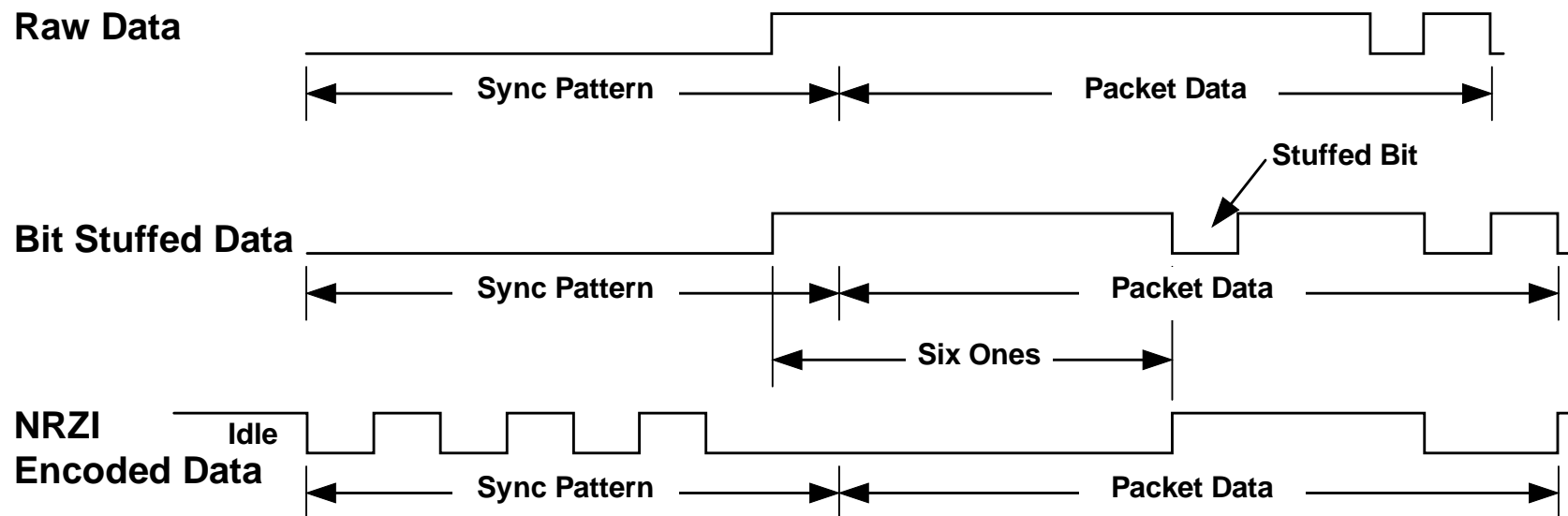
- Mode différentiel
  - "1" (D+) - (D-) > 200 mV
  - "0" (D+) - (D-) < -200 mV
  
- J            "0" (low speed)            "1" (full speed)
- K            "1" (low speed)            "0" (full speed)
- Idle        "0" (low speed)            "1" (full speed)
  
- SOP (Start Of Packet)            idle to K transition
- EOP (End Of Packet)            D+ and D- < VSE for 2 bits followed by idle

# Codage

- NRZI "1" no transition "0" transition

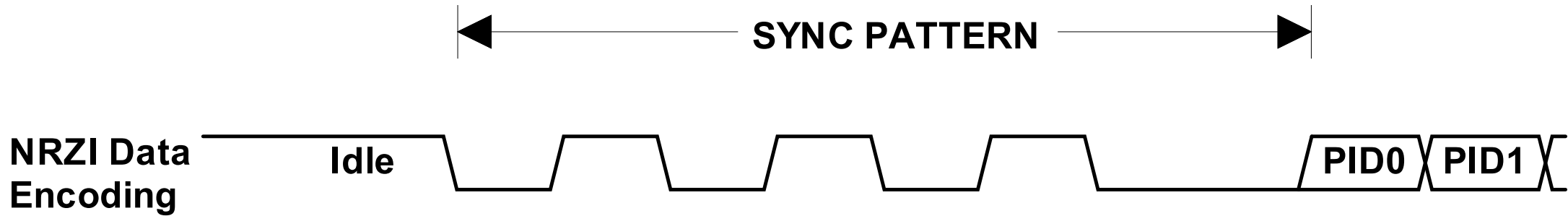


- Bit stuffing : un "0" est ajouté après 6 "1" consécutifs  
Data Encoding Sequence:



# Synchronisation

- Chaque paquet commence avec le champ SYNC = KJKJKJKK



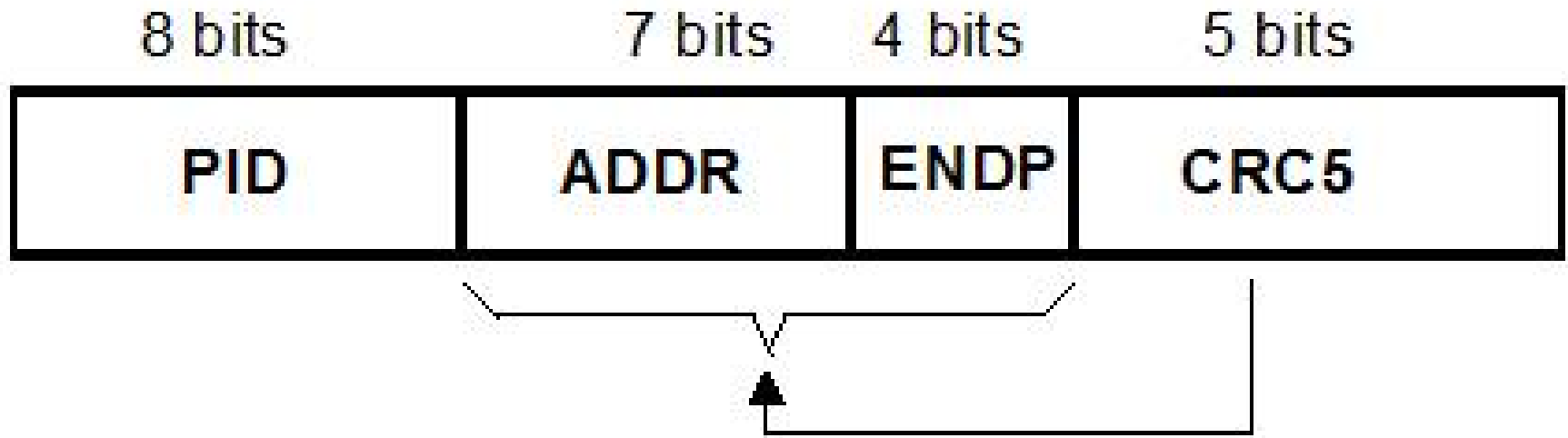
- Le champ PID suit le champ SYNC → next slide

