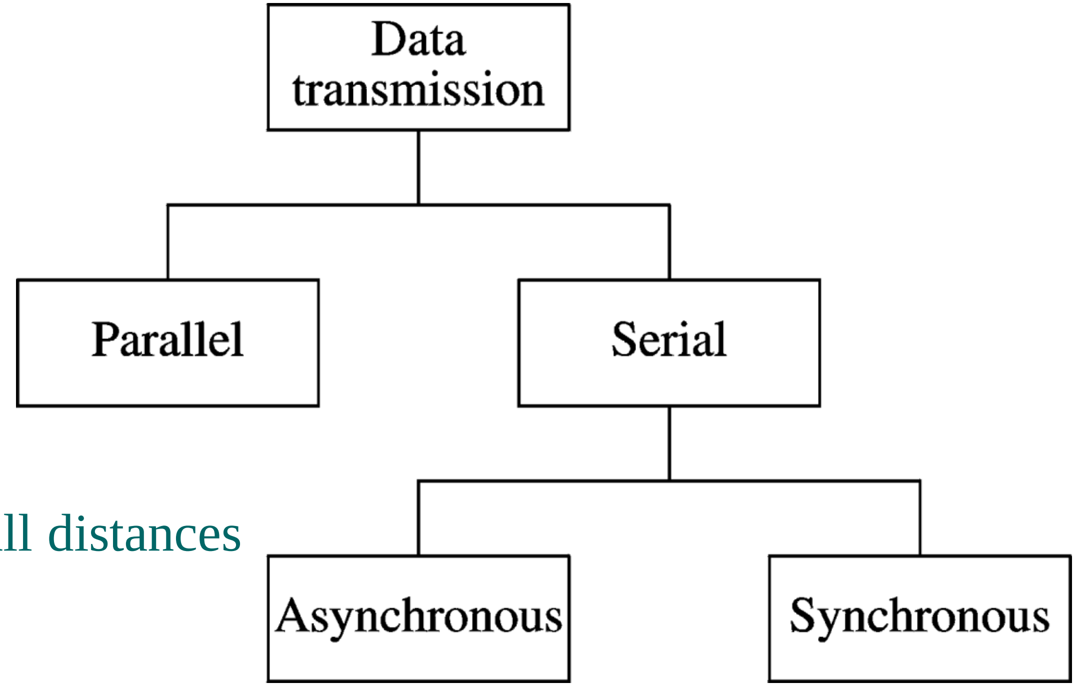
External Bus Interfacing Signals

Dr.N.Umapathi
Associate Professor/ECE
JITs – Karimnagar

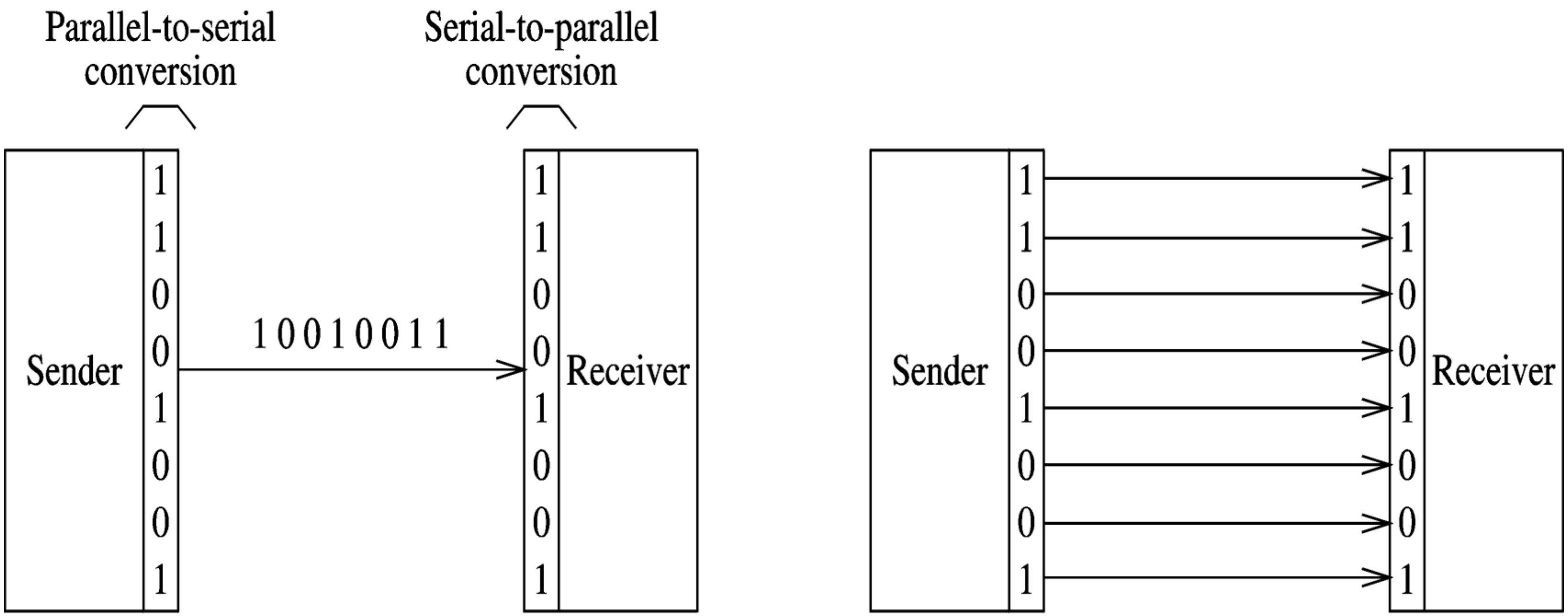
External Interface

- Two ways of interfacing I/O devices
  - ∗ Serial
    - » Cheaper
    - » Slower
  - ∗ Parallel
    - » Faster
    - » Data skew
    - » Limited to small distances



External Interface (cont'd)

Two basic modes of data transmission

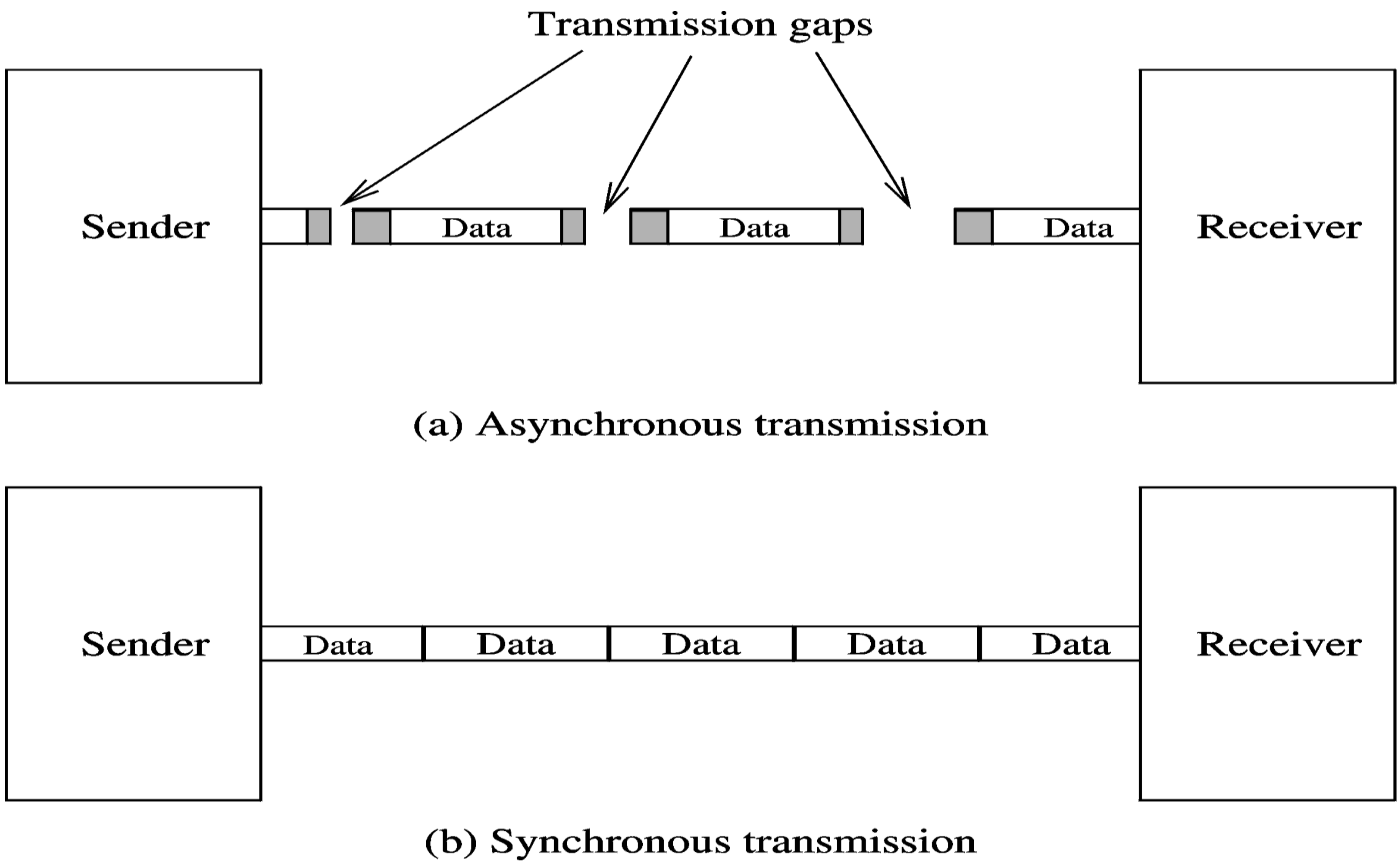


(a) Serial transmission
(b) Parallel transmission

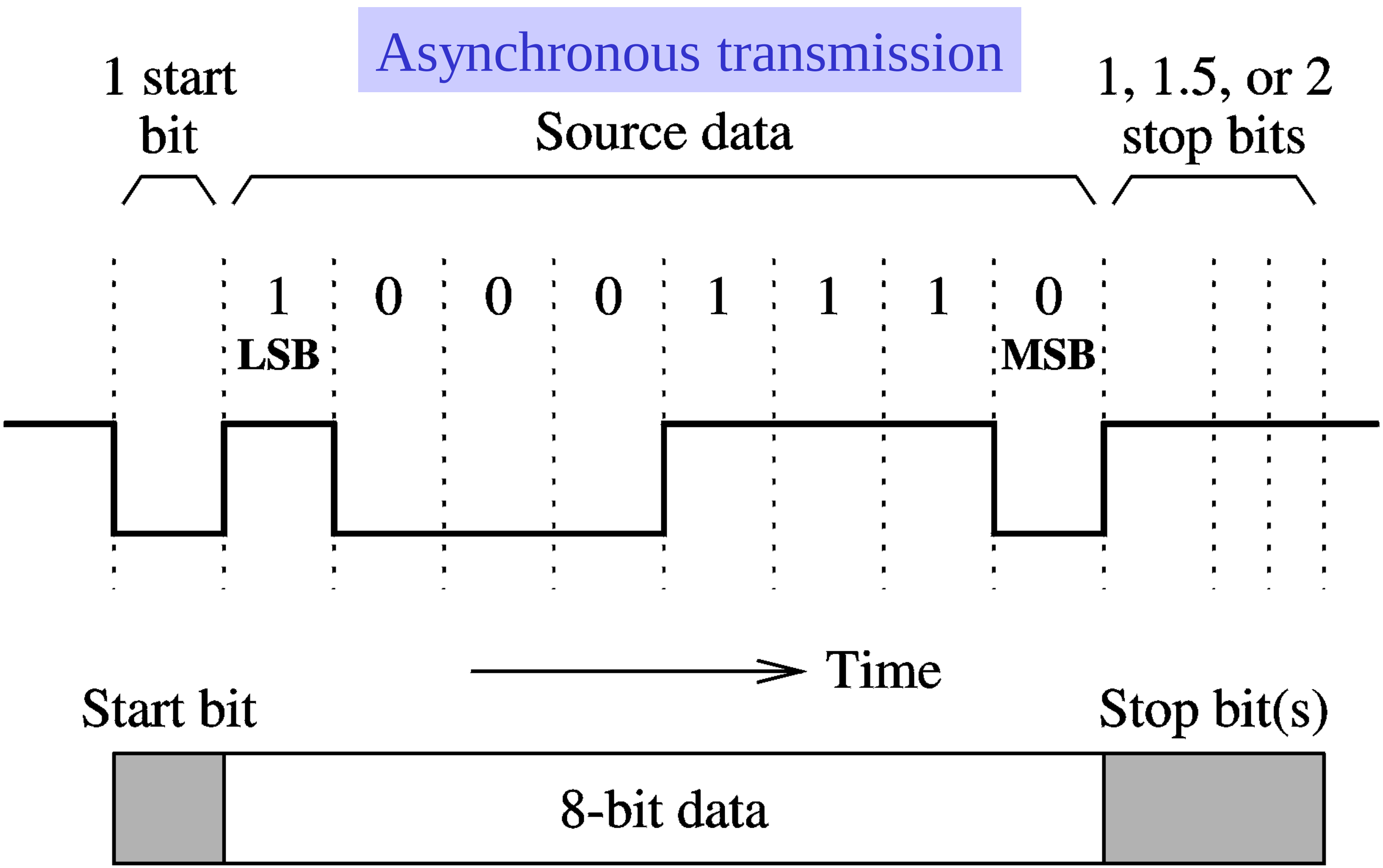
External Interface (cont'd)

