HACKING THE USB WORLD WITH FACEDANCER

A WHIRLWIND TOUR OF USB

KATE TEMKIN & DOMINIC SPILL

THE UNIVERSAL SERIAL BUS

Resources:

- http://www.beyondlogic.org/usbnutshell/
- USB Complete, 5th Edition (ISBN: 1931448280)
- USB specs: http://www.usb.org/developers/docs/usb20_docs/

Ultra-high-level:

- current specification: 3.1; 2.0 is commonly still used
  - most of the ‘security surface’ is still on USB 2; we cover it exclusively
- four current speeds: *low*, *high*, *full*, and *super*
- multiple types of connectors in multiple sizes: {mini, micro} A, B, AB, C

USB SPECS: SOURCE OF ALL KNOWLEDGE

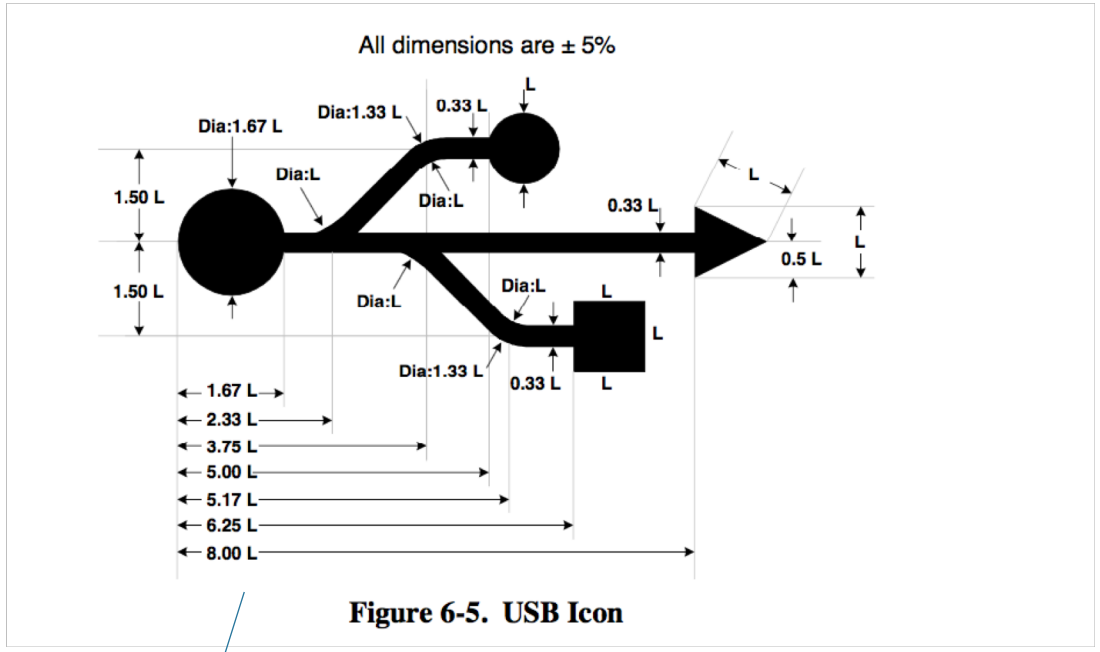
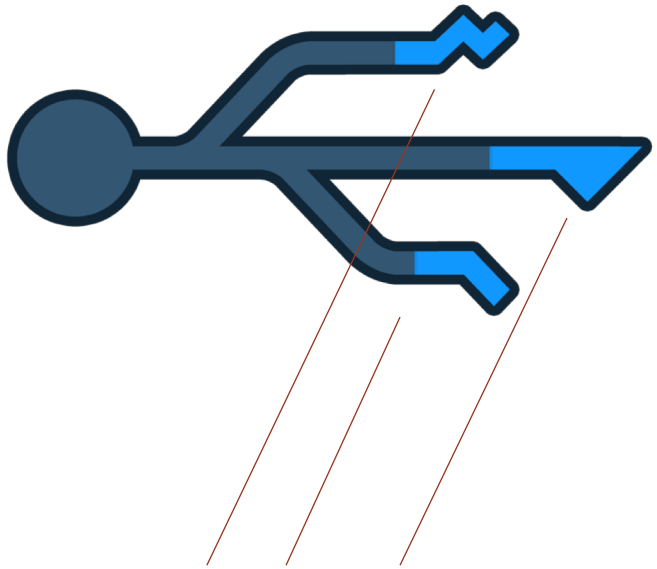