# Format des paquets (1/3)

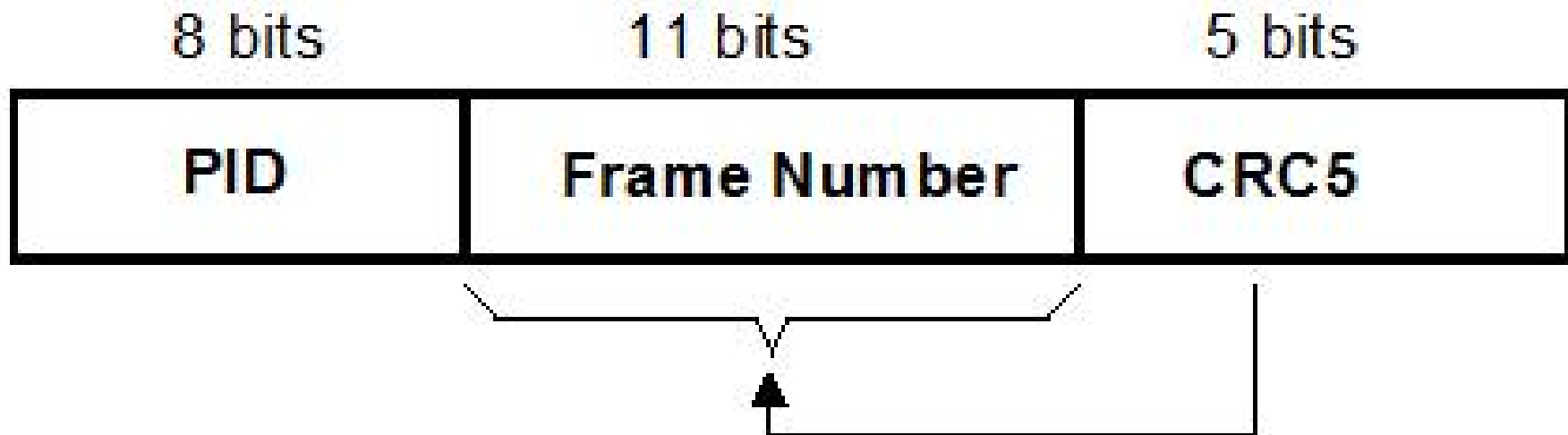
PID Type	PID Name	PID[3:0]	Description
Token	OUT	b0001	Address + endpoint number in host -> function transaction
	IN	b1001	Address + endpoint number in function -> host transaction
	SOF	b0101	Start of frame marker and frame number
	SETUP	b1101	Address + endpoint number in host -> function transaction for setup to a control endpoint
Data	DATA0	b0011	Data packet PID even
	DATA1	b1011	Data packet PID odd
Handshake	ACK	b0010	Receiver accepts error free data packet
	NAK	b1010	Rx device cannot accept data or Tx device cannot send data
	STALL	b1110	Endpoint is stalled
Special	PRE	b1100	Host-issued preamble. Enables downstream bus traffic to low speed devices.

# Format des paquets (2/3)

- Token Packet

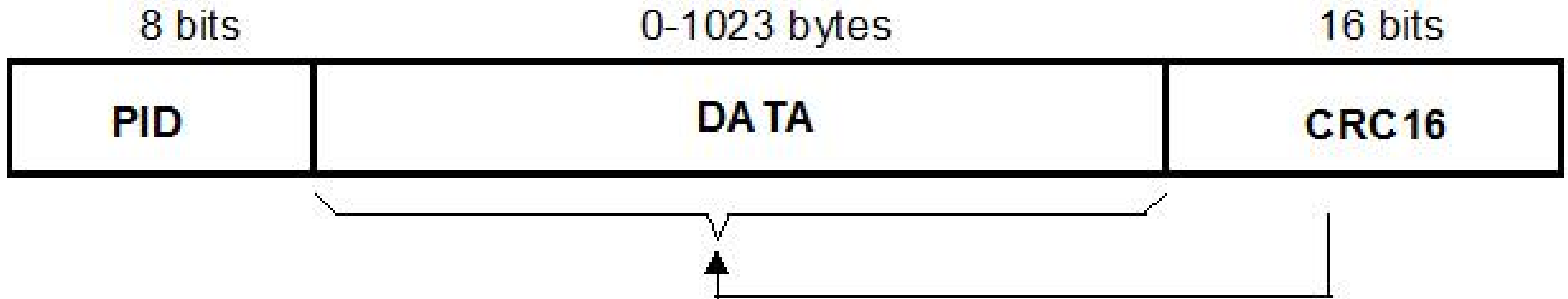


- SOF (Start Of Frame)