- Serial transmission
  - ∗ Asynchronous
    - » Each byte is encoded for transmission
      - – Start and stop bits
    - » No need for sender and receiver synchronization
  - ∗ Synchronous
    - » Sender and receiver must synchronize
      - – Done in hardware using phase locked loops (PLLs)
    - » Block of data can be sent
    - » More efficient
      - – Less overhead than asynchronous transmission
    - » Expensive

External Interface (cont'd)



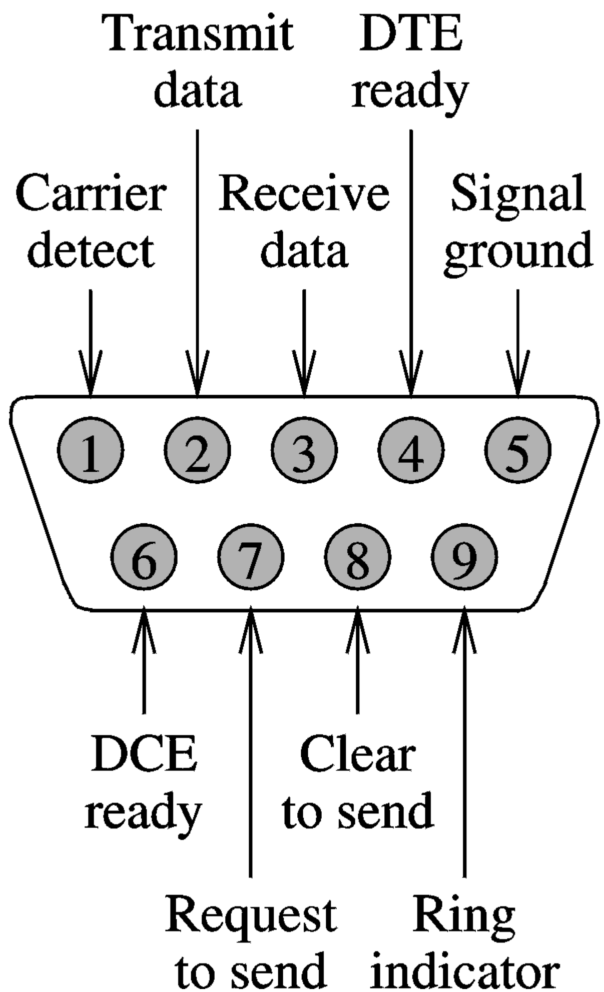
External Interface (cont'd)



To be used with S. Dandamudi, "Fundamentals of Computer Organization and Design," Springer, 2003.

External Interface (cont'd)

- EIA-232 serial interface
  - ∗ Low-speed serial transmission
  - ∗ Adopted by Electronics Industry Association (EIA)
    - » Popularly known by its predecessor RS-232
  - ∗ It uses a 9-pin connector DB-9
    - » Uses 8 signals
  - ∗ Typically used to connect a modem to a computer



External Interface (cont'd)

- Transmission protocol uses three phases
  - ∗ Connection setup
    - » Computer A asserts DTE Ready
      - – Transmits phone# via Transmit Data line (pin 2)
    - » Modem B alerts its computer via Ring Indicator (pin 9)
      - – Computer B asserts DTE Ready (pin 4)
      - – Modem B generates carrier and turns its DCE Ready
    - » Modem A detects the carrier signal from modem B
      - – Modem A alters its computer via Carrier Detect (pin 1)
      - – Turns its DCE Ready
  - ∗ Data transmission
    - » Done by handshaking using
      - – request-to-send (RTS) and clear-to-send (CTS) signals
  - ∗ Connection termination
    - » Done by deactivating RTS

External Interface (cont'd)

- Parallel printer interface
  - * A simple parallel interface
  - * Uses 25-pin DB-25
    - » 8 data signals
      - – Latched by strobe (pin 1)
    - » Data transfer uses simple handshaking
      - – Uses acknowledge (CK) signal
        - ✈After each byte, computer waits for ACK
    - » 5 lines for printer status
      - – Busy, out-of-paper, online/offline, autofeed, and fault
    - » Can be initialized with INIT
      - – Clears the printer buffer and resets the printer

External Interface (cont'd)

Table 19.3 Parallel printer interface signals

Pin #	Signal	Signal direction	Signal function
1	STROBE	PC $\Longrightarrow$ printer	Clock used to latch data
2	Data 0	PC $\Longrightarrow$ printer	Data bit 0 (LSB)
3	Data 1	PC $\Longrightarrow$ printer	Data bit 1
4	Data 2	PC $\Longrightarrow$ printer	Data bit 2
5	Data 3	PC $\Longrightarrow$ printer	Data bit 3
6	Data 4	PC $\Longrightarrow$ printer	Data bit 4
7	Data 5	PC $\Longrightarrow$ printer	Data bit 5
8	Data 6	PC $\Longrightarrow$ printer	Data bit 6
9	Data 7	PC $\Longrightarrow$ printer	Data bit 7 (MSB)
10	ACK	printer $\Longrightarrow$ PC	Printer acknowledges receipt of data
11	BUSY	printer $\Longrightarrow$ PC	Printer is busy
12	POUT	printer $\Longrightarrow$ PC	Printer is out of paper
13	SEL	printer $\Longrightarrow$ PC	Printer is online
14	AUTO FEED	printer $\Longrightarrow$ PC	Autofeed is on
15	FAULT	printer $\Longrightarrow$ PC	Printer fault
16	INIT	PC $\Longrightarrow$ printer	Clears printer buffer and resets printer
17	SLCT IN	PC $\Longrightarrow$ printer	TTL high level
18–25	Ground	N/A	Ground reference

External Interface (cont'd)

- SCSI
  * Pronounced "scuzzy"
  * Small Computer System Interface
    » Supports both internal and external connection
  * Comes in two bus widths
    » 8 bits
      – Known as ***narrow SCSI***
      – Uses a 50-pin connector
      – Device id can range from 0 to 7
    » 16 bits
      – Known as ***wide SCSI***
      – Uses a 68-pin connector
      – Device id can range from 0 to 15

External Interface (cont'd)

Table 19.4 Types of SCSI

SCSI type	Bus width (bits)	Transfer rate MB/s
SCSI 1	8	5
Fast SCSI	8	10
Ultra SCSI	8	20
Ultra 2 SCSI	8	40
Wide Ultra SCSI	16	40
Wide Ultra 2 SCSI	16	80
Ultra 3 (Ultra 160) SCSI	16	160
Ultra 4 (Ultra 320) SCSI	16	320

External Interface (cont’d)

Table 19.5 Narrow SCSI signals

Description	Signal	Pin	Pin	Signal	Description
Twisted pair ground	GND	1	26	D0	Data 0
Twisted pair ground	GND	2	27	D1	Data 1
Twisted pair ground	GND	3	28	D2	Data 2
Twisted pair ground	GND	4	29	D3	Data 3
Twisted pair ground	GND	5	30	D4	Data 4
Twisted pair ground	GND	6	31	D5	Data 5
Twisted pair ground	GND	7	32	D6	Data 6
Twisted pair ground	GND	8	33	D7	Data 7
Twisted pair ground	GND	9	34	DP	Data parity bit
Ground	GND	10	35	GND	Ground
Ground	GND	11	36	GND	Ground
Reserved		12	37		Reserved
No connection		13	38	TermPwr	Termination power (+5 V)

cont’d

External Interface (cont'd)

Reserved		14	39		Reserved
Ground	GND	15	40	GND	Ground
Twisted pair ground	GND	16	41	ATN	Attention
Ground	GND	17	42	GND	Ground
Twisted pair ground	GND	18	43	BSY	Busy
Twisted pair ground	GND	19	44	ACK	Acknowledge
Twisted pair ground	GND	20	45	RST	Reset
Twisted pair ground	GND	21	46	MSG	Message
Twisted pair ground	GND	22	47	SEL	Selection
Twisted pair ground	GND	23	48	C/D	Command/data
Twisted pair ground	GND	24	49	REQ	Request
Twisted pair ground	GND	25	50	I/O	Input/output

External Interface (cont'd)

- SCSI uses client-server model
  - * Uses terms ***initiator*** and ***target*** for client and server
    - » Initiator issues commands to targets to perform a task
      - – Initiators are typically SCSI host adaptors
    - » Targets receive the command and perform the task
      - – Targets are SCSI devices like disk drives
- SCSI transfer proceeds in phases
  - * Command
  - * Message in
  - * Message out
  - * Data in
  - * Data out
  - * Status

IN and OUT from the initiator point of view

External Interface (cont'd)

* SCSI uses asynchronous mode for all bus negotiations
  - » Uses handshaking using REQ and ACK signals for each byte of data
* On a synchronous SCSI
  - » Data are transferred synchronously
  - » REQ-ACK signals are not used for each byte
  - » A number of bytes (e.g., 8) can be sent without waiting for ACK
    - – Improves throughput
    - – Minimizes adverse impact of cable propagation delay

USB

- Universal Serial Bus
  * Originally developed in 1995 by a consortium including
    - » Compaq, HP, Intel, Lucent, Microsoft, and Philips
  * USB 1.1 supports
    - » Low-speed devices (1.5 Mbps)
    - » Full-speed devices (12 Mbps)
  * USB 2.0 supports
    - » High-speed devices
      - – Up to 480 Mbps (a factor of 40 over USB 1.1)
    - » Uses the same connectors
      - – Transmission speed is negotiated on device-by-device basis

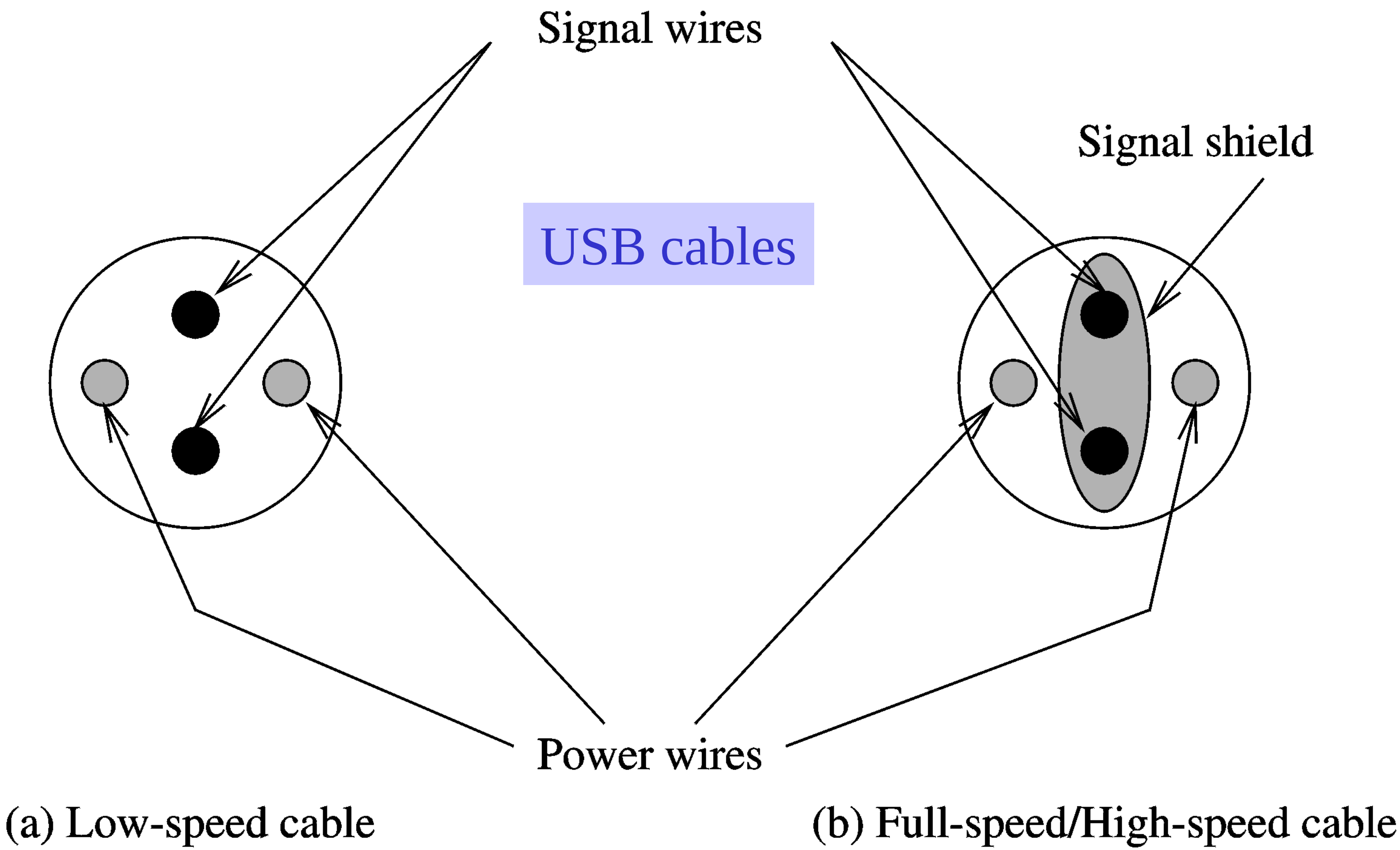
USB (cont'd)