Figure 6-5. USB Icon

full of important information about USB
and, uh, also diagrams like this one



spec **violation**

nothing like some spec violation to start off a Monday

left: from USB spec
right: from this course

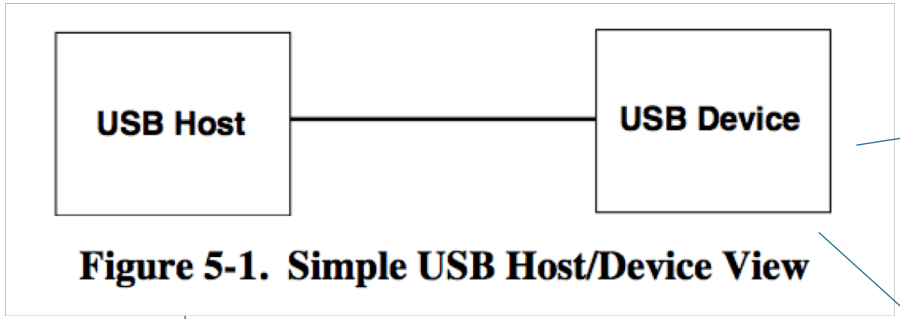


Figure 5-1. Simple USB Host/Device View

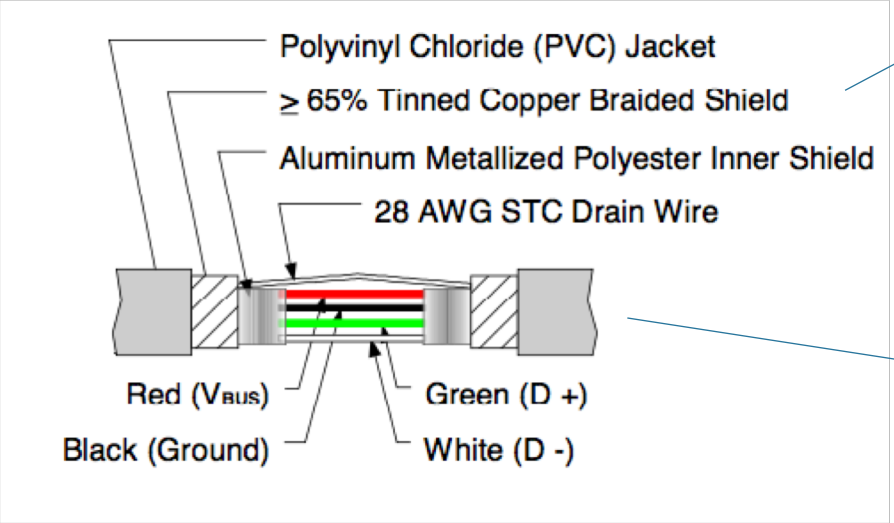
actual diagram from the USB specification
[probably not simplified enough]

the device faces *upstream*, and responds (only) when the host makes requests of it

the host faces *downstream*, provides power, and does most of the heavy lifting

PHYSICAL LAYER // USB 2.0

All diagrams from the USB 2.0 specification.



all communications occur over a single differential pair— essentially a *single signal* over two wires

a limited amount of power (5V @ 500mA) is also provided from *host* to *device*

PHYSICAL LAYER // USB 2.0

All diagrams from the USB 2.0 specification.

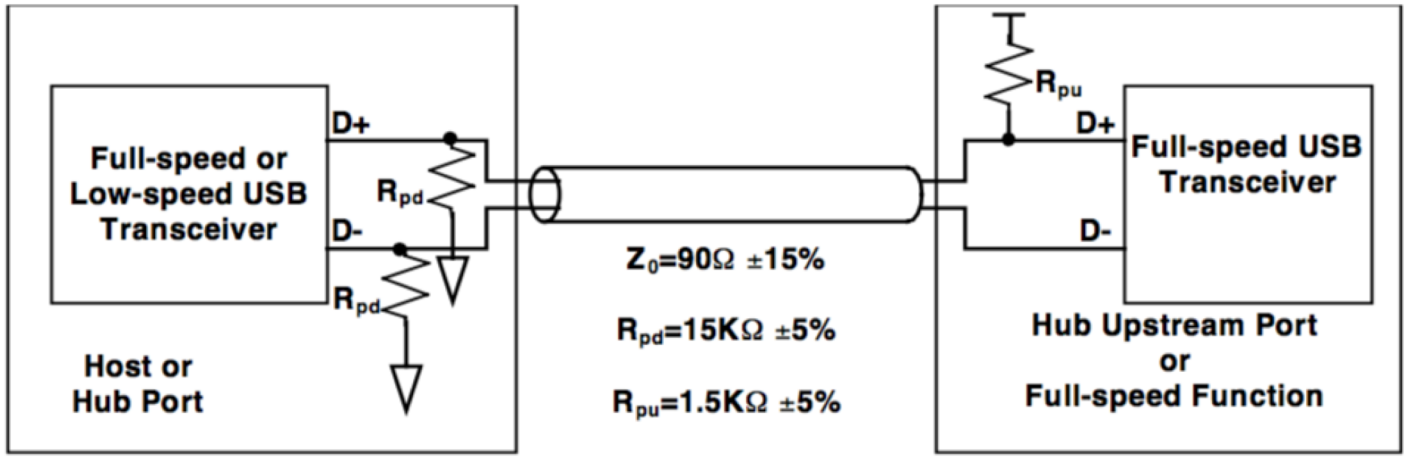


Figure 7-20. Full-speed Device Cable and Resistor Connections

a pull-up resistor indicates the presence of a *USB device*

the location of the resistor indicates if the device is *low* speed or *full/high*