# Format des paquets (3/3)

- Data Packet

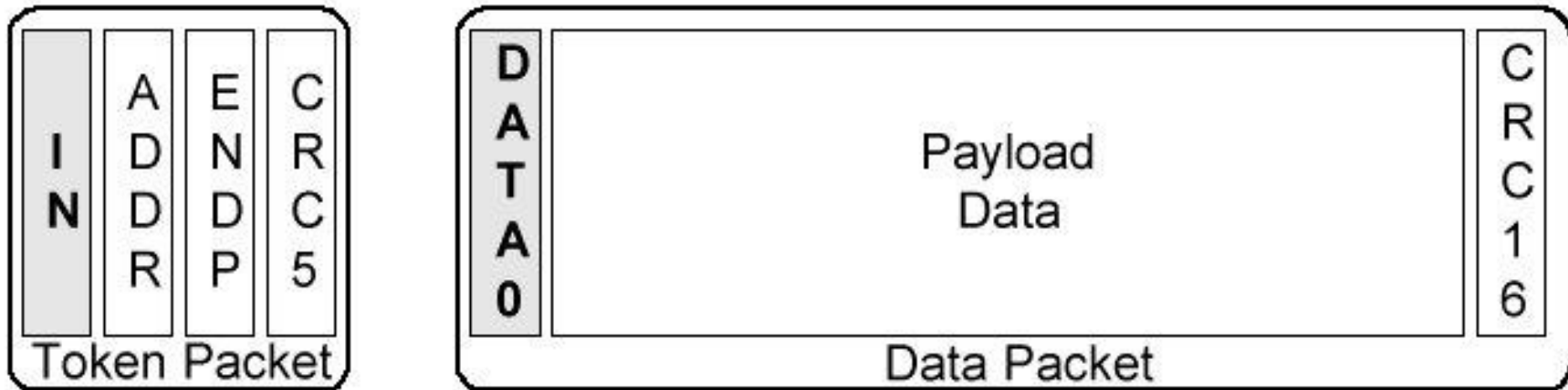


- Handshake Packet





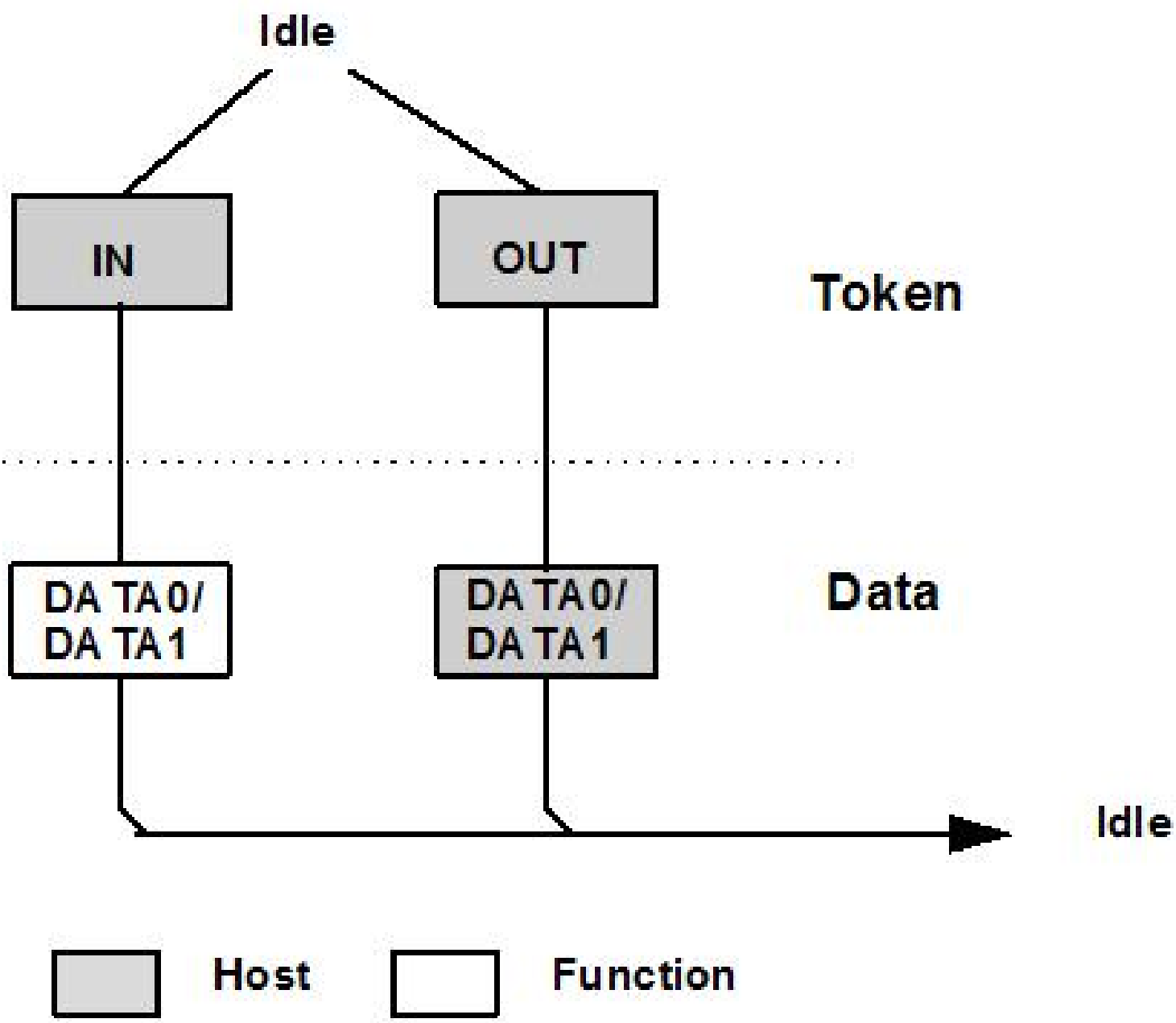
# Isynchronous transaction



- Transaction composée de 2 paquets
- Le Host donne le jeton IN (lecture) au périphérique ADDR et au pipe ENDP (Endpoint) → analogie avec port TCP-UDP
- Le périphérique génère la Data Packet

# Isochronous transaction (suite)

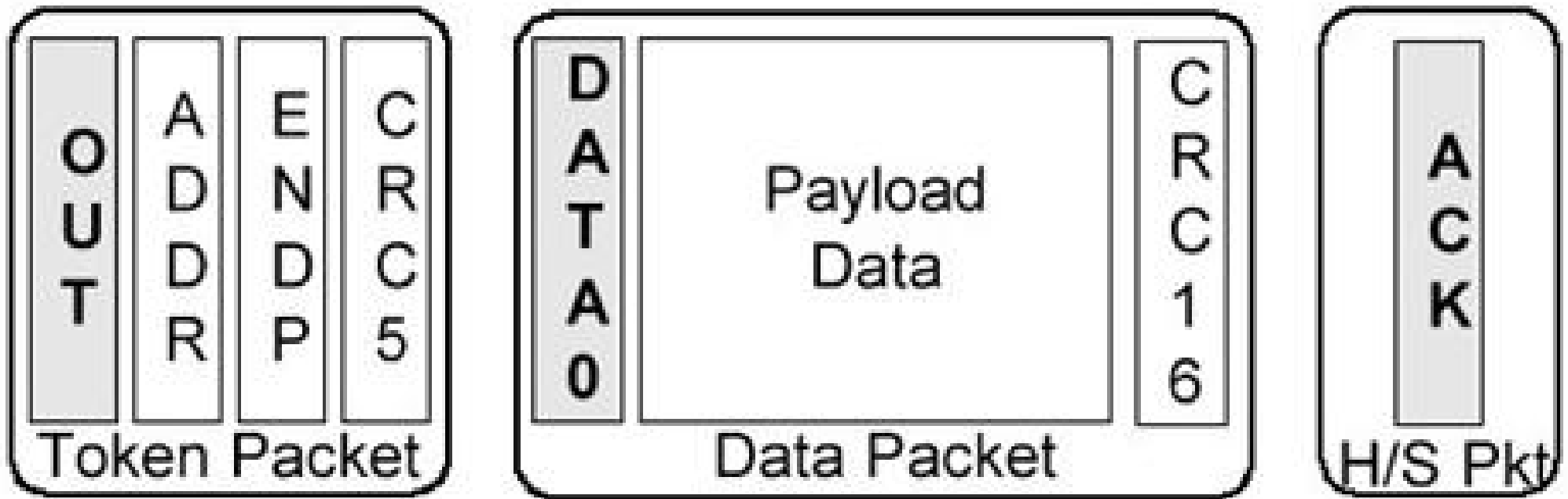
- Les paquets grisés sont émis par le Host



# *Isochronous transfer type*

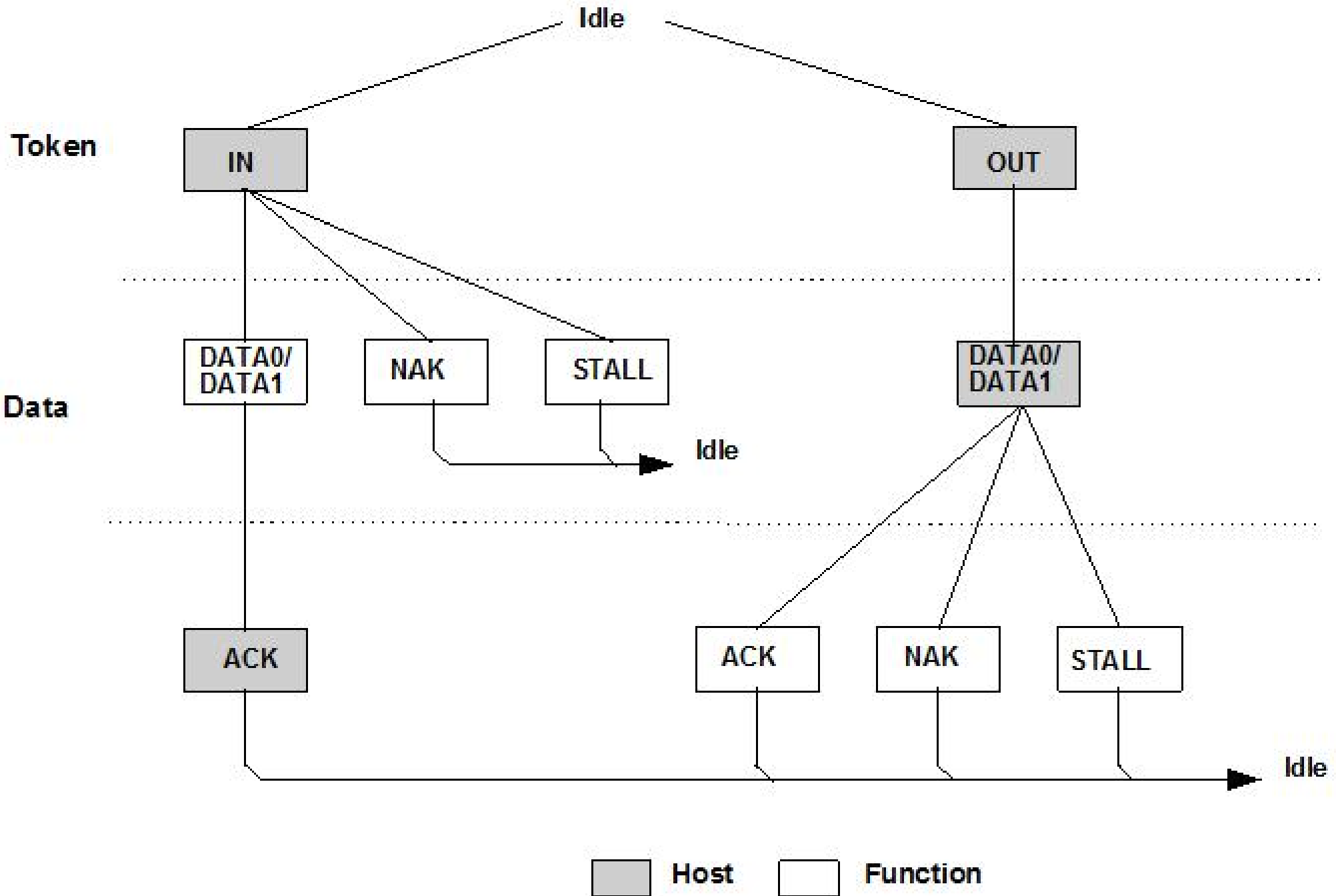
- Echange unidirectionnel des données utiles
- 2 pipes sont donc nécessaires pour un échange bidirectionnel
- Taille max de 1023 bytes
- Limité au full speed device (12 Mbit/s)
  - Les périphériques de type Low Speed ne peuvent pas utiliser ce type de flux