- Motivation for USB
  - ∗ Avoid device-specific interfaces
    - » Eliminates multitude of interfaces
      - – PS/2, serial, parallel, monitor, microphone, keyboard,…
  - ∗ Avoid non-shareable interfaces
    - » Standard interfaces support only one device
  - ∗ Avoid I/O address space and IRQ problems
    - » USB does not require memory or address space
  - ∗ Avoid installation and configuration problems
    - » Don't have to open the box to install and configure jumpers
  - ∗ Allow hot attachment of devices

USB (cont'd)

- Additional advantages of USB
  - Power distribution
    - Simple devices can be bus-powered
      - Examples: mouse, keyboards, floppy disk drives, wireless LANs, …
  - Control peripherals
    - Possible because USB allows data to flow in both directions
  - Expandable through hubs
  - Power conservation
    - Enters suspend state if there is no activity for 3 ms
  - Error detection and recovery
    - Uses CRC

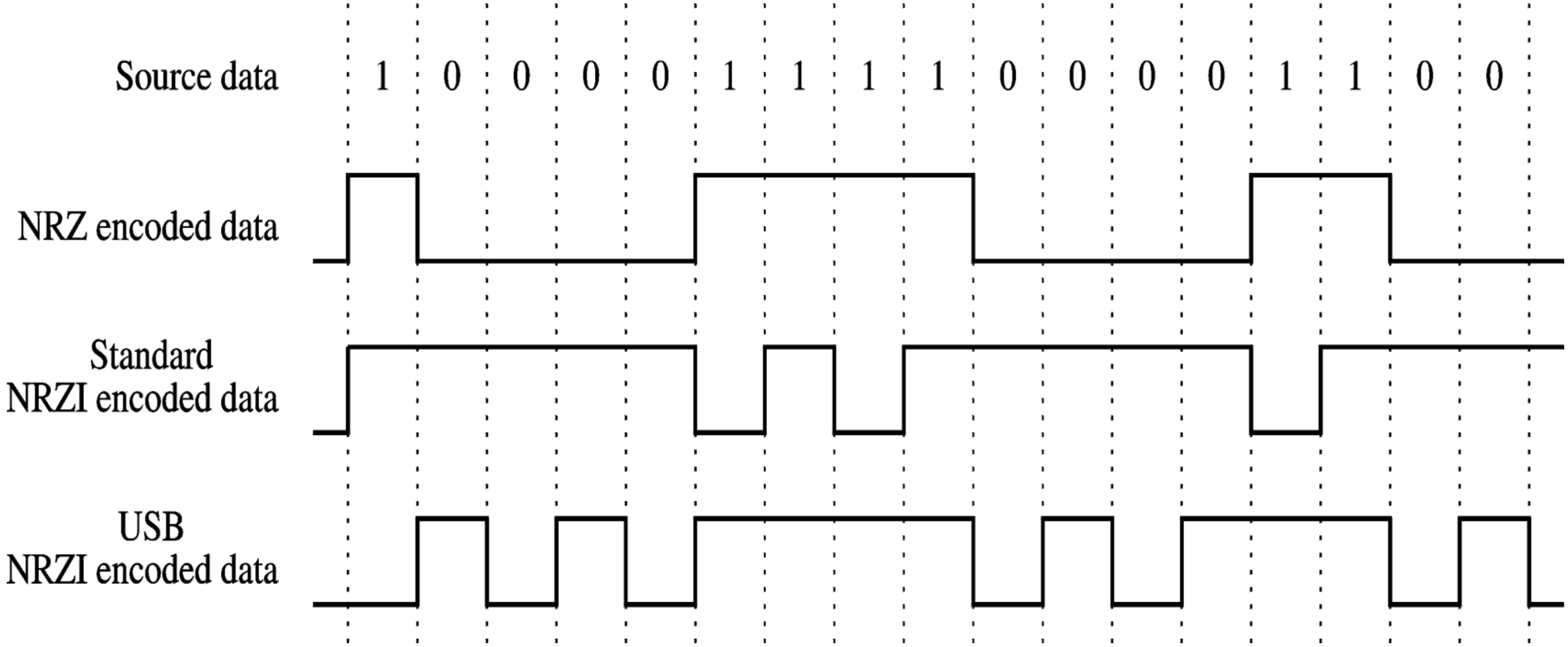
USB (cont'd)



Signal wires
Signal shield
USB cables
Power wires
(a) Low-speed cable
(b) Full-speed/High-speed cable

USB (cont'd)

- USB encoding
  - ∗ Uses NRZI encoding
    - » Non-Return to Zero-Inverted

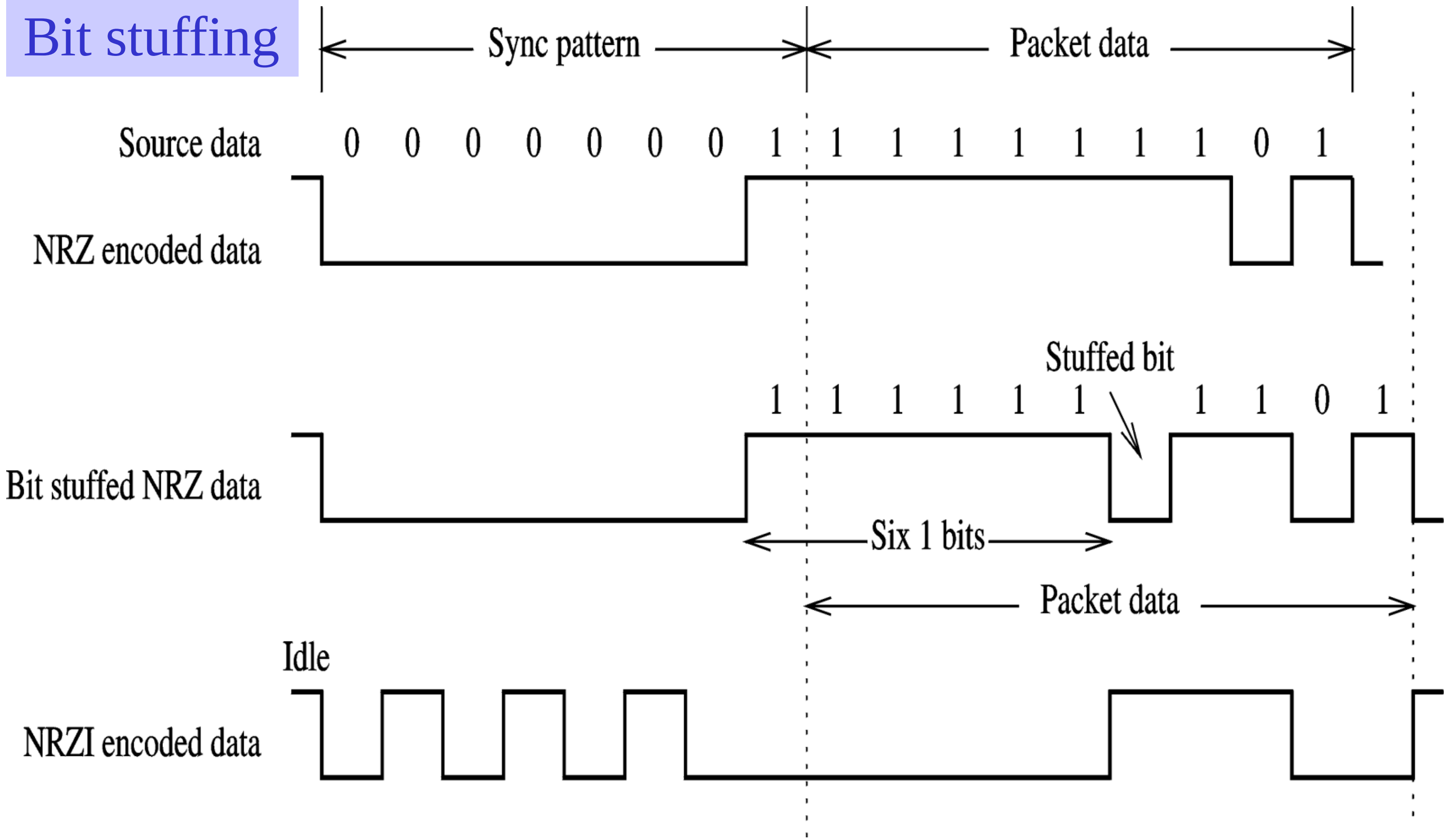


USB (cont'd)

- NRZI encoding
  - A signal transition occurs if the next bit is zero
    - It is called ***differential encoding***
  - Two desirable properties
    - Signal transitions, not levels, need to be detected
    - Long string of zeros causes signal changes
  - Still a problem
    - Long strings of 1s do not causes signal change
  - To solve this problem
    - Uses ***bit stuffing***
      - A zero is inserted after every six consecutive 1s

USB (cont'd)

Bit stuffing



To be used with S. Dandamudi, "Fundamentals of Computer Organization and Design," Springer, 2003.

USB (cont'd)

- Transfer types
  - » Four types of transfer
  - ∗ Interrupt transfer
    - » Uses polling
      - – Polling interval can range from 1 ms to 255 ms
  - ∗ Isochronous transfer
    - » Used in real-time applications that require constant data transfer rate
      - – Example: Reading audio from CD-ROM
    - » These transfers are scheduled regularly
    - » Do not use error detection and recovery

USB (cont'd)

* Control transfer
  » Used to configure and set up USB devices
  » Three phases
    – Setup stage
      ✈ Conveys type of request made to target device
    – Data stage
      ✈ Optional stage
      ✈ Control transfers that require data use this stage
    – Status stage
      ✈ Checks the status of the operation
  » Allocates a guaranteed bandwidth of 10%
  » Error detection and recovery are used
    – Recovery is by means of retries

USB (cont'd)

* Bulk transfer
  - » For devices with no specific data transfer rate requirements
    - – Example: sending data to a printer
  - » Lowest priority bandwidth allocation
  - » If the other three types of transfers take 100% of the bandwidth
    - – Bulk transfers are deferred until load decreases
  - » Error detection and recovery are used
    - – Recovery is by means of retries

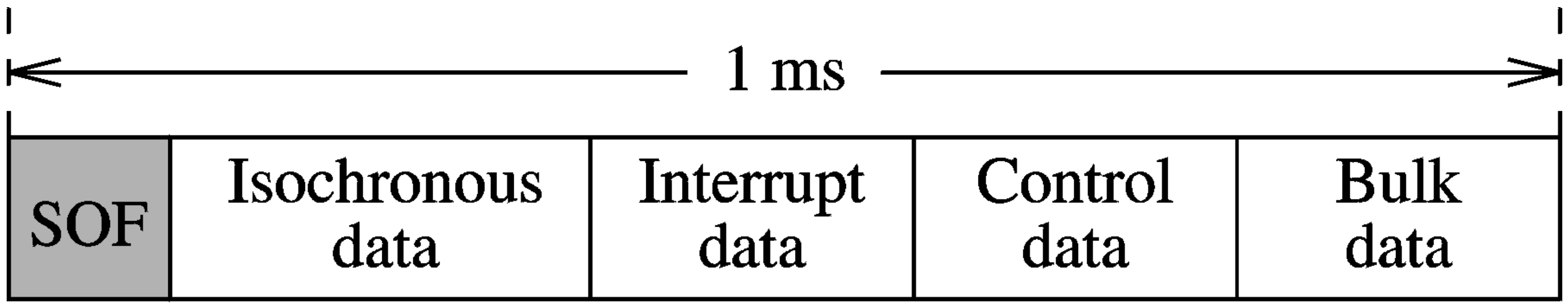
USB (cont'd)

- USB architecture
  - USB host controller
    - Initiates transactions over USB
  - Root hub
    - Provides connection points
  - Two types of host controllers
    - Open host controller (OHC)
      - Defined by Intel
    - Universal host controller (UHC)
      - Specified by National Semiconductor, Microsoft, Compaq
    - Difference between the two
      - How they schedule the four types of transfers

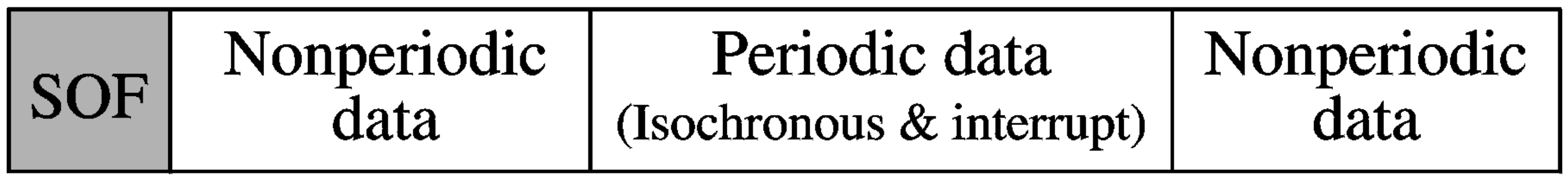
USB (cont'd)

- UHC scheduling
  - Schedules periodic transfers first
    - Periodic transfers: isochronous and interrupts
    - Can take up to 90% of bandwidth
  - These transfers are followed by control and bulk transfers
    - Control transfers are guaranteed 10% of bandwidth
  - Bulk transfers are scheduled only if there is bandwidth available

USB (cont'd)



(a) UHC scheduling



(b) OHC scheduling

USB (cont'd)

- OHC scheduling
  - ∗ Different from UHC scheduling
  - ∗ Reserves space for non-periodic transfers first
    - » Non-periodic transfers: control and bulk
    - » 10% bandwidth reserved
  - ∗ Next periodic transfers are scheduled
    - » Guarantees 90% bandwidth
  - ∗ Left over bandwidth is allocated to non-periodic transfers

To be used with S. Dandamudi, "Fundamentals of Computer Organization and Design," Springer, 2003.