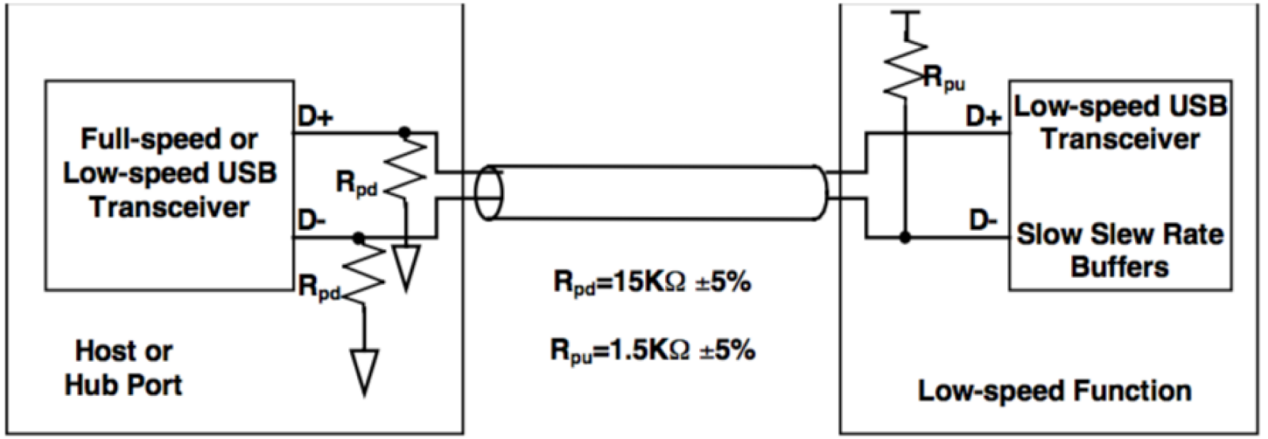
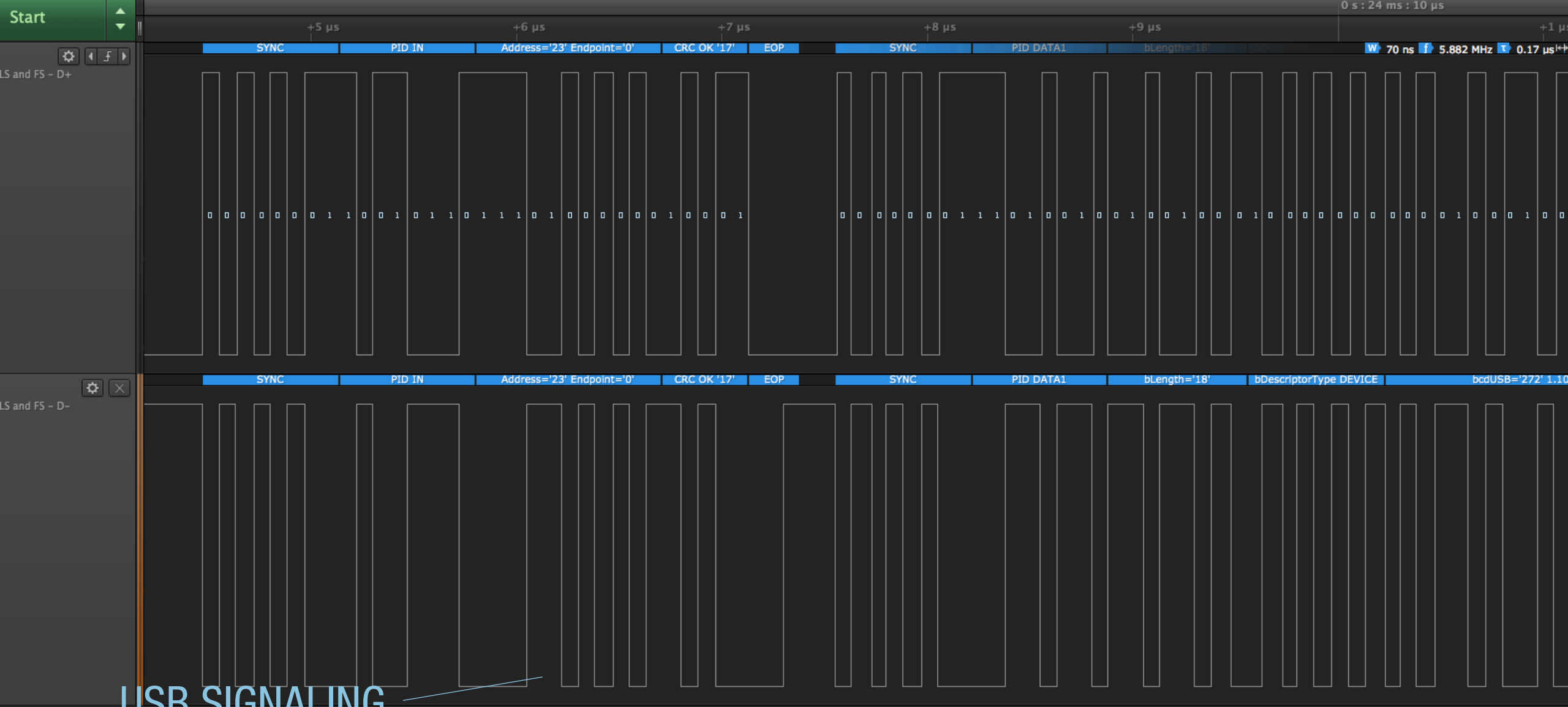
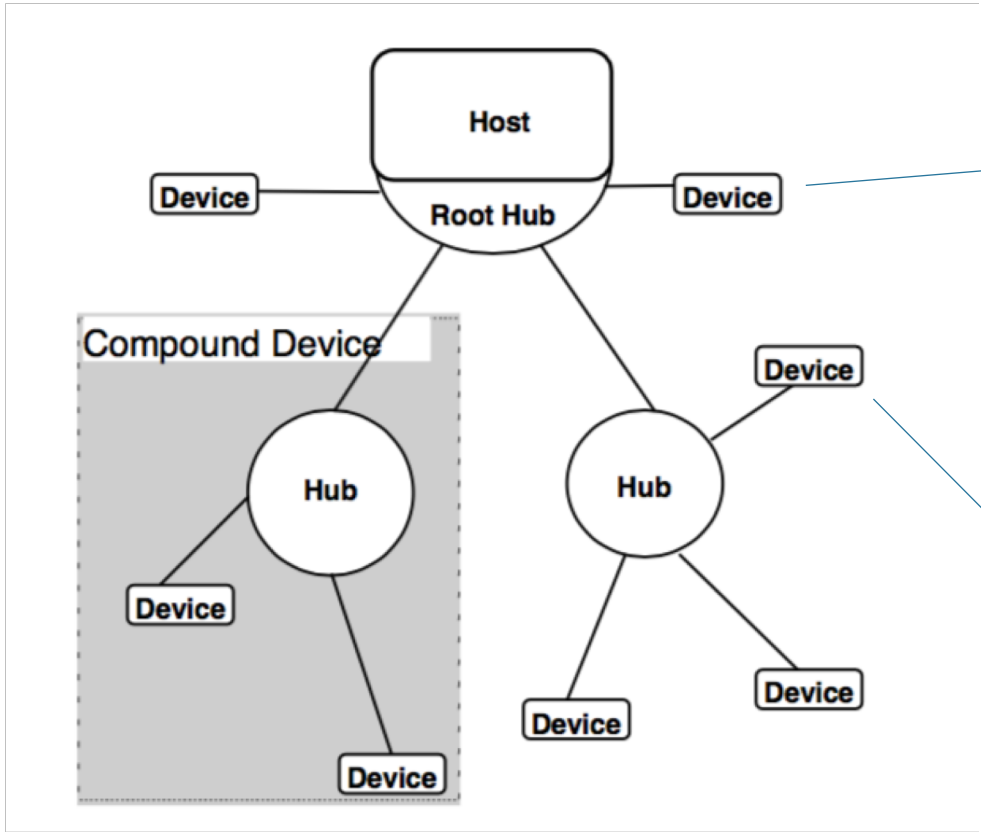