# Bulk transaction



- Transaction composée de 3 paquets
- Le Host donne le jeton OUT (écriture) au périphérique
- Le Host génère le Data Packet
- Le périphérique envoie la confirmation (Handshake Packet)

# Bulk transaction (suite)



# ***Bulk transfer type***

- Echange unidirectionnel des données utiles
- 2 pipes sont donc nécessaires pour un échange bidirectionnel
- Taille de 8 / 16 / 32 / 64 bytes
- Limité au full speed device (12 Mbit/s)

# Ex 1

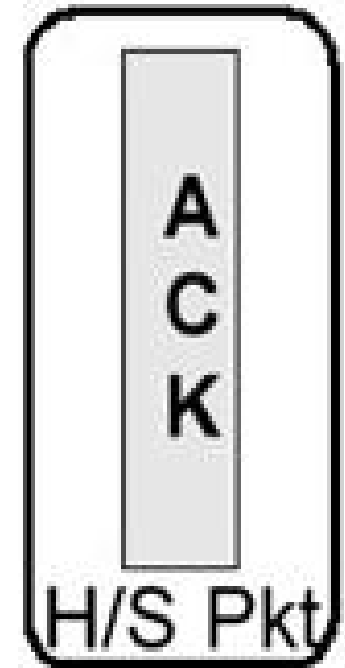
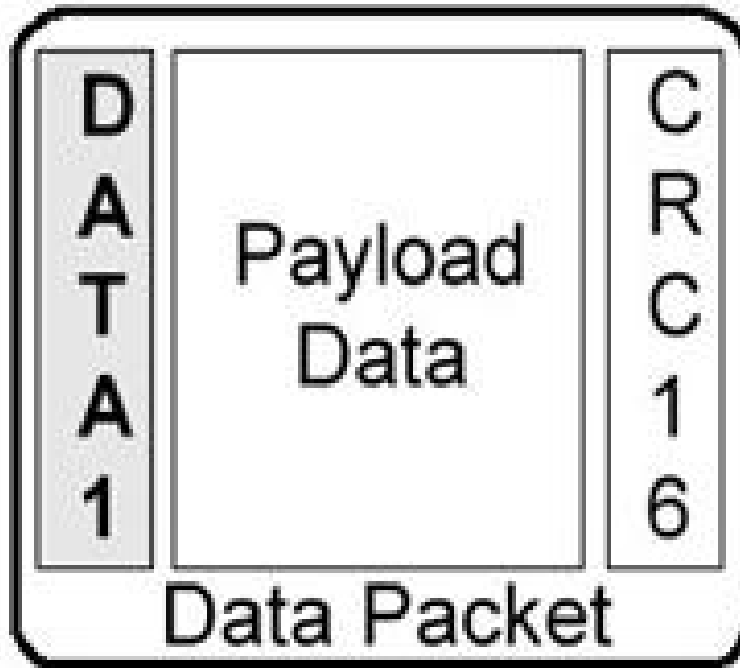
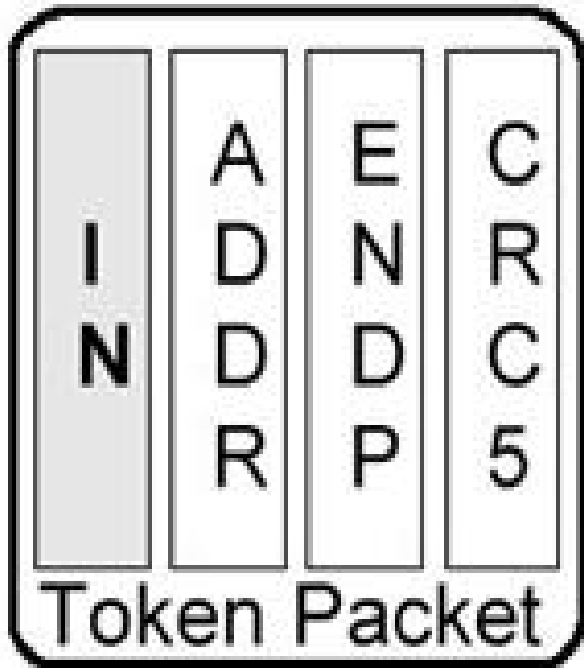
- Déterminer à partir de la figure précédente toutes les transactions possibles
- Préciser qui génère chaque paquet en utilisant un diagramme en flèches
- Expliquer chaque cas :
  - Lecture sans erreur
  - Erreur de transmission
  - Tampon vide
  - Périphérique défectueux
  - ...

## ***Ex 2***

- Calculer le débit utile max d'un flux isochrone unidirectionnel
- Calculer l'overhead