USB (cont'd)

- Bus powered devices
  - Low-power
    - Less than 100 mA
    - Can be bus-powered
  - High-powered
    - Between 100 mA and 500 mA
      - Full-powered ports can power these devices
    - Can be designed to have their own power
    - Operate in three modes
      - Configured (500 mA)
      - Unconfigured (100 mA)
      - Suspended (about 2.5 mA)

USB (cont'd)

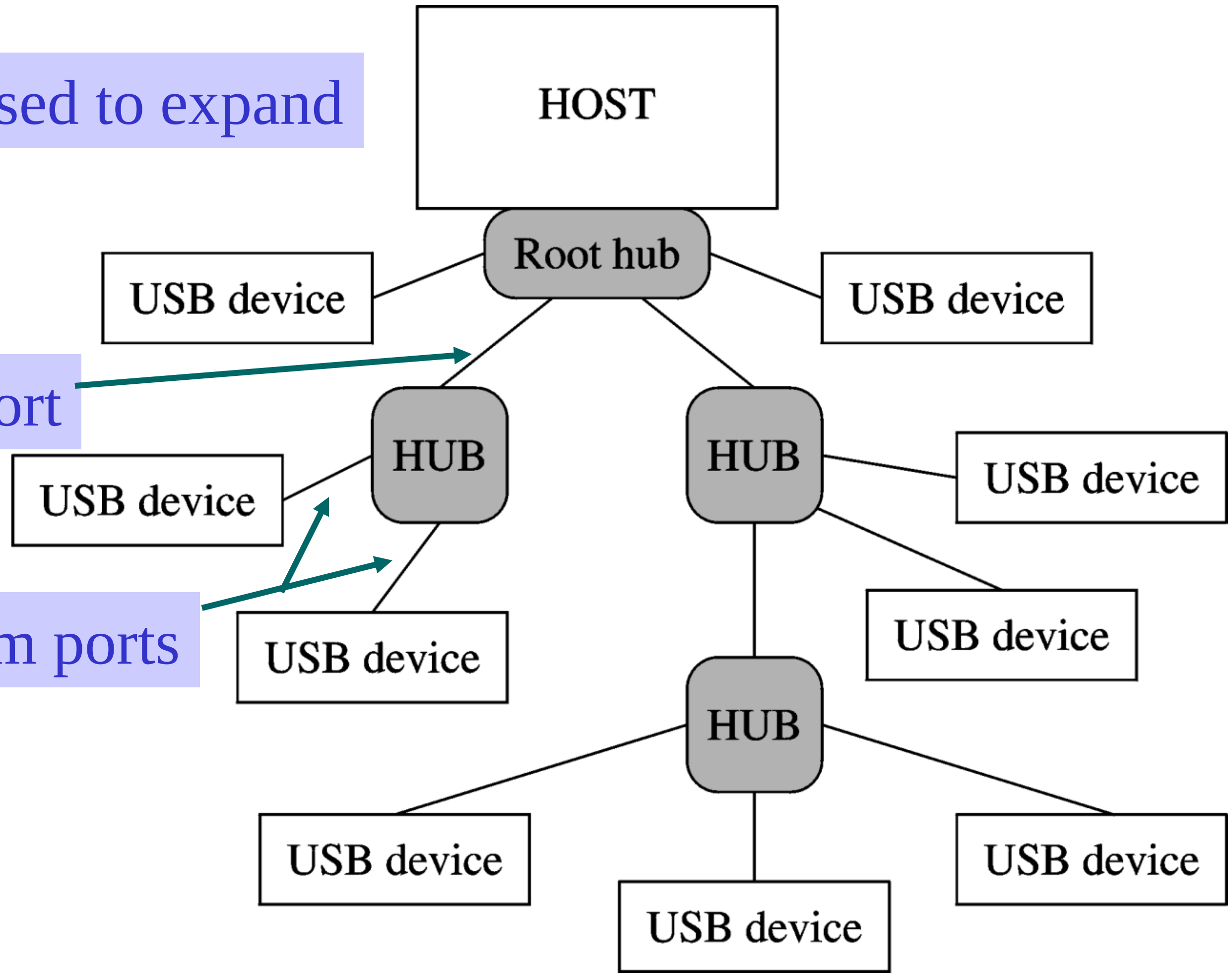
- USB hubs
  - Bus-powered
    - No extra power supply required
    - Must be connected to an upstream port that can supply 500 mA
    - Downstream ports can only supply 100 mA
      - Number of ports is limited to four
      - Support only low-powered devices
  - Self-powered
    - Support 4 high-powered devices
    - Support 4 bus-powered USB hubs
  - Most 4-port hubs are dual-powered

USB (cont'd)

Hubs can be used to expand

Upstream port

Downstream ports

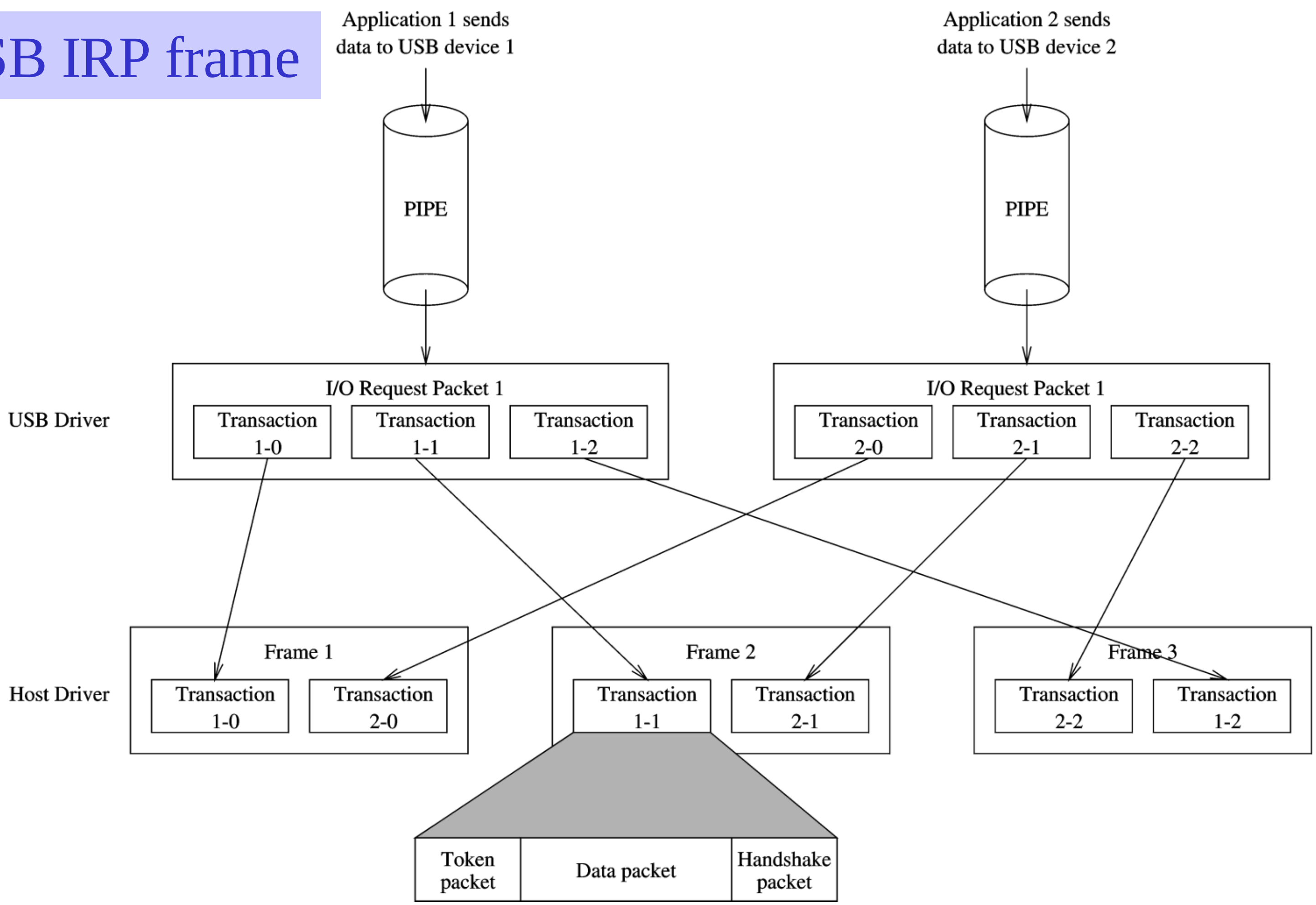


USB (cont'd)

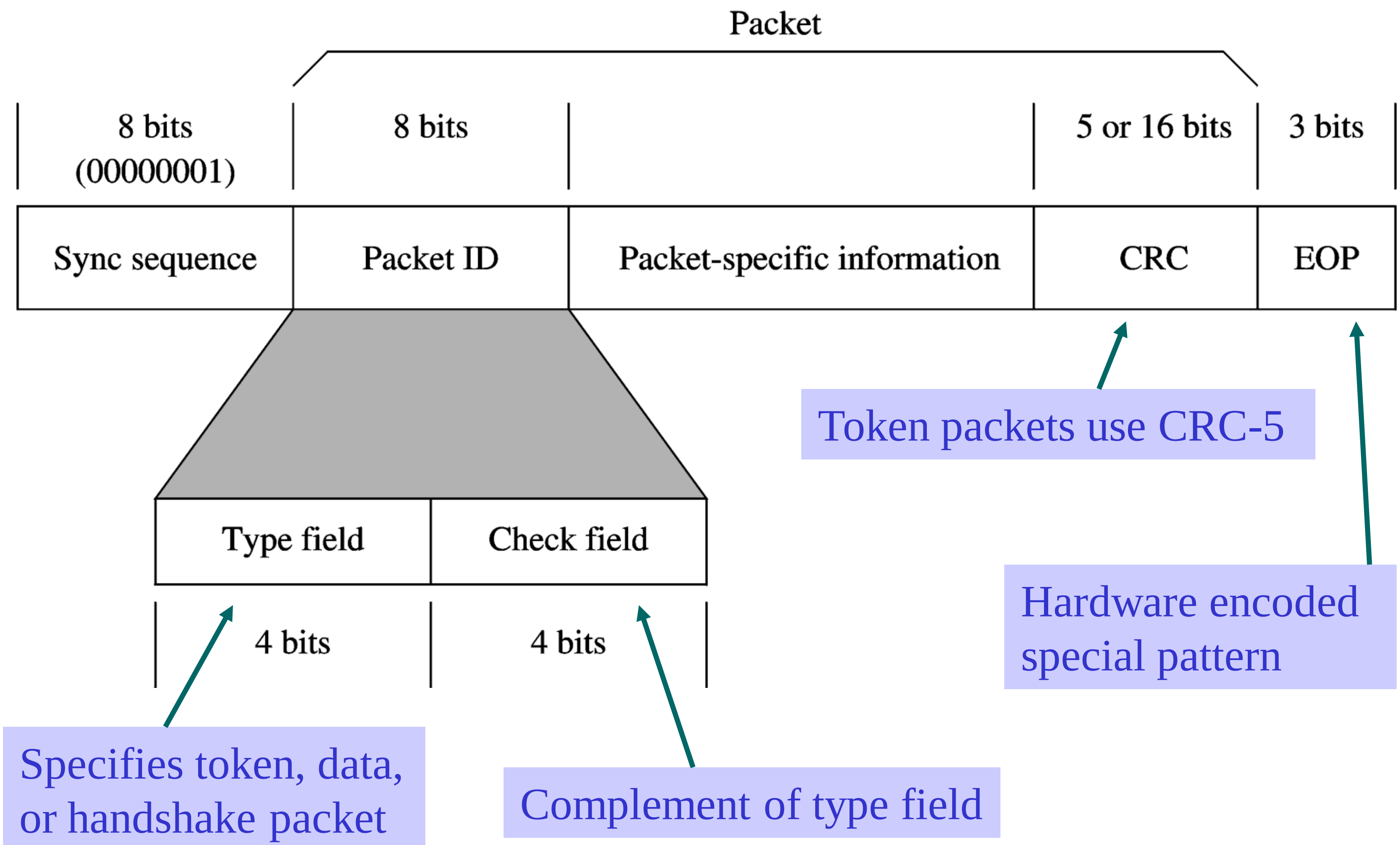
- USB transactions
  - Transfers are done in one or more transactions
    - » Each transaction consists of several packets
  - Transactions may have between 1 and 3 phases
    - » Token packet phase
      - – Specifies transaction type and target device address
    - » Data packet phase (optional)
      - – Maximum of 1023 bytes are transferred
    - » Handshake packet phase
      - – Except for isochronous transfers, others use error detection for guaranteed delivery
      - – Provides feedback on whether data has been received without error