Figure 7-21. Low-speed Device Cable and Resistor Connections



USB SIGNALING

- Single differential pair: signals (almost) always inverse of each other
- Half-duplex by necessity– only one participant controls the lines

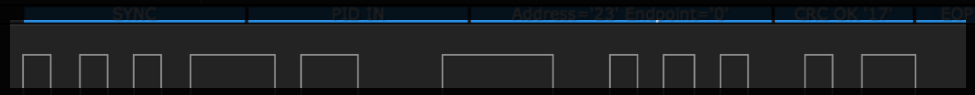


usb is designed to allow communications with more than one device simultaneously

devices, in turn can communicate more than one data stream at once, emulating multiple logical connections

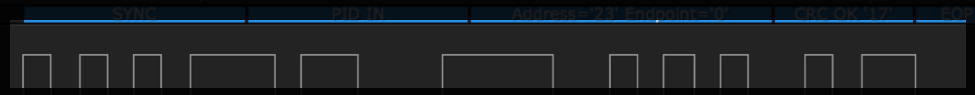
PHYSICAL LAYER // USB 2.0

All diagrams from the USB 2.0 specification.



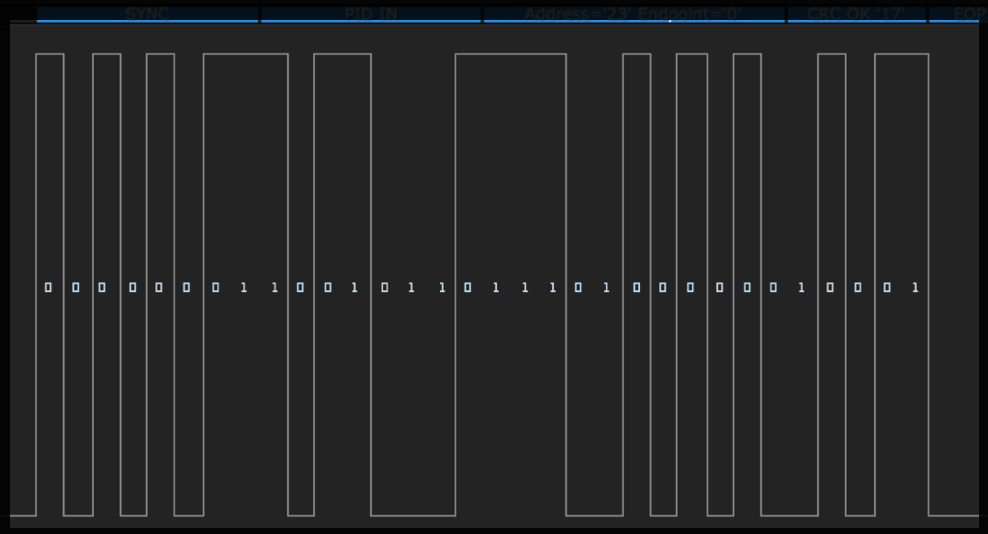
SHARING THE LINES

- Each communication is directed at a particular *device* and *endpoint*
- The *host* decides who gets to talk, and when
  - Devices only “speak when spoken to”



