Protocols for Wired and Wireless Networks in Vehicle Systems

Syed Masud Mahmud, Ph.D.
Electrical and Computer Engg. Dept.
Wayne State University
Detroit MI 48202

Copyright © 2002 by Syed Masud Mahmud

Email: smahmud@eng.wayne.edu
Phone: (313) 577-3855
Fax: (313) 577-1101
Webpage: http://www.ece.eng.wayne.edu/~smahmud

References

- **This talk has been prepared based on the materials presented in the following references:**

1. **CAN Protocol Standard, Motorola, Document Number: BCANPSV2.0.**
2. **Overview of CAN, Motorola.**
3. **VAN in details and Technology, http://www.van-mux.org/concept0.htm**
4. **Introduction to LIN, Hans- Christian von der Wense, Munich, Germany, Motorola, March 2000.**
5. **Specification of the Bluetooth System Vol. 1 & 2. http://www.bluetooth.com**
6. **Bluetooth Tutorial – Specifications, Palowireless, http://www.palowireless.com/infotooth/tutorial.asp**

References

7. Bluetooth Tutorial, Aman Kansal, http://www.bluetooth.amankansal.com
8. i-Beans and Bluetooth, Millennial Net, http://www.millennial.net/blue.htm
9. Computer Networking Basics, by Carlo Fonda and Fulvio Postogna, The Abdus Salam ICTP, http://www.ictp.trieste.it/~radionet/1998_school/networking_presentation/index.html
10. MC68377 Reference Manual, Motorola, 2000.
11. Mobic Mobile Community, http://www.mobic.com/termtech/bluetooth1_0-body.html
12. M. J. Schofield's web site http://www.mjschofield.com/implment.htm

Specification of the Bluetooth System Volume 1 & 2.

Copyright © 1999 by
Ericsson, IBM, Intel, Nokia,Toshiba

Articles Contributed by
3COM, Combit Inc., Convergence, Ericsson, Extended Systems, IBM, Intel, Motorola, Nokia, Puma Technology, Toshiba, Xtraworx

Target Audience For This Workshop

- Those people who just started working or are planning to work in the area of CAN and BLUETOOTH.
- Some background in digital logic, microprocessors and microcontrollers would help to understand the talk.