USB (cont'd)

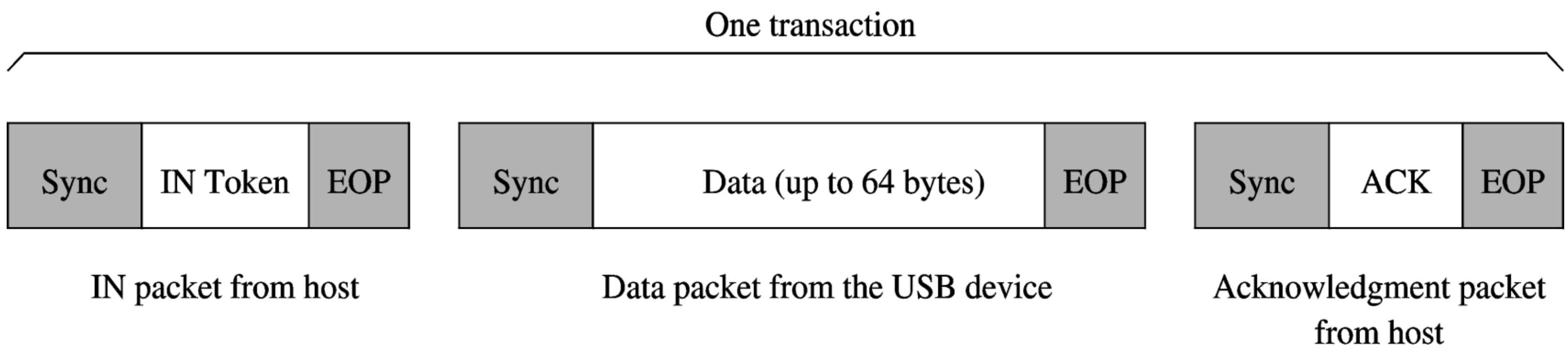
USB IRP frame



USB (cont'd)

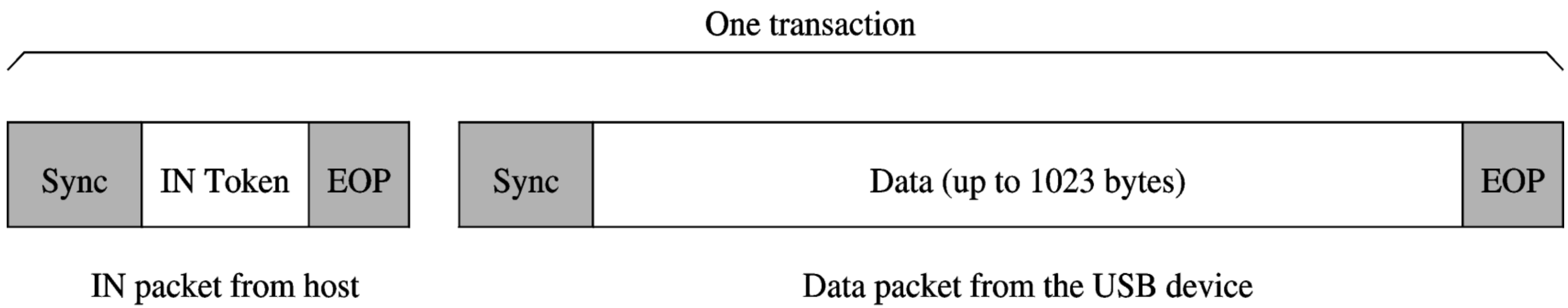

Packet
8 bits
(00000001)
8 bits
5 or 16 bits
3 bits
Sync sequence
Packet ID
Packet-specific information
CRC
EOP
Token packets use CRC-5
Type field
Check field
4 bits
4 bits
Hardware encoded special pattern
Specifies token, data, or handshake packet
Complement of type field

USB (cont'd)



USB 1.1 transactions

(a) IN transaction without errors



(b) IN transaction during an isochronous transfer

USB (cont'd)

- USB 2.0
  - USB 1.1 uses 1 ms frames
  - USB 2.0 uses 125 μs frames
    - 1/8 of USB 1.1
  - Supports 40X data rates
    - Up to 480 Mbps
  - Competitive with
    - SCSI
    - IEEE 1394 (FireWire)
  - Widely available now

IEEE 1394

- Apple originally developed this standard for high-speed peripherals
  - ∗ Known by a variety of names
    - » Apple: FireWire
    - » Sony: i.ILINK
  - ∗ IEEE standardized it as IEEE 1394
    - » First released in 1995 as IEEE 1394-1995
    - » A slightly revised version as 1394a
    - » Next version 1394b
  - ∗ Shares many of the features of USB

IEEE 1394 (cont'd)

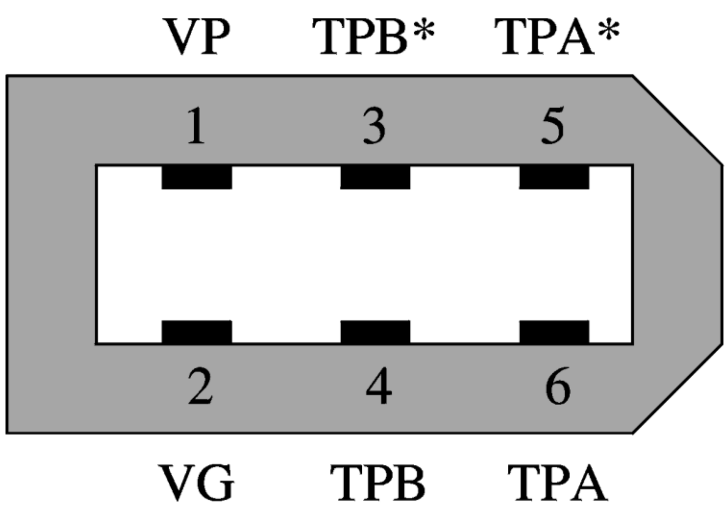
- Advantages
  - ∗ High speed
    - » Supports three speeds
      - – 100, 200, 400 Mbps
        - ✈Competes with USB 2.0
      - – Plans to boost it to 3.2 Gbps
  - ∗ Hot attachment
    - » Like USB
    - » No need to shut down power to attach devices
  - ∗ Peer-to-peer support
    - » USB is processor-centric
    - » Supports peer-to-peer communication without involving the processor

IEEE 1394 (cont'd)

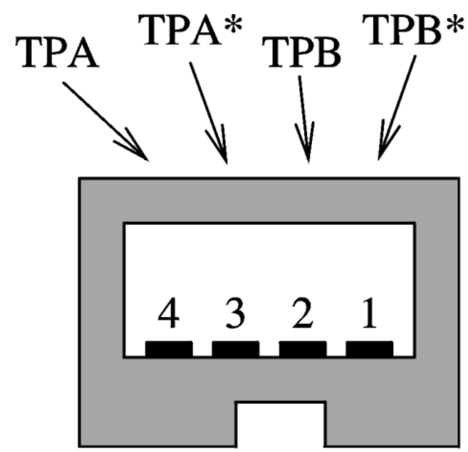
* Expandable bus
  * » Devices can be connected in daisy-chain fashion
  * » Hubs can used to expand
* Power distribution
  * » Like the USB, cables distribute power
    * – Much higher power than USB
      * ✈Voltage between 8 and 33 V
      * ✈Current an be up to 1.5 Amps
* Error detection and recovery
  * » As in USB, uses CRC
  * » Uses retransmission in case of error
* Long cables
  * » Like the USB

IEEE 1394 (cont'd)

IEEE 1394 6-pin and 4-pin connectors



(a) 6-pin connector

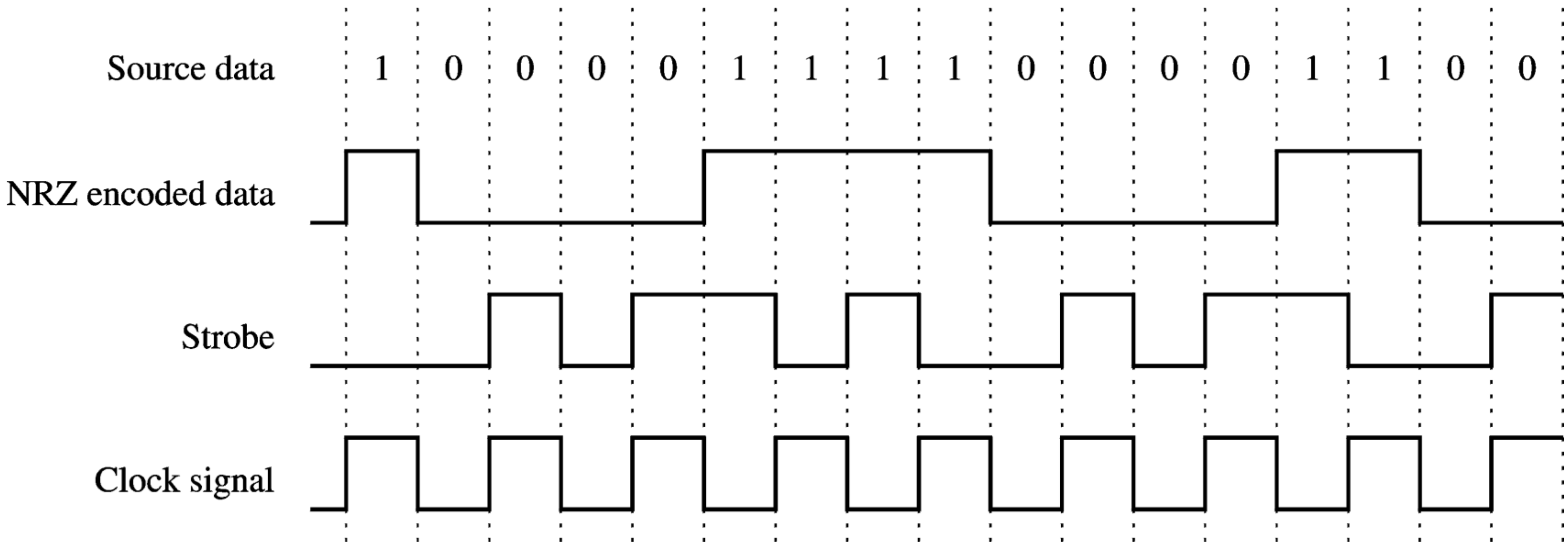


(b) 4-pin connector

4-pin connector does not distribute power

IEEE 1394 (cont'd)

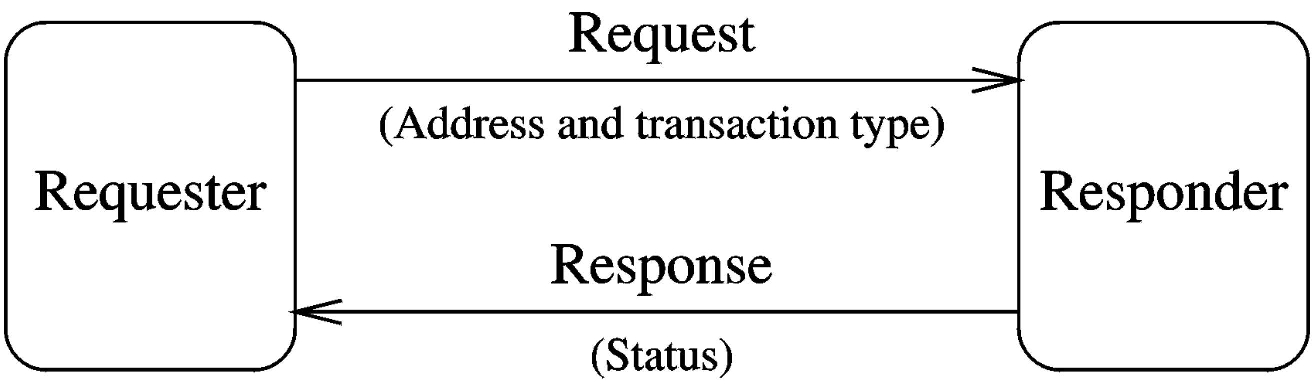
- Encoding
  - ∗ Uses a simple NRZ encoding
  - ∗ Strobe signal is encoded
    - » Changes the signal even if successive bits are the same