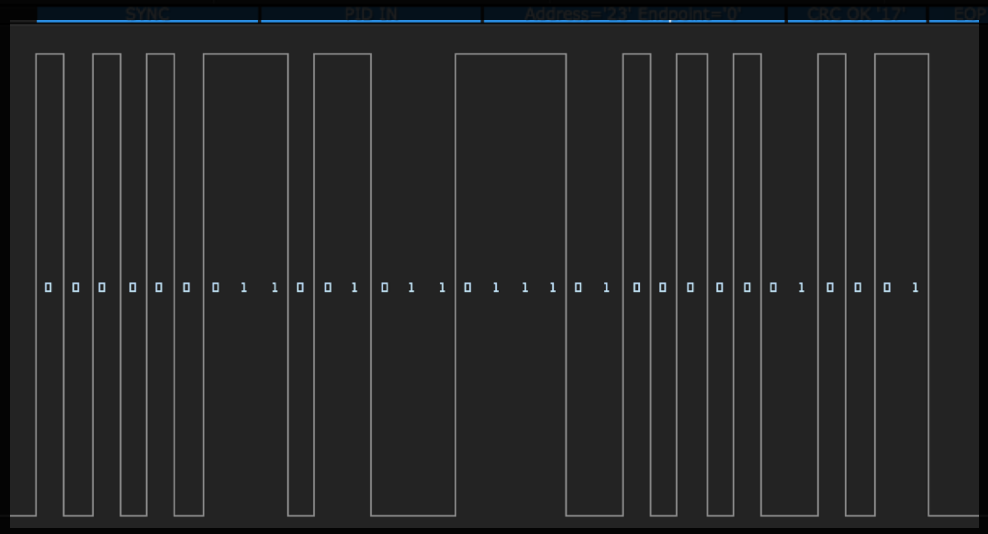
SHARING THE LINES

- Each device is assigned an *address*, and listens for packets directed to it
- Each device can accept multiple *endpoints*, which allow communications to be functionally grouped



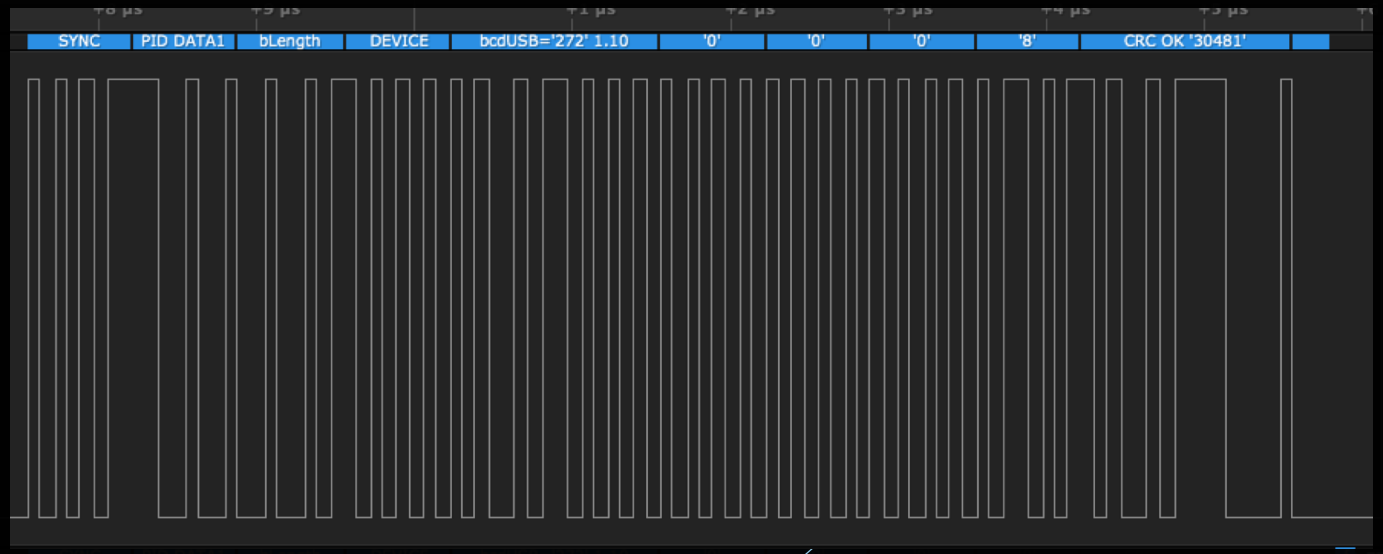
USB PACKETS

- The ‘atoms’ that make up USB communications
- Always contain a *packet ID (PID)* that specifies the packet type



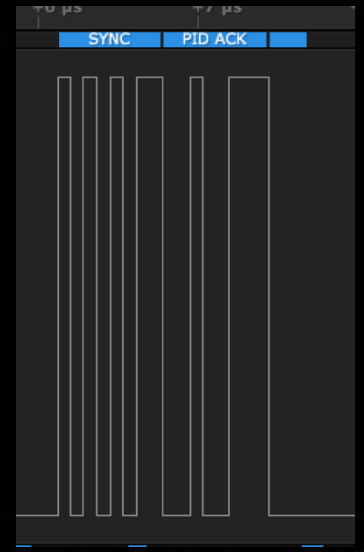
TOKEN PACKETS

- Start each USB communication.
- Specify the *direction* of the communication, the *device* that should be listening, and the *endpoint* the device should route the packet to.
- *OUT* and *SETUP* tokens indicate that data is about to be sent *to the device*.
- IN tokens temporarily ‘give’ the bus to a given device, allowing it to speak.