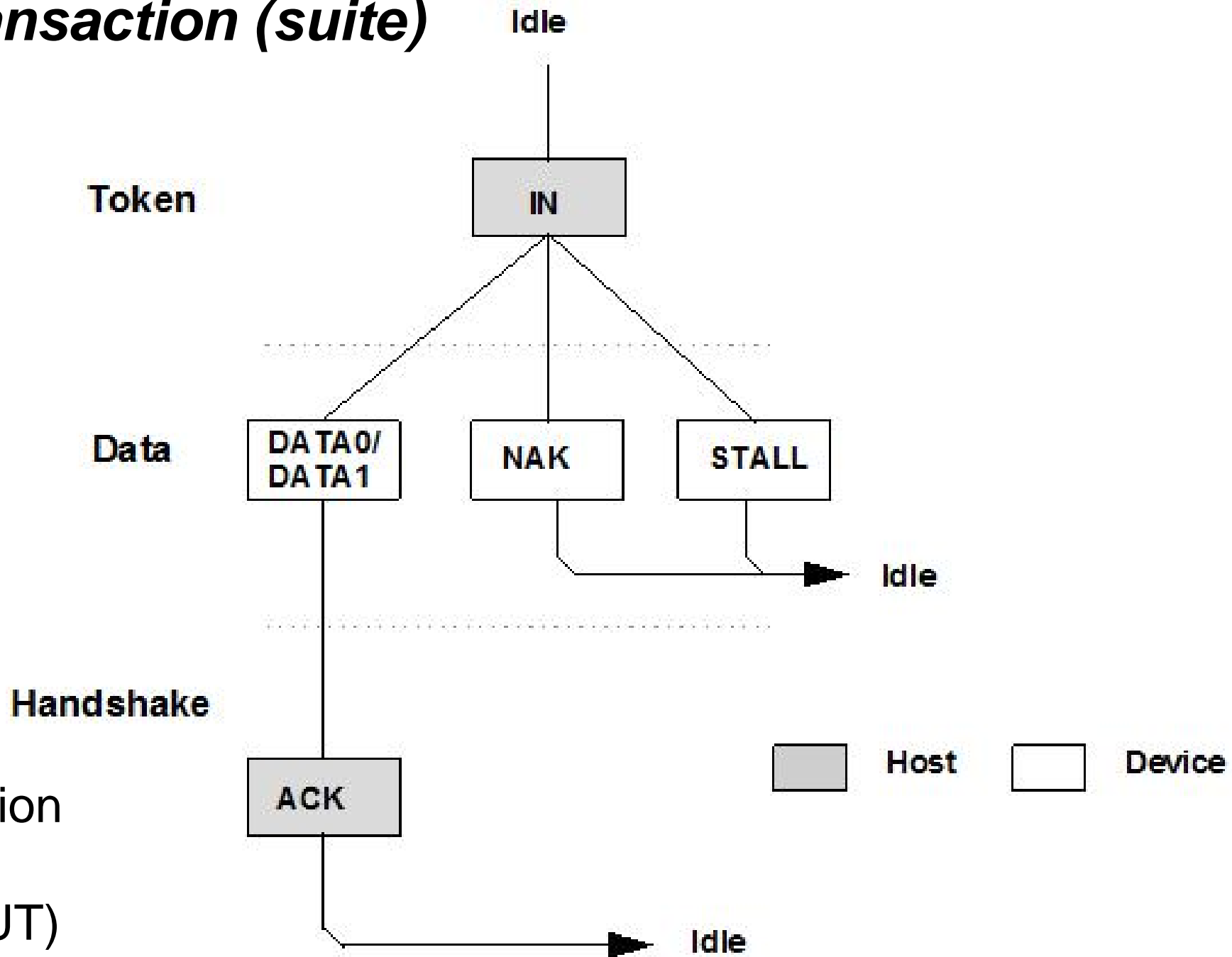


# Interrupt transaction



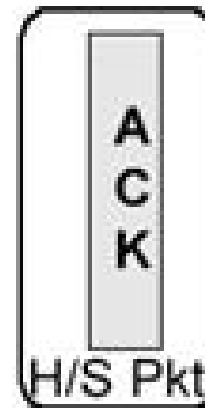
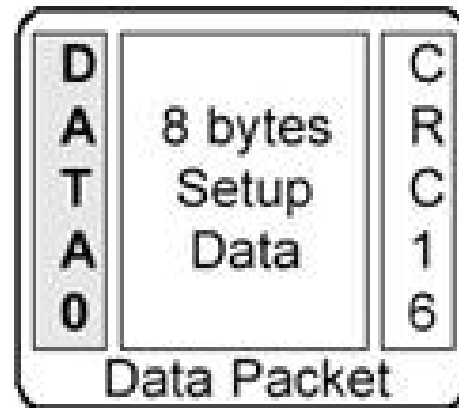
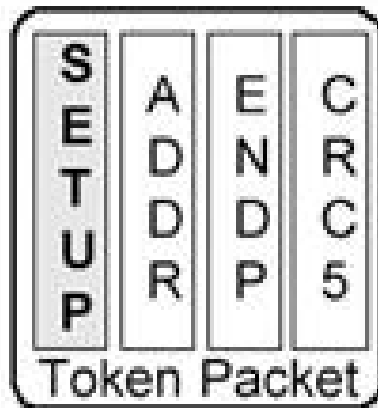
- Transaction composée de 3 paquets
- Le Host donne le jeton OUT (écriture) au périphérique
- Le Host génère le Data Packet
- Le périphérique envoie la confirmation (Handshake Packet)

# Interrupt transaction (suite)

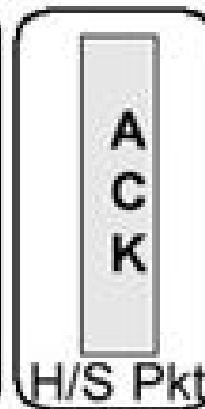
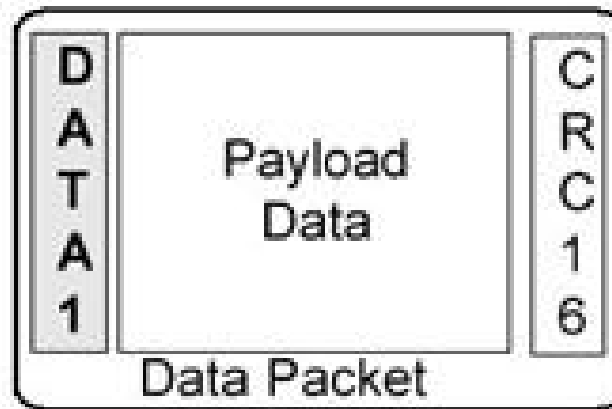
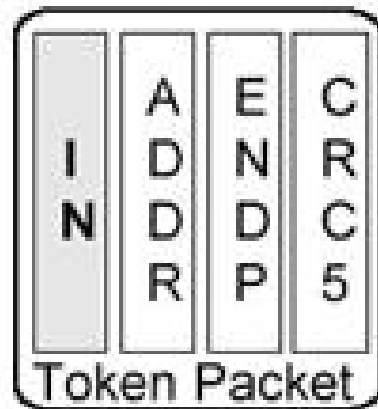


- La spécification 1.1 ajoute l'écriture (OUT)

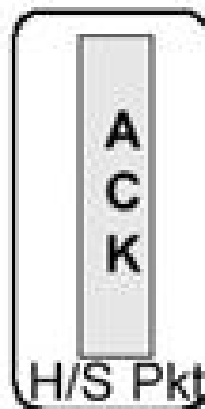
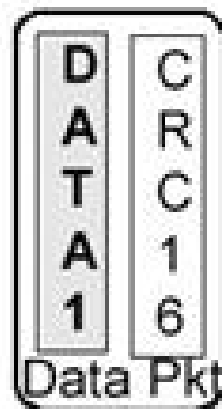
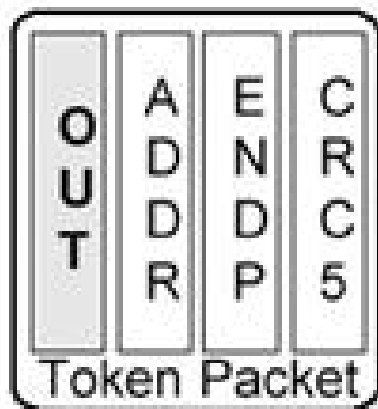
# Messages de type Control



SETUP  
Stage



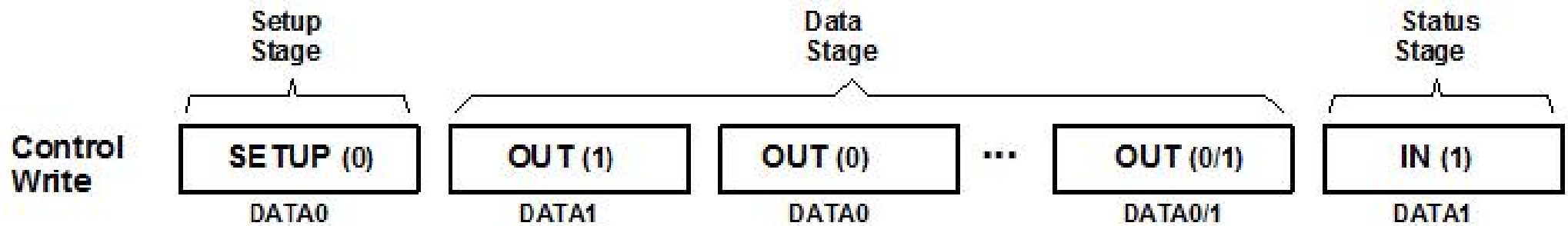
DATA  
Stage  
(optional)



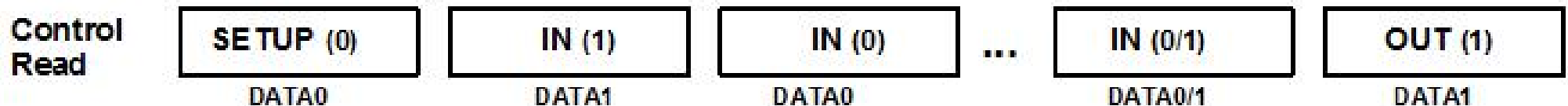
STATUS  
Stage

# Messages de type Control (suite)

- Le flux de type Control impose une structure de messages composée de Stages
- Control Write = écriture de données vers le périphérique