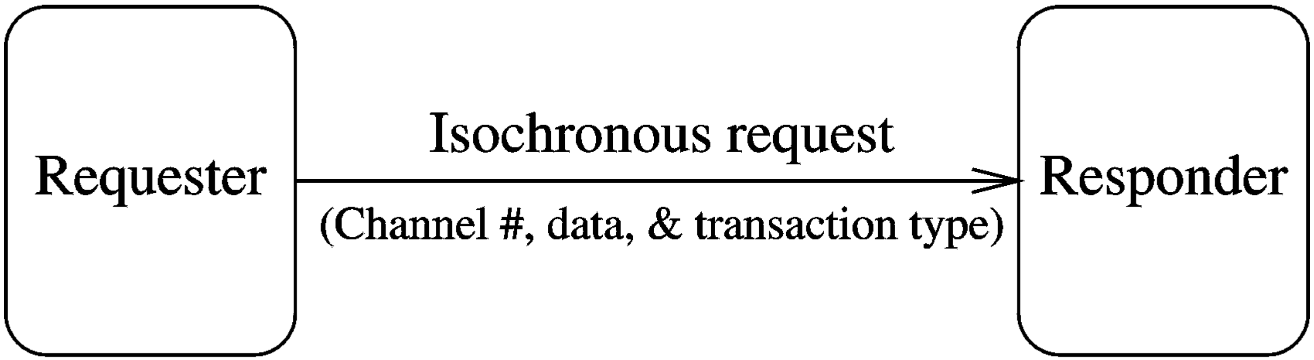
IEEE 1394 (cont'd)

- Transfer types
  - Asynchronous
    - » For applications that require correct delivery of data
      - – Example: writing a file to a disk drive
    - » Uses an acknowledgement to confirm delivery
    - » Guaranteed bandwidth of 20%
  - Isochronous
    - » For real-time applications
    - » No acknowledgement
    - » Up to 80% of bandwidth allocated
  - Bandwidth allocation on a cycle-by-cycle basis
    - » Cycle time: 125 μs

IEEE 1394 (cont'd)



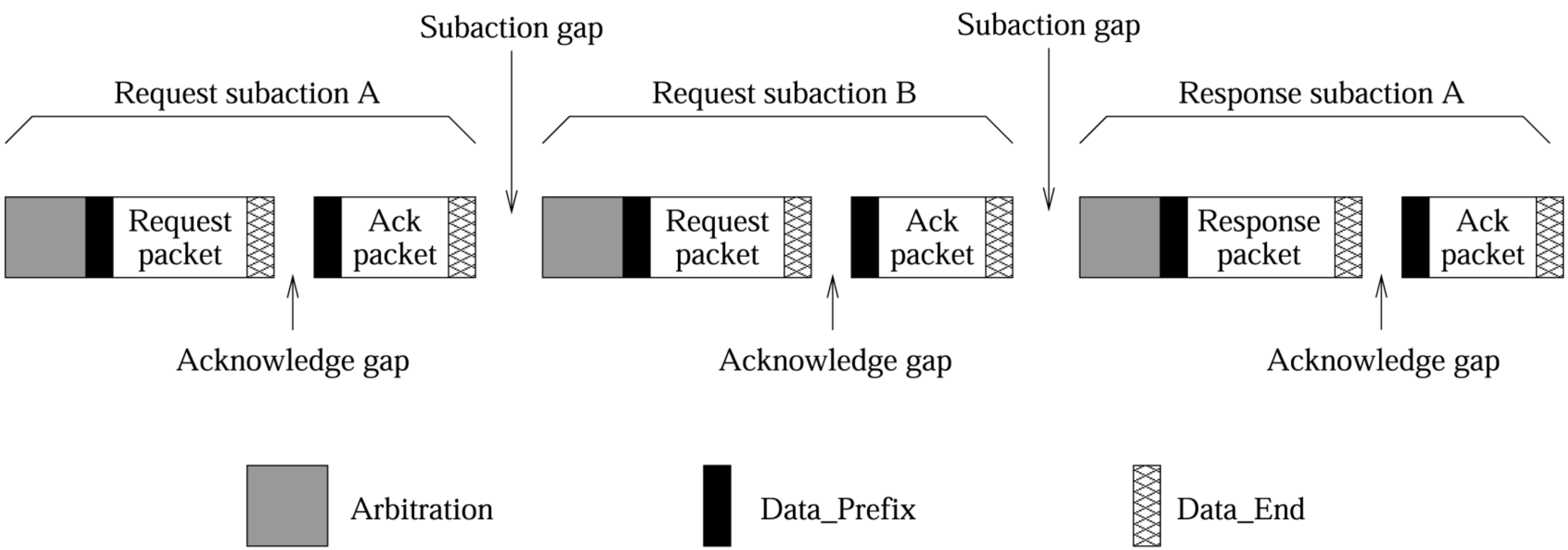
(a) Asynchronous transfer



(b) Isochronous transfer

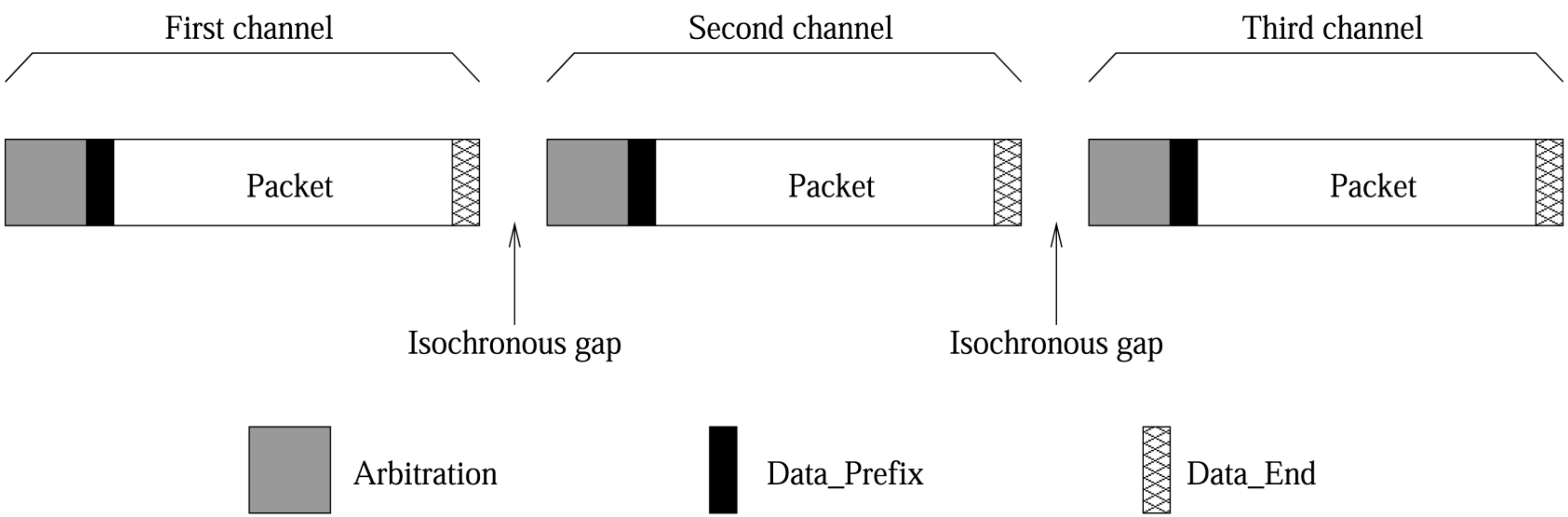
IEEE 1394 (cont'd)

- Transactions
  * Follow request and reply format
  * Each packet is encapsulated between **`Data_Prefix`** and **`Data_end`**



IEEE 1394 (cont'd)

- Isochronous transactions
  - ∗ Similar to asynchronous transactions
  - ∗ Main difference:
    - » No acknowledgement packets



IEEE 1394 (cont'd)

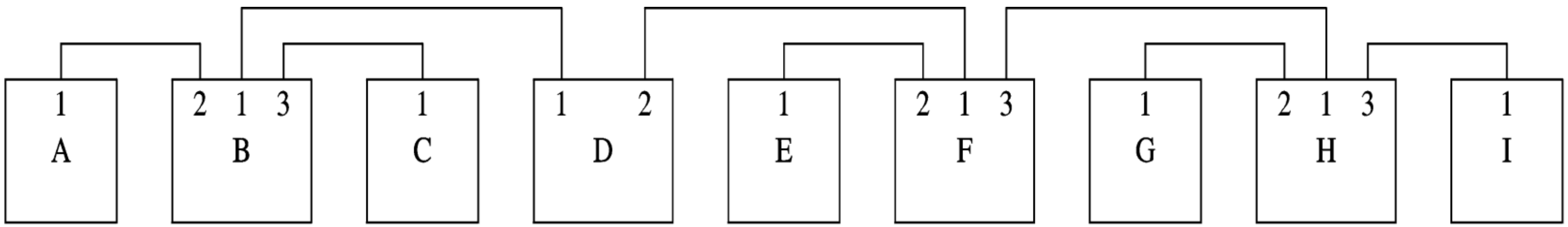
- Bus arbitration
  - Needed because of peer-to-peer communication
  - Arbitration must respect
    - Bandwidth allocation to isochronous channels
    - Fairness-based allocation for asynchronous channels
  - Uses fairness interval
    - During each interval
      - All nodes with pending asynchronous transaction are allowed bus ownership once
  - Nodes with pending isochronous transactions go through arbitration during each cycle
  - IRM is used for isochronous bandwidth allocations

IEEE 1394 (cont'd)

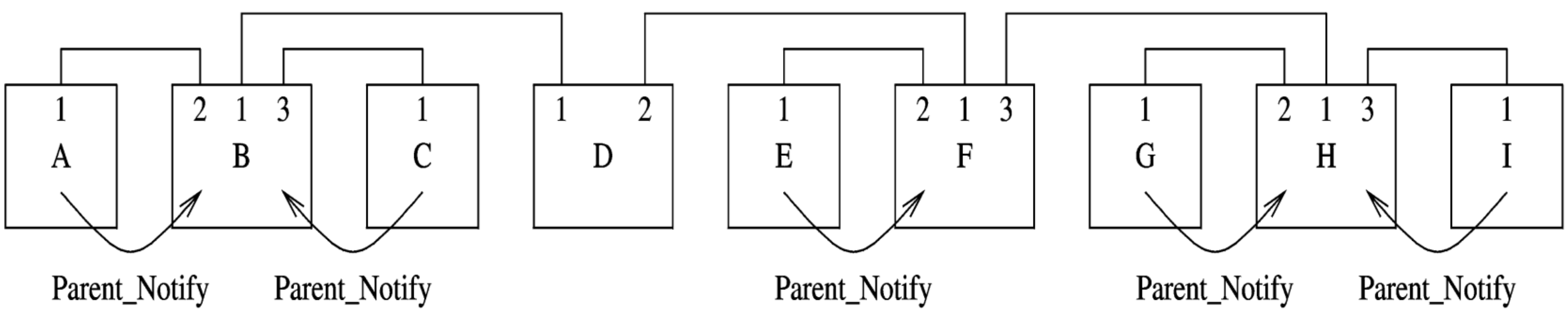
- Configuration
  - ∗ Does not require the host system
  - ∗ Consists of two main phases
    - » Tree identification
      - – Used to find the network topology
      - – Uses two special signals
        - → **`Parent_notify`** and **`Child_Notify`**
    - » Self-identification
      - – Done after the tree identification
      - – Assigns unique ids to nodes

IEEE 1394 (cont'd)