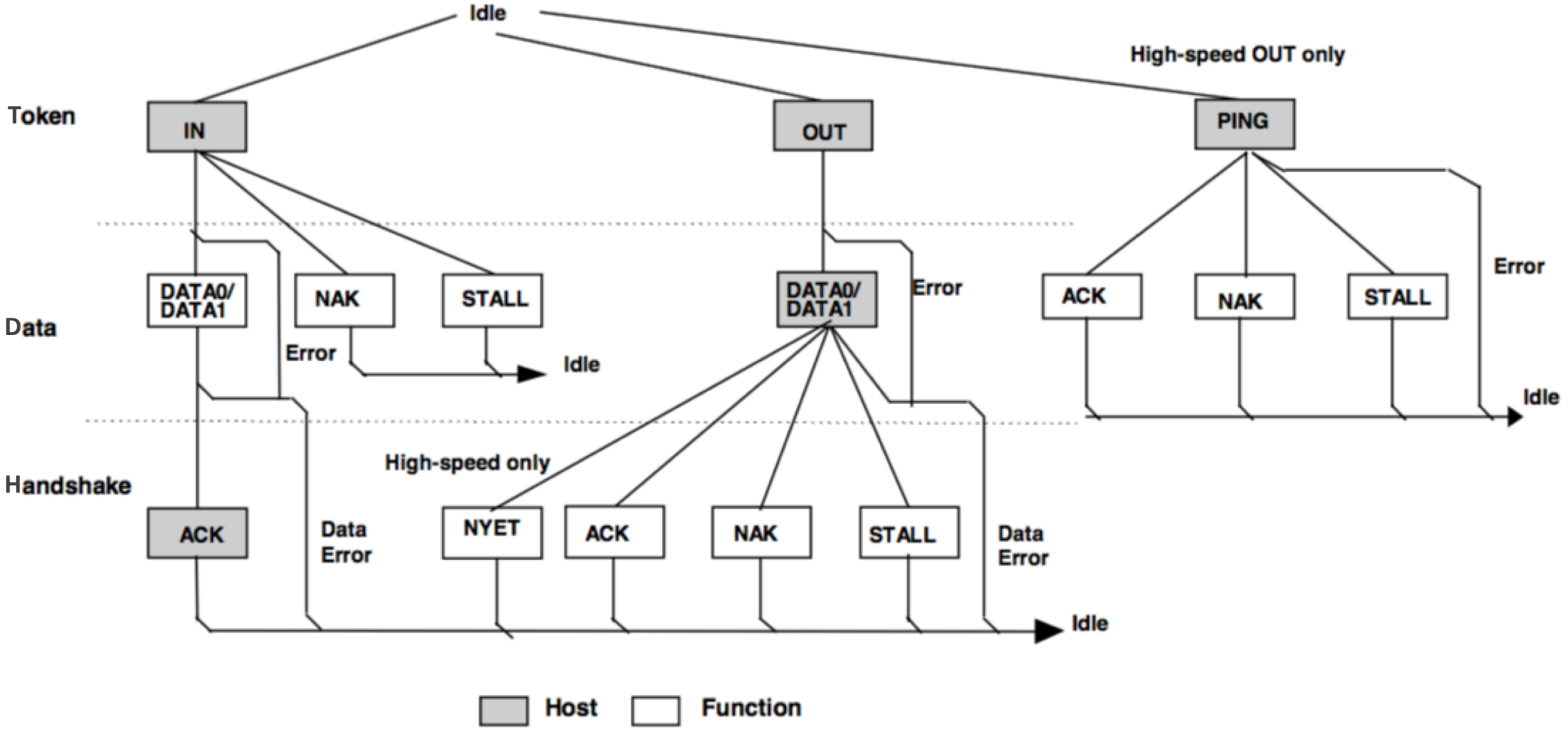
DATA PACKETS

- Follow an IN/OUT/SETUP token.
  - Can originate from the device, following an IN token.
- Carry the actual USB data.
- Tokens typically alternate between DATA0 and DATA1, providing some protection against missed packets.
- Contain simple CRC16 error checking.



HANDSHAKE PACKETS

- Used to indicate the status of a communication.
- Three main types, indicated by PID:
  - ACK- “communication successful”
  - NAK- “not successful; retry”
  - STALL- “not supported; don’t retry”



USB TRANSACTIONS

All diagrams from the USB 2.0 specification.

each transaction starts with a *token packet* that contains direction and destination information

if data is ready, a *data packet* is sent that contains the data; otherwise, a NAK or STALL handshake packet is sent

if data was transmitted, the other side responds with an ACK or NAK packet

USB TRANSACTIONS

All diagrams from the USB 2.0 specification.