Overview of the Talk

Seven Layers of OSI

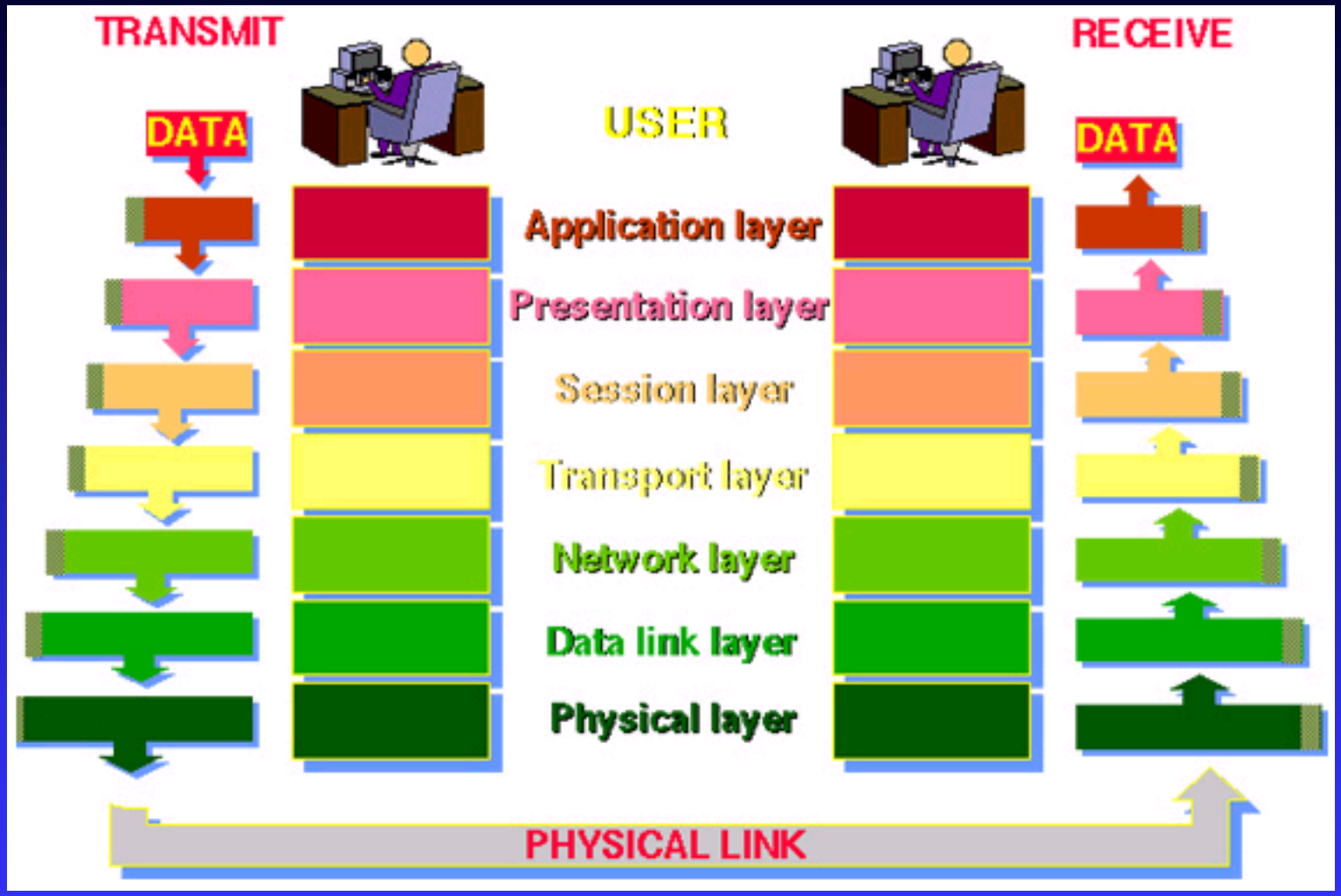
Wired Networks

- CAN (Controller Area Network)
- VAN (Vehicle Area Network)
- Other Network Protocols, e.g. J1850, LIN, BEAN

Wireless Networks

- **Bluetooth**
- **i-BEAN**

The Seven Layers of OSI



Courtesy of The Abdus Salam International Centre for Theoretical Physics.
http://www.ictp.trieste.it/~radionet/1998_school/networking_presentation/OSI-layers.html

Layer 7: Application Layer

- This layer supports a collection of miscellaneous protocols for high level applications.
- Electronic mail, file transfer, connecting remote terminals, etc.
- Examples: SMTP, FTP, Telnet, HTTP, etc.

Layer 6: Presentation Layer

- Concerned with the semantics of the bits.
- Define records and fields in them.
- Sender can tell the receiver of the format.
- Makes machines with different internal representations to communicate.
- If implemented, the best layer for cryptography.

Layer 5: Session Layer

- Just theory! Very few applications use it.
- Enhanced version of transport layer.
- Dialog control, synchronization facilities.

Layer 4: Transport Layer

- Breaks the message (from sessions layer) into smaller packets, assigns sequence number and sends them.
- Lost packets arriving out of order must be reordered.

Layer 3: Network Layer

- Concerned with the transmission of packets.
- Choose the best path to send a packet (routing).
- Shortest (distance) route vs. route with least delay.

Layer 2: Data Link Layer

- Handles errors in the physical layer.
- Groups bits into frames and ensures their correct delivery.
- Adds some bits at the beginning and end of each frame plus the checksum.
- Receiver verifies the checksum.
- If the checksum is not correct, it asks for retransmission. (send a control message).

Layer 1: Physical Layer

- Concerned with the transmission of bits.
- Voltage level for a 0, voltage level for a 1.
- Bit rate.
- Two way or one-way transmission
- Standardized protocols dealing with electrical, mechanical and signaling interfaces.

Controller Area Network (CAN)

CAN Protocol: CAN 2.0A & CAN 2.0B

- Basic Concepts & Definitions
- Identifiers & Arbitration
- Robustness & Flexibility
- Message Formats
- Errors at Message and Bit Level
- Error Handling and Confinement

Controller Area Network (CAN)

- CAN Implementations
- The Requirements of a CAN Controller
- Full CAN vs. Basic CAN Controllers
- Message Buffering & Filtering
- Motorola CAN Modules