- Control Read = lecture de données produites par le périphérique



# Message Get Descriptor sur le default pipe

Packet #	Sync	SETUP	ADDR	ENDP	CRC5
106	00000001	0xB4	0x00	0x0	0x08

Packet #	Sync	DATA0	DATA			CRC16
107	00000001	0xC3	80 06 00 01 00 00 40 00			0xBB29

Packet #	Sync	ACK
108	00000001	0x4B

Packet #	Sync	IN	ADDR	ENDP	CRC5
110	00000001	0x96	0x00	0x0	0x08

Packet #	Sync	DATA1	DATA			CRC16
111	00000001	0xD2	12 01 00 01 00 00 00 08			0xC8E7

Packet #	Sync	ACK
112	00000001	0x4B

Packet #	Sync	OUT	ADDR	ENDP	CRC5
115	00000001	0x87	0x00	0x0	0x08

Packet #	Sync	DATA1	DATA	CRC16
116	00000001	0xD2		0x0000

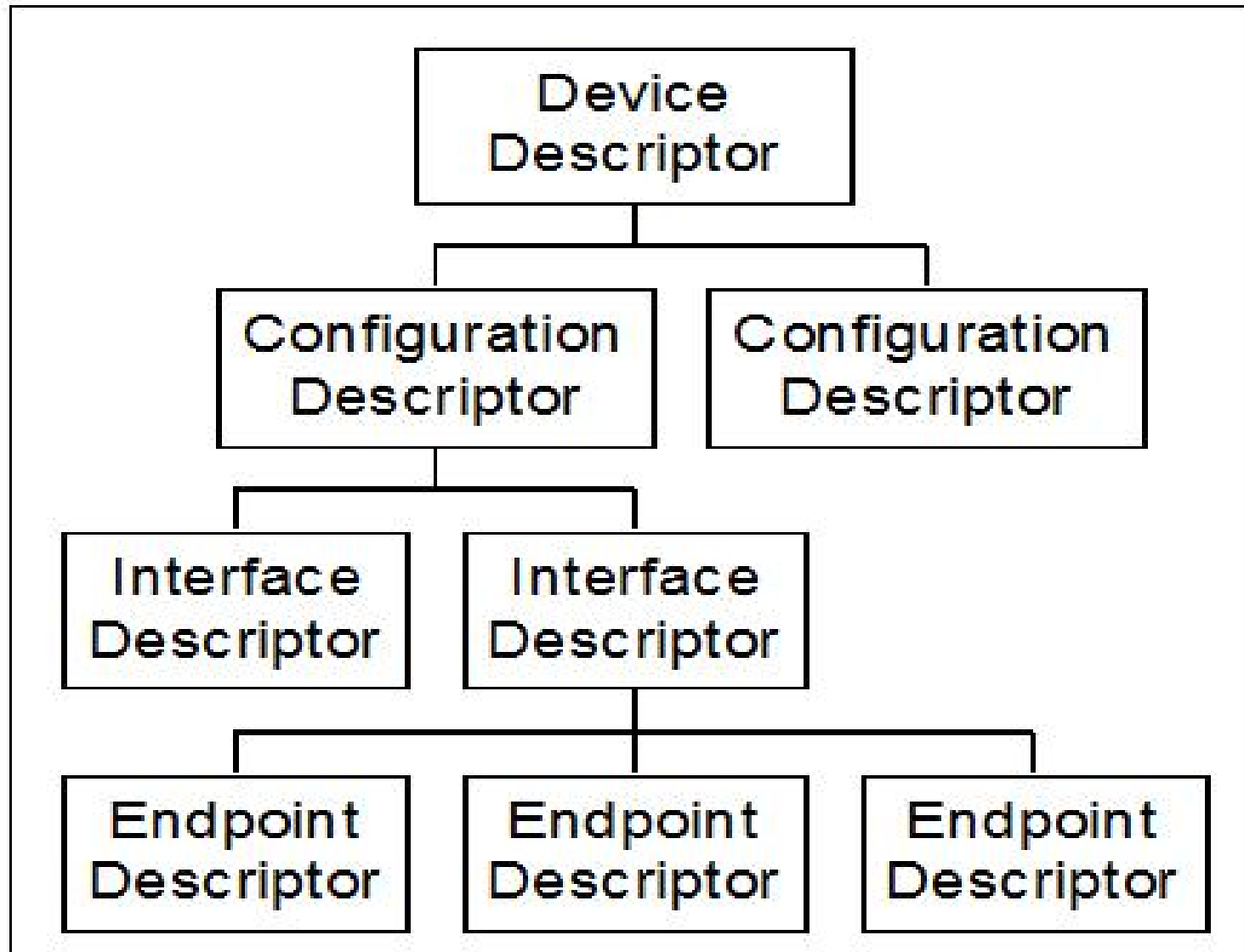
Packet #	Sync	ACK
117	00000001	0x4B

# Enumération (1/2)

- Procédure de détection automatique d'un nouveau périphérique par le Host
- Reset
  - Le Host, après avoir détecté un nouveau périphérique lui génère un Reset
  - Cette opération doit initialiser le périphérique qui ne peut alors que consommer un max de 100 mA via USB
  - Il possède l'adresse 0 et attend
- Get Device Descriptor
  - Le Host identifie le périphérique
  - Le périphérique retourne une table contenant VendorID, ProductID, ClassType, ...

# Descriptor (1/3)

- Chaque périphérique est caractérisé par les paramètres suivants :



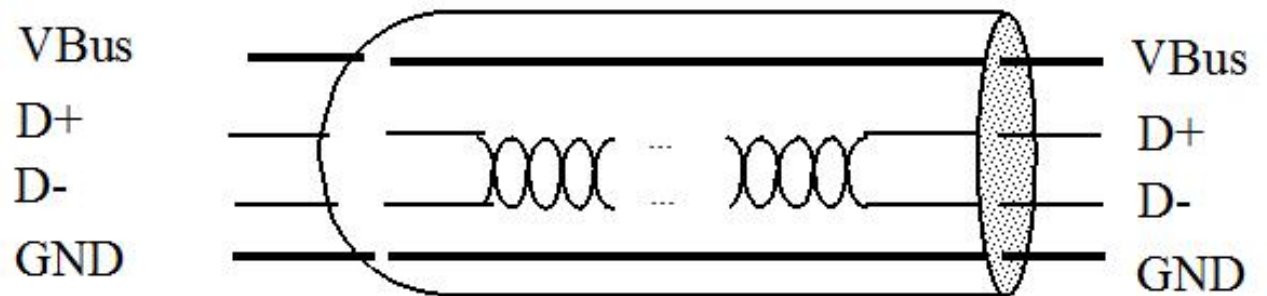
## Descriptor (2/3)

- Device            USB release = 0100            (USB spec. 1.00)  
                      vendor identifier (assigned by USB)  
                      product identifier (assigned by manufacturer)  
                      device release  
                      number of configurations
- Configuration    attributes (bus or self powered) → next slide  
                      max power (from the bus in 2 mA units)  
                      selector  
                      number of interfaces