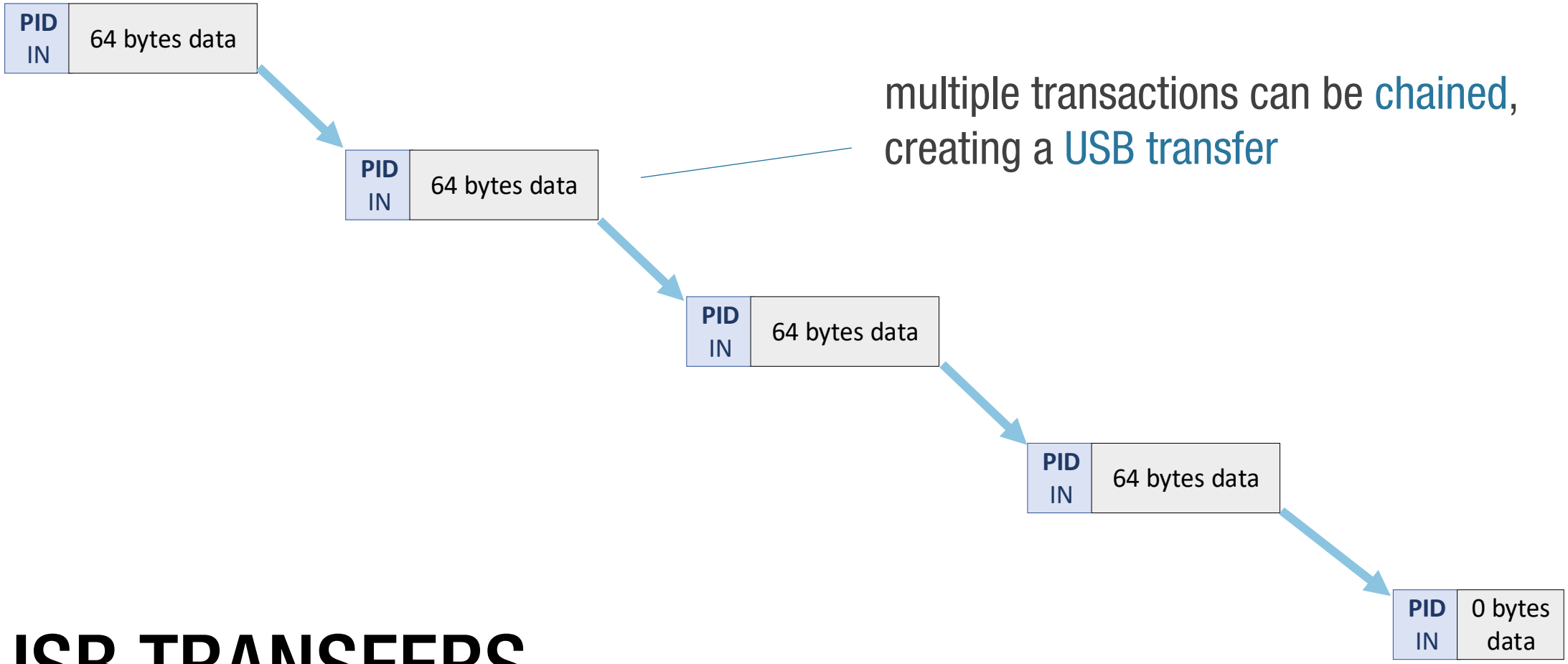
Field	Value
Length	256
Address	0x1000

Field	Value
Length	192
Address	0x1040

Field	Value
Length	128
Address	0x1080

Field	Value
Length	64
Address	0x10C0

Field	Value
Length	0
Address	0x1100



USB TRANSFERS

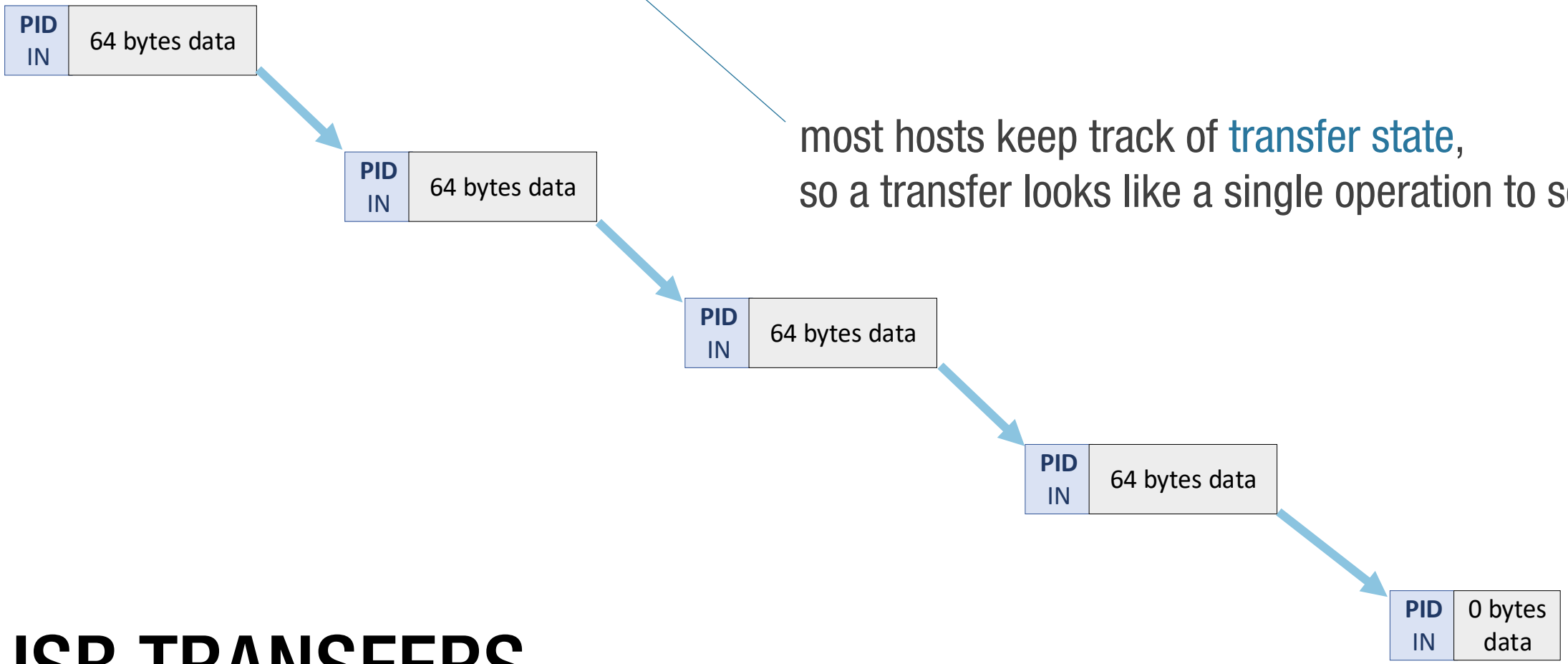
Field	Value
Length	256
Address	0x1000

Field	Value
Length	192
Address	0x1040

Field	Value
Length	128
Address	0x1080

Field	Value
Length	64
Address	0x10C0

Field	Value