Tree identification



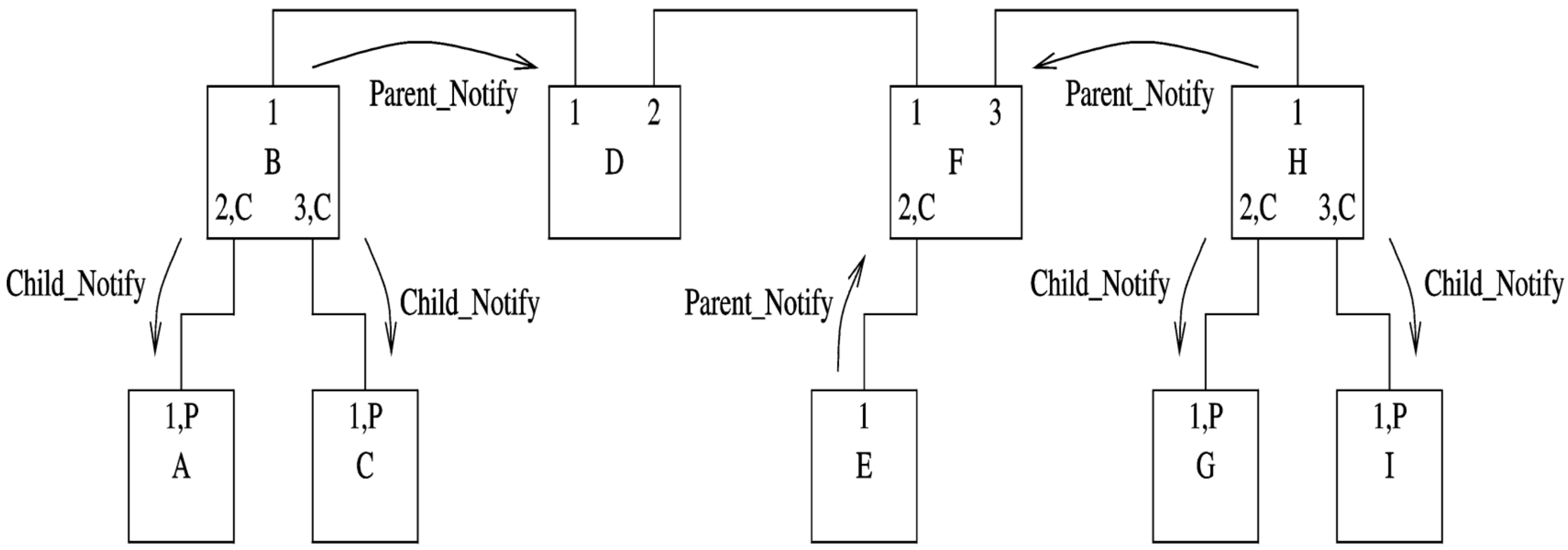
(a) Original unconfigured network



(b) Leaf nodes send Parent_Notify signal to their parent nodes

IEEE 1394 (cont'd)

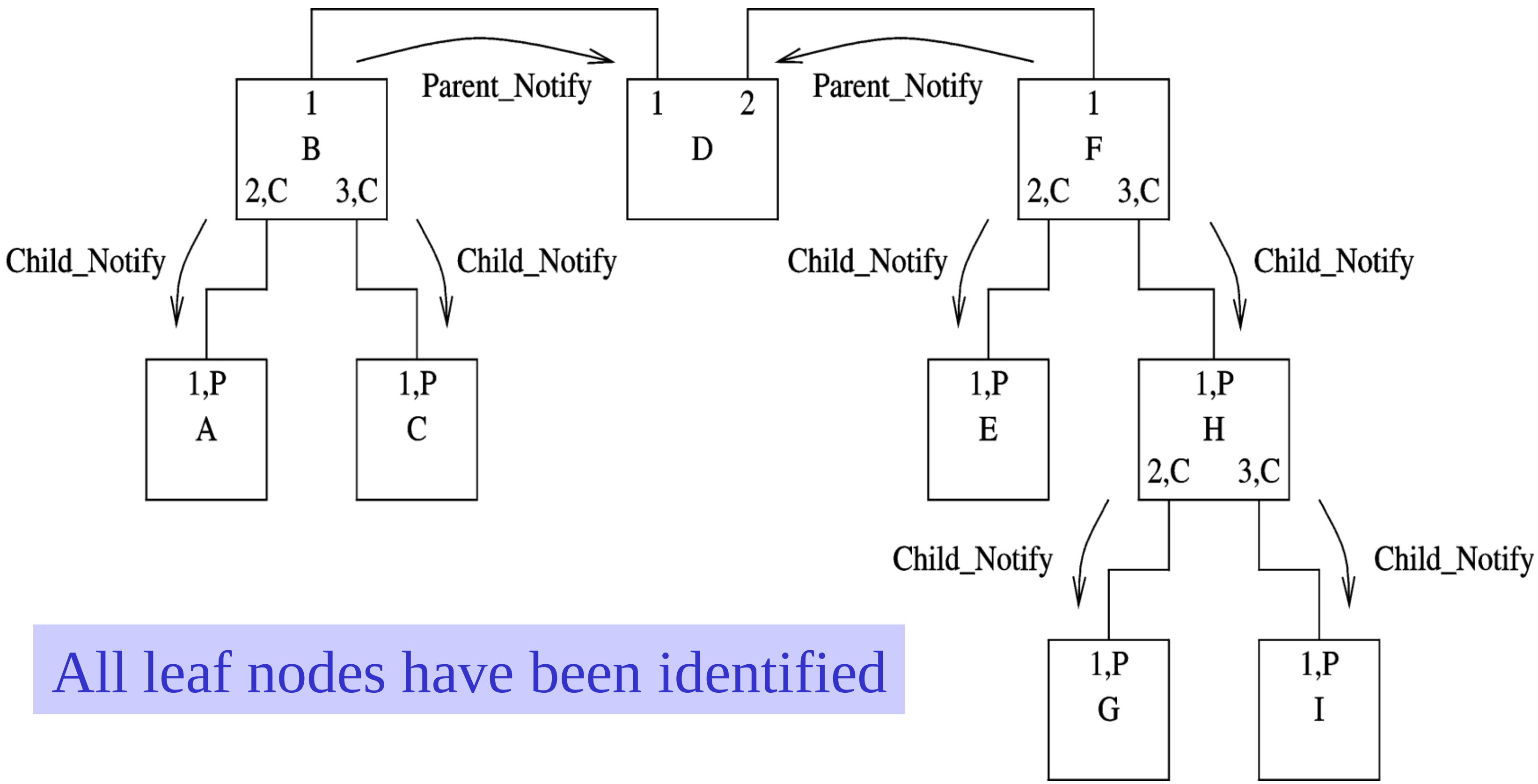
Tree identification



(c) Topology starts to take shape with nodes A, C, E, G, and I identified as leaf nodes

IEEE 1394 (cont'd)

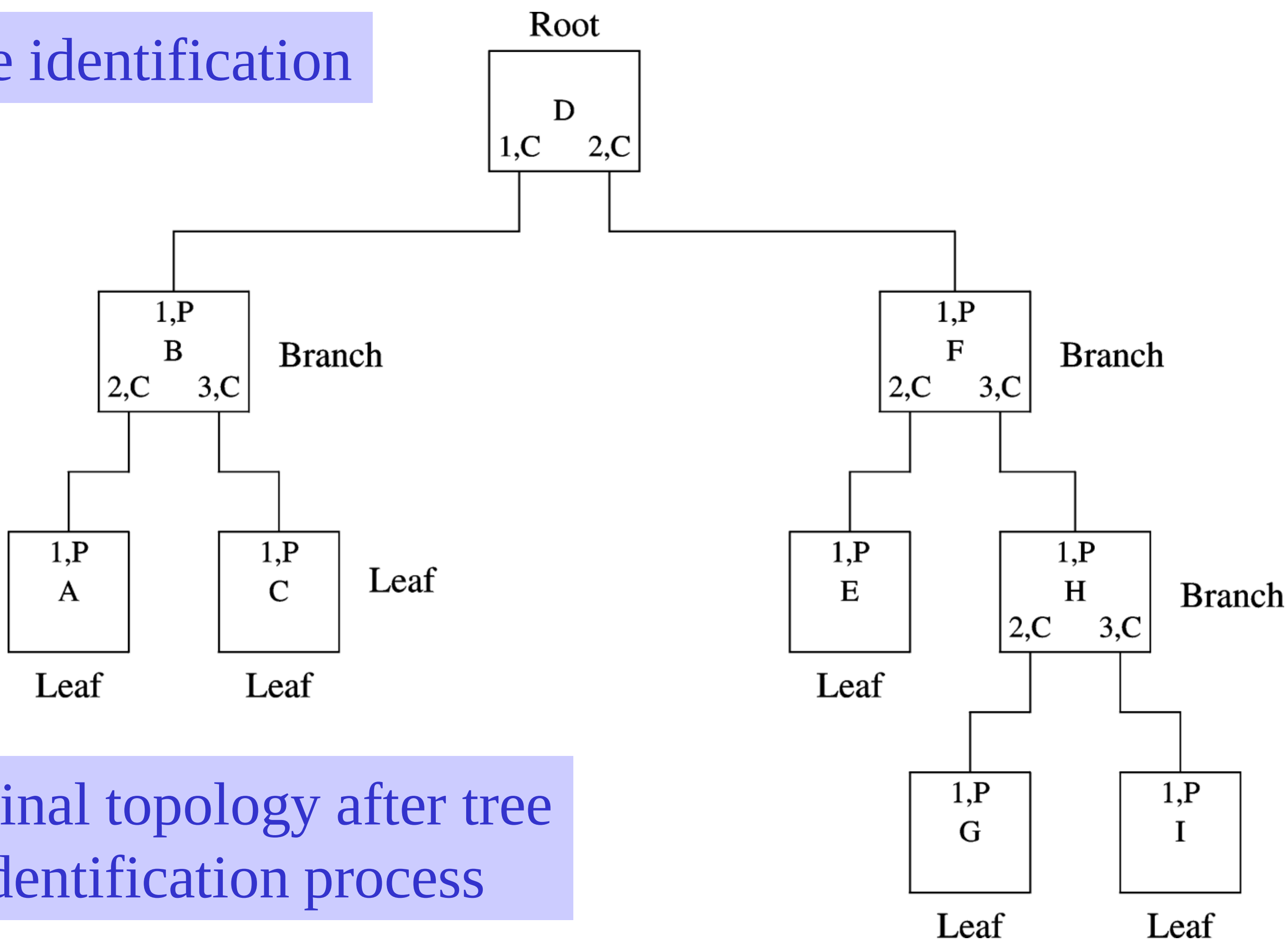
Tree identification



All leaf nodes have been identified

IEEE 1394 (cont'd)

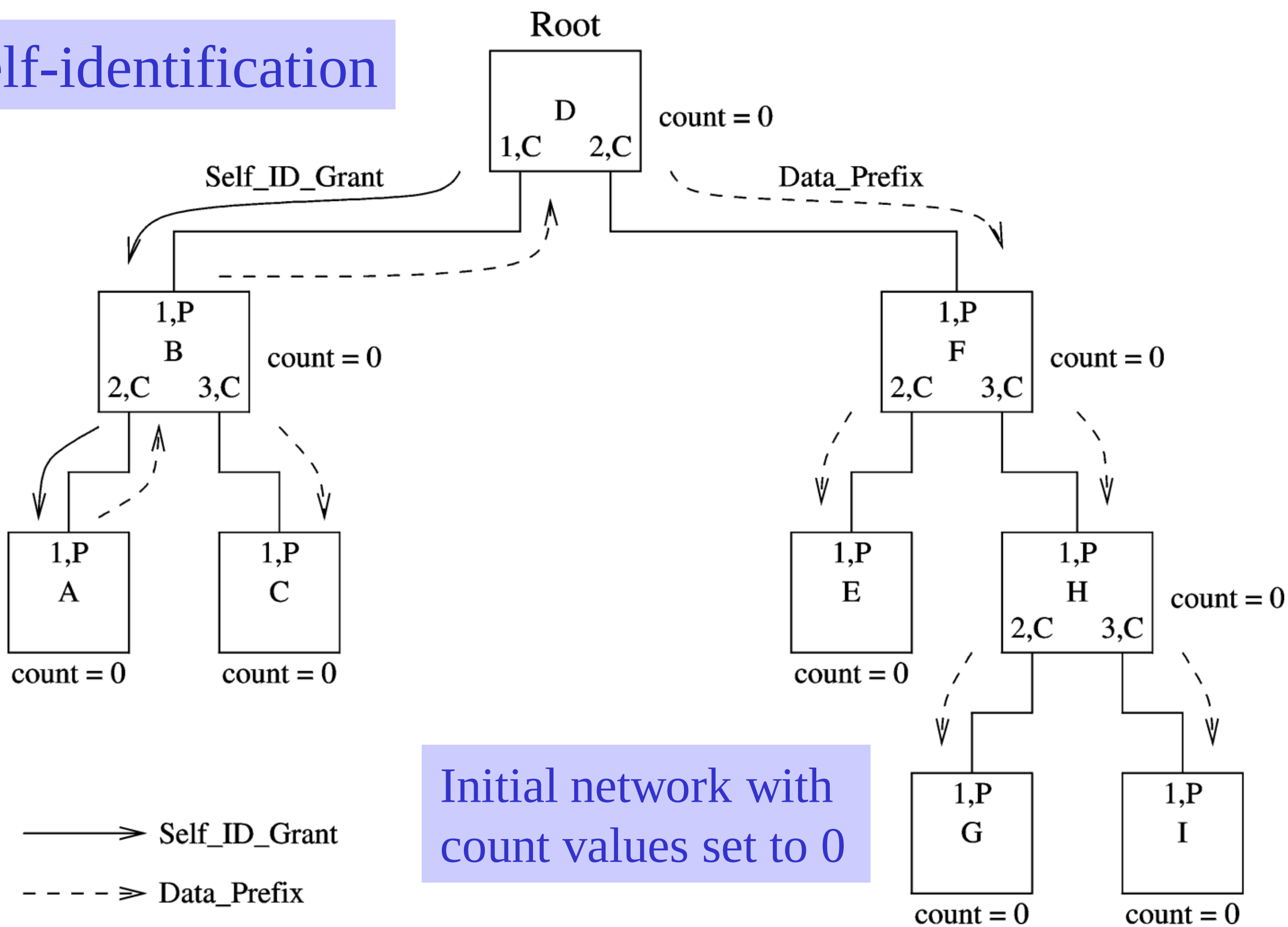
Tree identification



Final topology after tree identification process

IEEE 1394 (cont'd)