# Self-powered & Bus-powered

- Le câble comprend 4 fils dont 2 sont réservés à l'alimentation

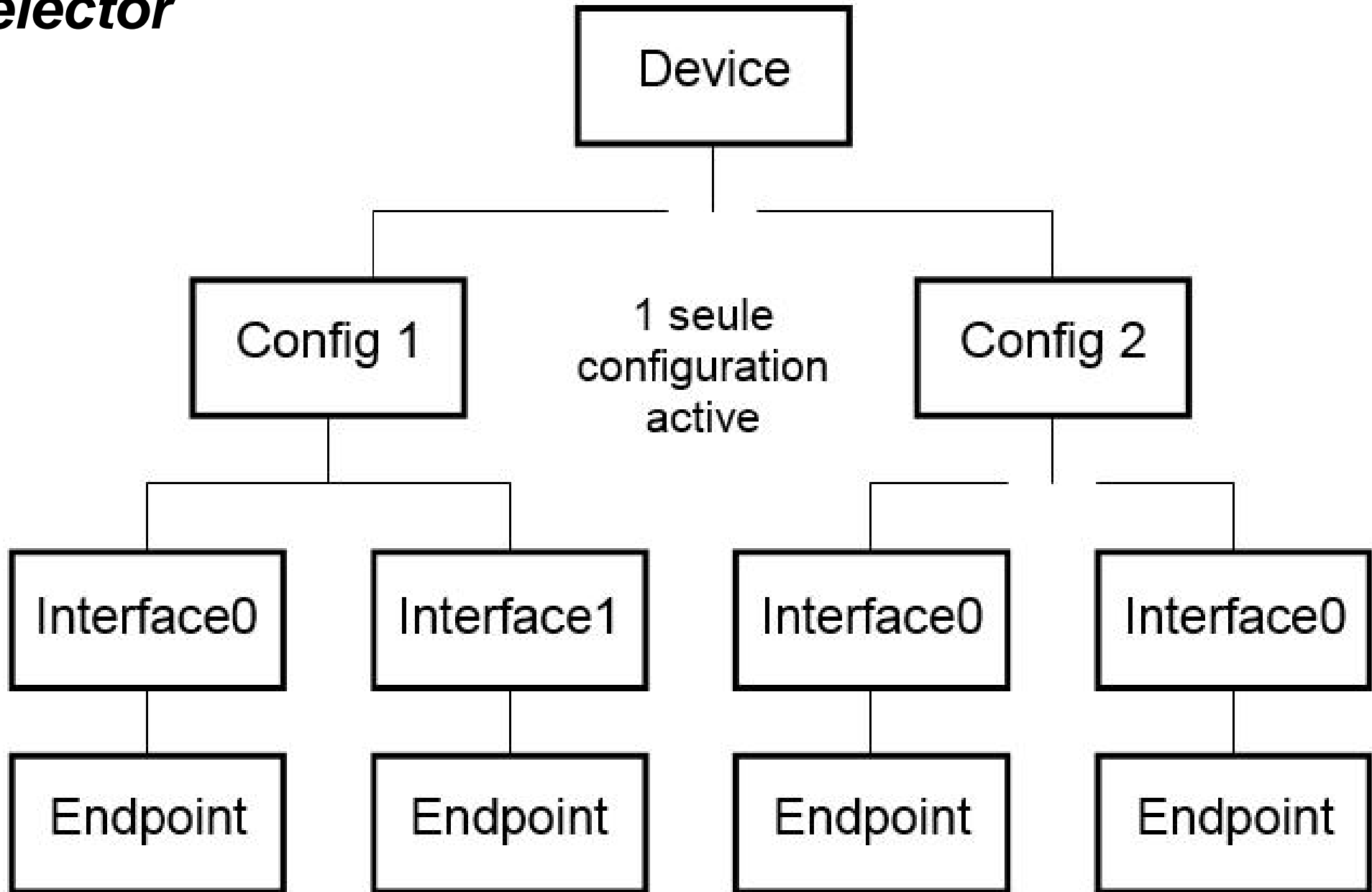


- 2 modes d'alimentation sont possibles pour les périphériques et les hubs
- Bus-powered → alimentation via USB
- Self-powered → propre alimentation
- Un self-powered hub peut fournir 500 mA à chaque périphérique
- Un bus-powered hub peut fournir 100 mA à chaque périphérique

# Descriptor (3/3)

- Interface
  - class code (assigned by USB)
  - selector
  - number of endpoints
- Endpoint
  - address (direction)
  - attribute (bulk, control, interrupt, isochronous)
  - max packet size
  - polling interval in ms (interrupt)

# Selector



2 interfaces  
indépendantes

1 seule  
interface active

## ***Énumération (2/2)***

- **Set Address**

Le Host attribue une adresse (1..127)

- **Get Configuration Descriptor**

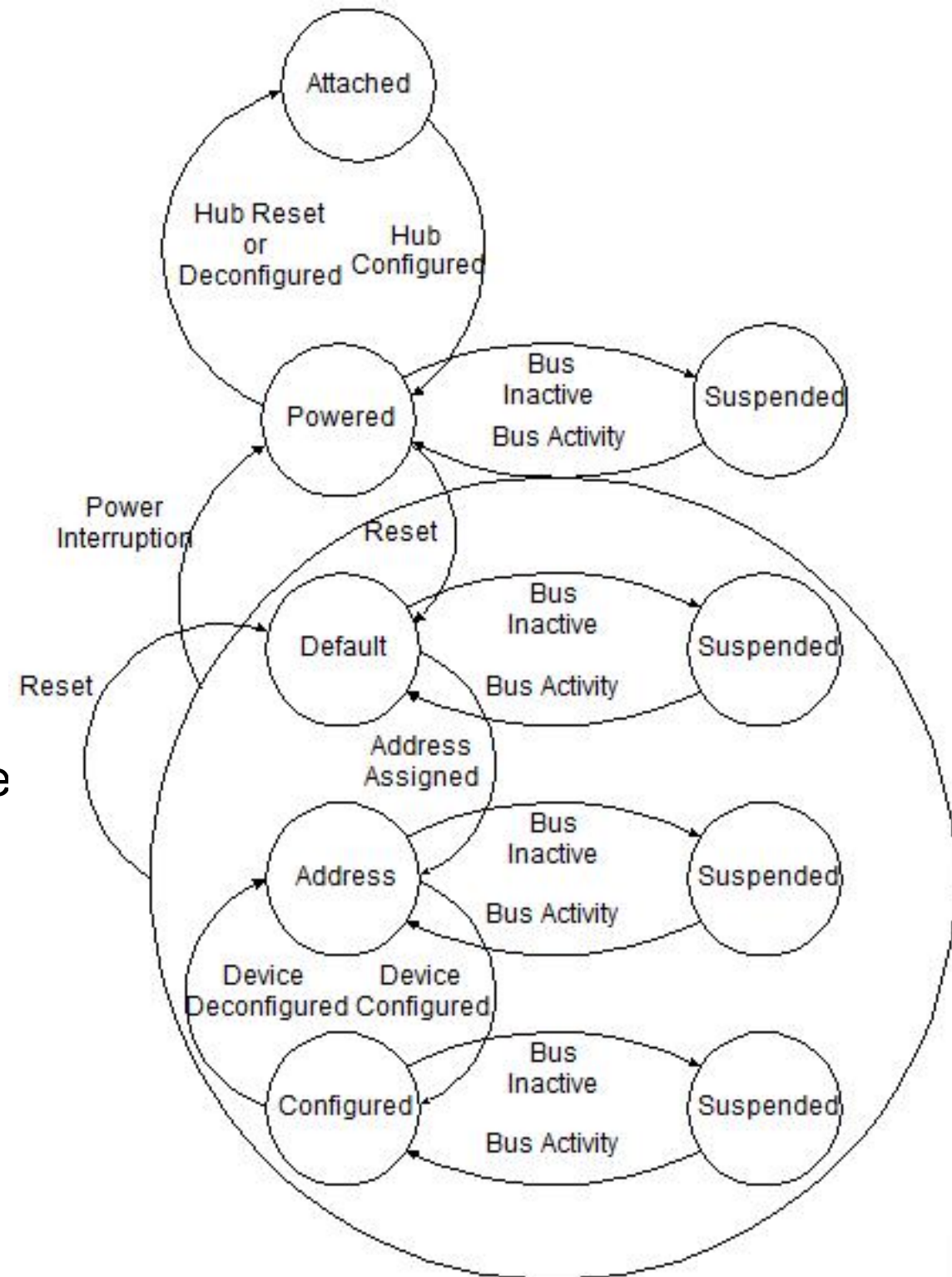
Le Host détermine les caractéristiques du périphérique et charge les pilotes (drivers) spécifiques.