Length	0
Address	0x1100



most hosts keep track of transfer state,
so a transfer looks like a single operation to software

USB TRANSFERS

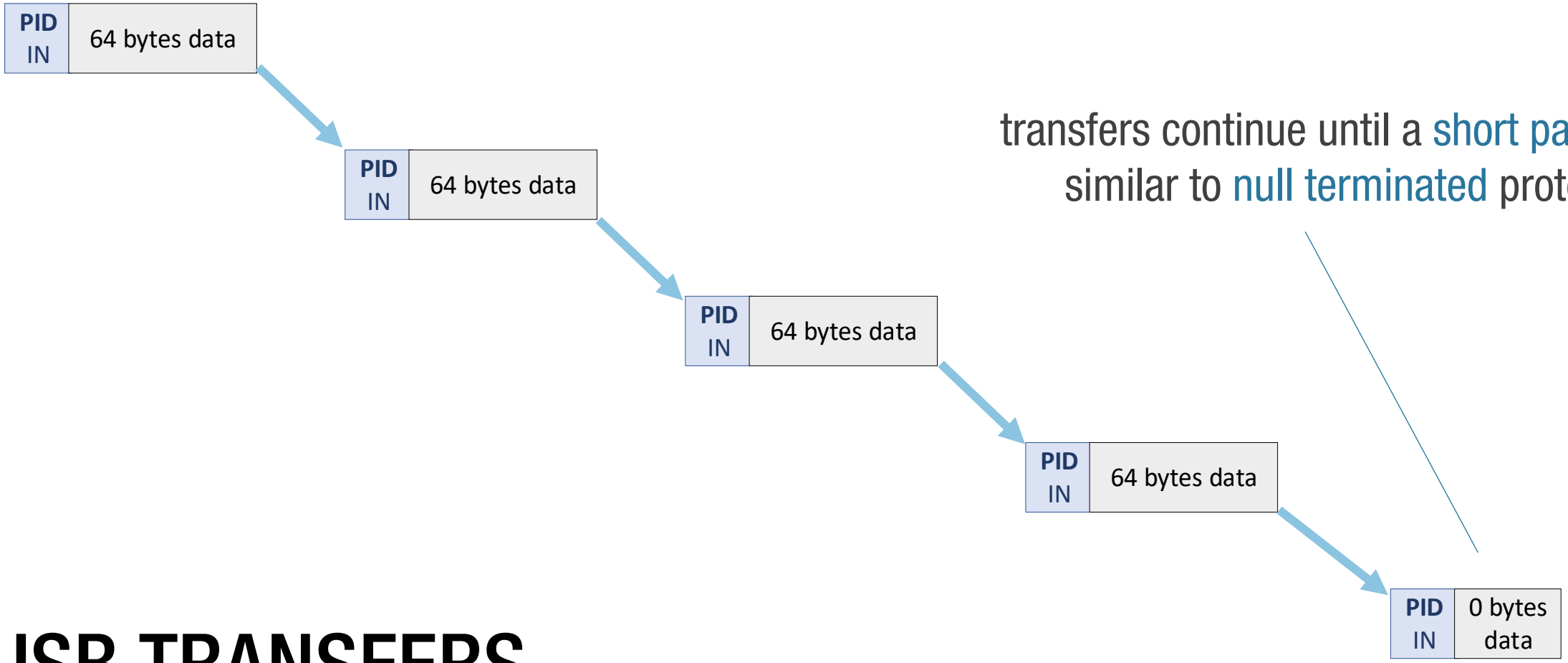
Field	Value
Length	256
Address	0x1000

Field	Value
Length	192
Address	0x1040

Field	Value
Length	128
Address	0x1080

Field	Value
Length	64
Address	0x10C0

Field	Value
Length	0
Address	0x1100



transfers continue until a short packet;
similar to null terminated protocols

USB TRANSFERS