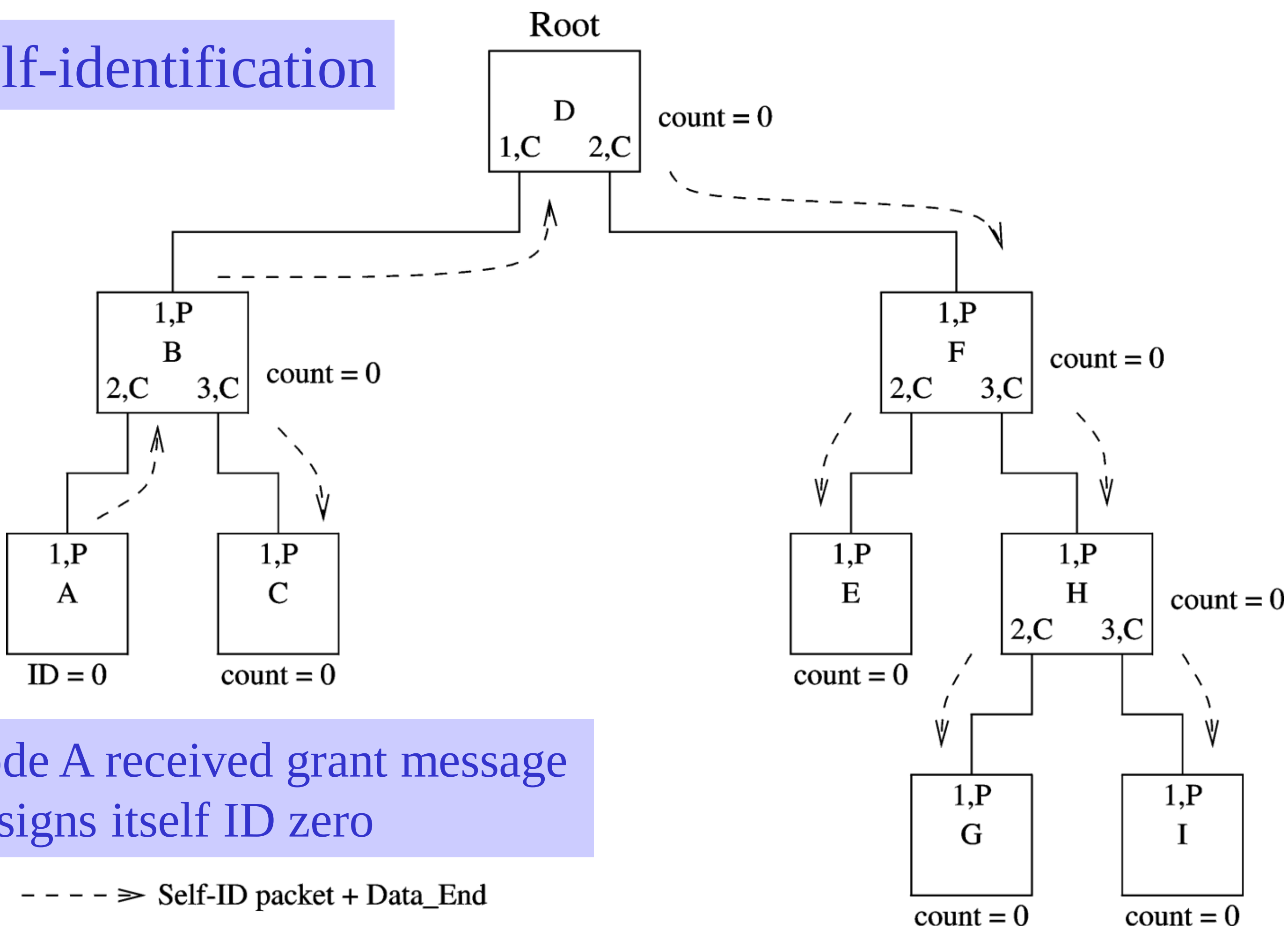
Self-identification



Initial network with count values set to 0

IEEE 1394 (cont'd)

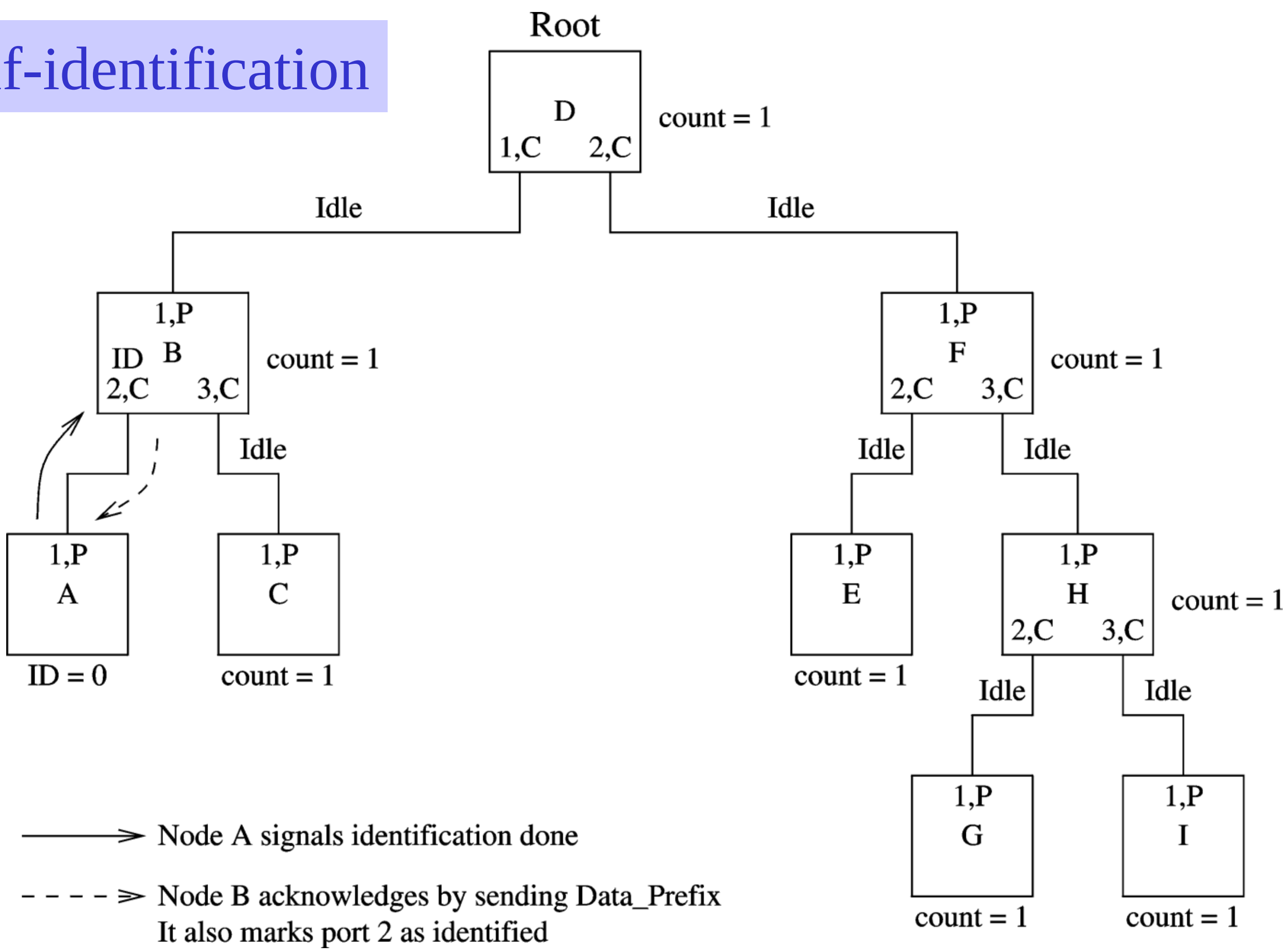
Self-identification



Node A received grant message
Assigns itself ID zero

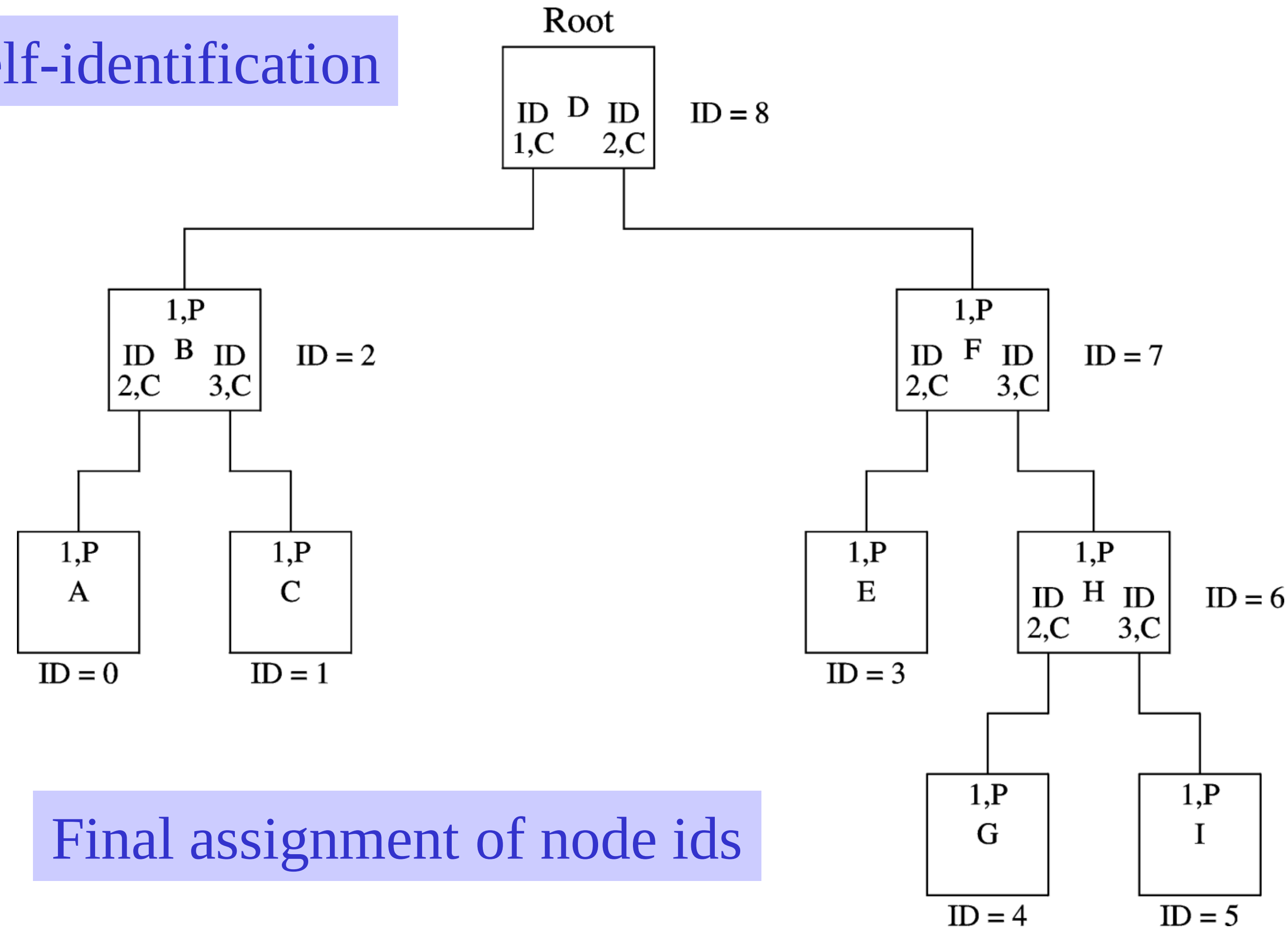
IEEE 1394 (cont'd)

Self-identification



IEEE 1394 (cont'd)

Self-identification



Final assignment of node ids

Bus Wars

- SCSI is dominant in disk and storage device interfaces
  - ∗ Parallel interface
  - ∗ Its bandwidth could go up to 640 MB/s
- IEEE 1394
  - ∗ Serial interface
  - ∗ Supports peer-to-peer applications
  - ∗ Dominant in video applications
- USB
  - ∗ Useful in low-cost, host-to-peripheral applications
  - ∗ USB 2.0 provides high-speed support

Last slide