- **Set Configuration**

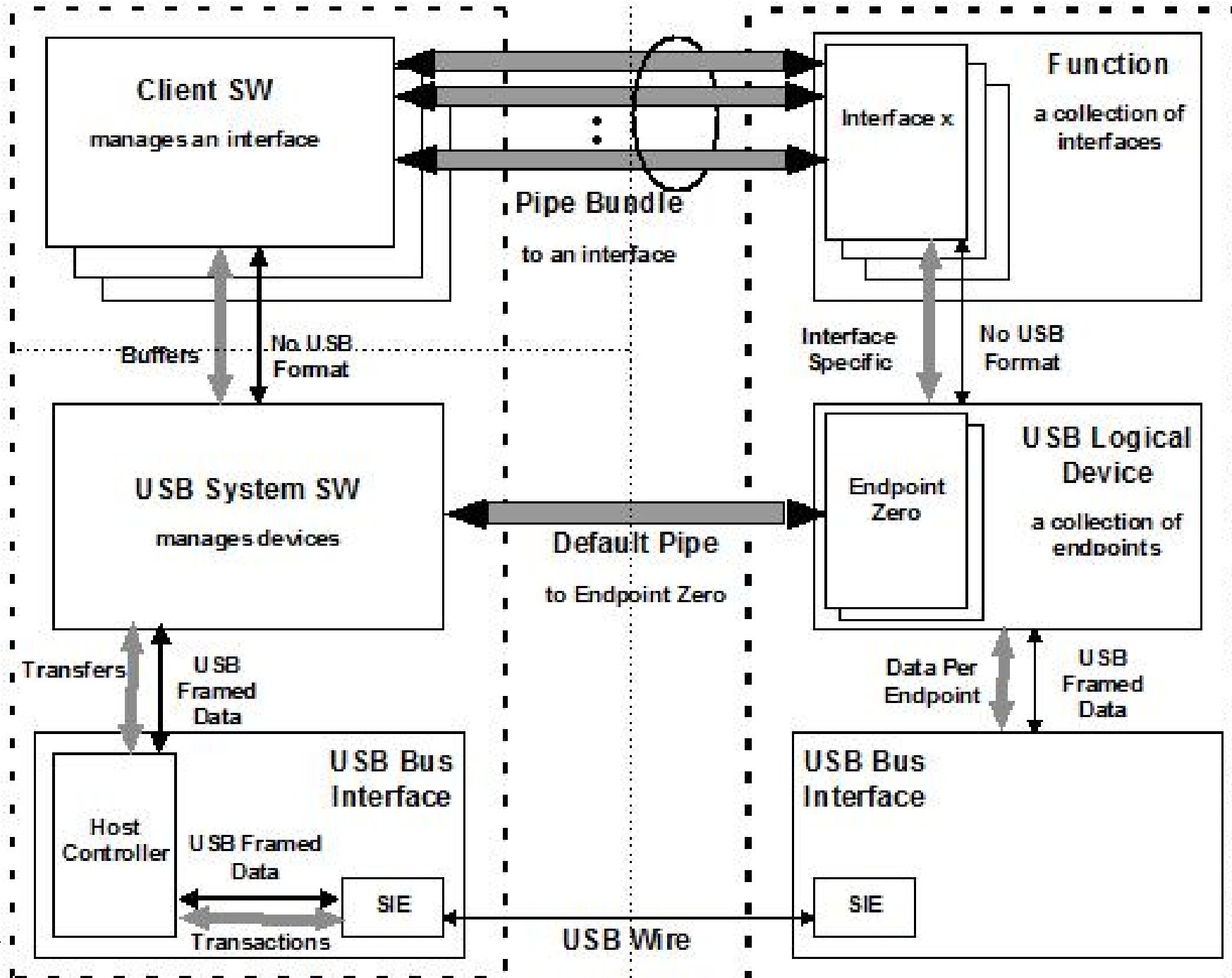
Le périphérique est autorisé à consommer MaxPower et à fonctionner normalement (sous le contrôle du Host)

# Device States

- Le périphérique va passer successivement dans les états :
- Attached  
Périphérique raccordé
- Powered  
Périphérique alimenté
- Default (Reseted)  
Périphérique initialisé (USB)
- Address (Assigned)  
Périphérique possède son adresse
- Configured  
Périphérique configuré par l'application
- Suspended  
Périphérique en mode faible consommation



# Vue d'ensemble



# Labo USB

- Flux isochrone (isoc.ufo)

Item	Device	Endp...	In...	S...	Speed	Comment
Enter text here	E...	E...	E.	E	E...	Enter text here
+ ← IN transaction	3	1			FS	96 bytes (13 D7 D7 D7 61 D8 31 ...
🕒 Start of Frame					FS	1'014
+ ← IN transaction	3	1			FS	96 bytes (77 D6 E3 D6 CD D7 FF ...

- Flux bulk (bulk.ufo)

+ → OUT transaction	2	1			ACK FS	128 bytes (00 00 00 00 01 00 00 ...
+ 🕒 Start of Frame (34)					FS	476 -> 509
+ ← IN transaction	2	1			ACK FS	128 bytes (00 00 00 00 01 00 00 ...
+ 🕒 Start of Frame (192)					FS	510 -> 701
+ → OUT transaction	2	1			ACK FS	128 bytes (00 00 00 00 01 00 00 ...
+ 🕒 Start of Frame (59)					FS	702 -> 760
+ ← IN transaction	2	1			ACK FS	128 bytes (00 00 00 00 01 00 00 ...

# USB 2.0

- Maximum speed of 480 Mbit/s (now called Hi-Speed)
- No physical or electrical difference between USB 1.1 and USB 2.0 connectors
- [http://wiki.answers.com/Q/What\\_is\\_the\\_difference\\_between\\_USB\\_1.1\\_and\\_USB\\_2.0](http://wiki.answers.com/Q/What_is_the_difference_between_USB_1.1_and_USB_2.0)
- Add a new power state between enabled and suspended states
- Any device that is in this power state does not have to reduce its power consumption
- <http://www.differencebetween.net/technology/difference-between-usb-1-1-and-2-0/>
- Backwards compatible with the full speed and low speed transfer rates of USB 1.1
- <http://www.milestek.com/blog/index.php/2009/09/what-is-the-difference-between-usb-11-usb-20-and-usb-30/>



# USB 3.0

- SuperSpeed USB has a 5 Gbps signaling rate offering 10x performance increase over Hi-Speed USB.
- SuperSpeed USB is a Sync-N-Go technology that minimizes user wait-time.
- SuperSpeed USB will provide Optimized Power Efficiency. No device polling and lower active and idle power requirements.
- SuperSpeed USB is backwards compatible with USB 2.0
- <http://www.milestek.com/blog/index.php/2009/09/what-is-the-difference-between-usb-11-usb-20-and-usb-30/>

# Sol 1

Host  $\leftarrow \rightarrow$  Device

- Lecture

Sans erreur  $\rightarrow$  IN  $\leftarrow$  DATA  $\rightarrow$  ACK

Erreur de trm  $\rightarrow$  IN  $\leftarrow$  DATA

Tampon vide  $\rightarrow$  IN  $\leftarrow$  NAK

Défectueux  $\rightarrow$  IN  $\leftarrow$  STALL

- Ecriture

Sans erreur  $\rightarrow$  OUT  $\rightarrow$  DATA  $\leftarrow$  ACK

Erreur de trm  $\rightarrow$  OUT  $\rightarrow$  DATA

Tampon plein  $\rightarrow$  OUT  $\rightarrow$  DATA  $\leftarrow$  NAK

Défectueux  $\rightarrow$  OUT  $\rightarrow$  DATA  $\leftarrow$  STALL

## Sol 2

- Selon slide 19  
Périphérique de type Full Speed = 12 Mbit/s  
Données utiles = 1023 byte
- Selon slide 8  
1 jeton chaque ms
- Débit utile =  $1023 / 10^{-3} = 1'023'000 \text{ byte/s} = \sim 8.2 \text{ Mbit/s}$
- Overhead selon slides 17