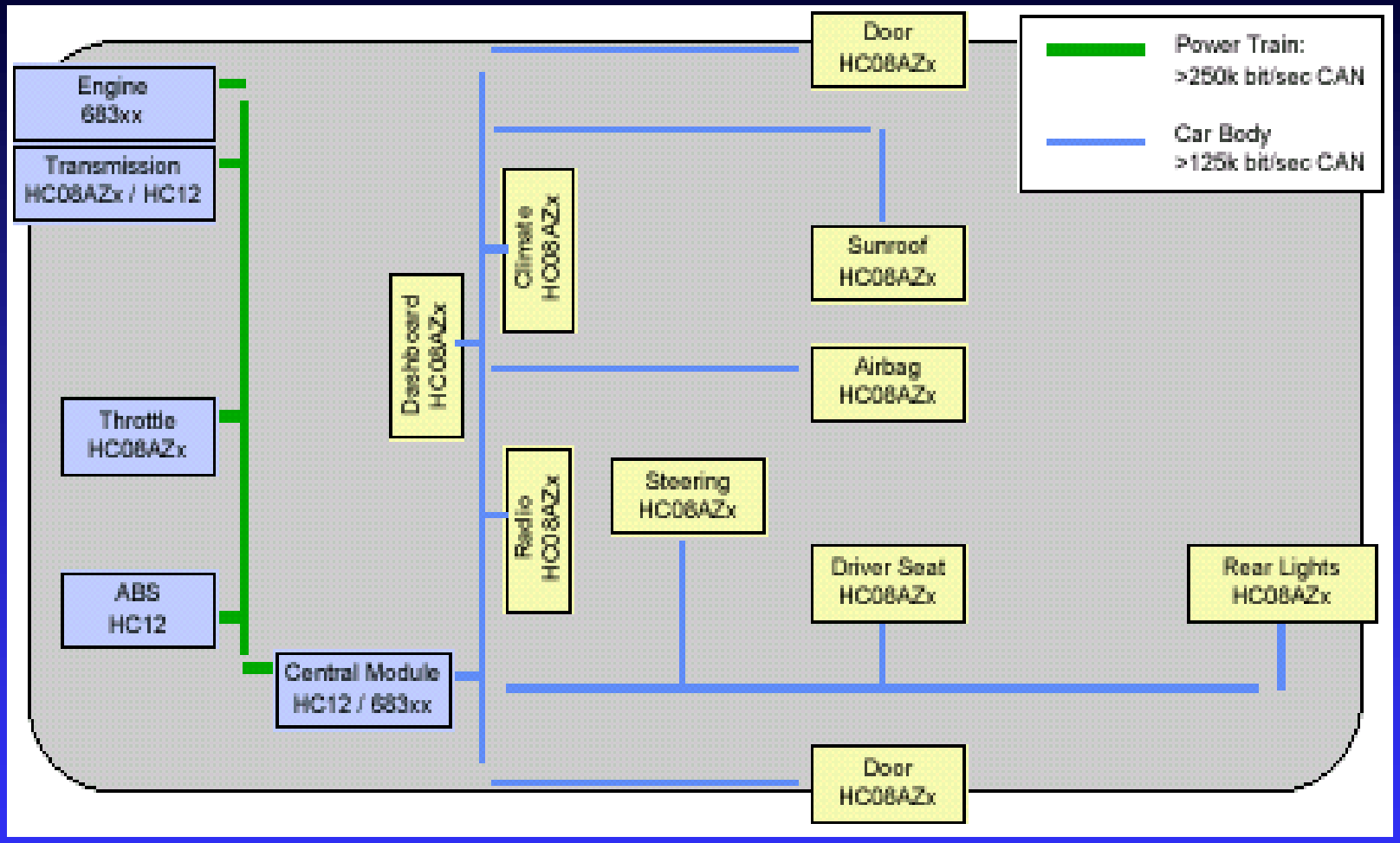
Controller Area Network (CAN)

- CAN is a serial protocol.
- It's a multimaster, multicast protocol without routing.
- It supports distributed real-time control with high level of data integrity.
- It was defined by BOSCH for automotive applications.
- Currently CAN is not restricted to auto industry.

Controller Area Network (CAN)

- It satisfies the communication needs of a wide range of applications, from high-speed networks to low-cost multiplexing.
- Multiple CANs with different speeds can be used in a particular system.

A Typical CAN System



Courtesy of Motorola

Layers of CAN

- CAN has been divided into three layers.
  - The Object Layer
  - The Transfer Layer
  - The Physical Layer
- The Object and Transfer Layers comprise all services and functions of the *Data Link* layer defined by the ISO/OSI model.

CAN Object Layer

- The scope of the Object Layer includes:
  - Determining which messages are to be transmitted
  - Deciding which messages received by the transfer layer are to be used.
  - Providing an interface to the application layer related hardware

CAN Transfer Layer

Transfer Layer mainly deals with the following issues:

- Controlling the framing
- Performing arbitration
- Checking errors
- Signaling errors
- Fault confinement
- Determines whether or not the bus is free for a new transmission.
- Determines whether reception of a message has just started.

CAN Physical Layer

- It performs the actual transfer of bits between different nodes.
- It varies according to the requirements of individual applications.

CAN: Basic Concepts

Transmitter

- A node which originates a message is called the transmitter.
- It remains as the transmitter until the bus becomes idle or it loses arbitration.

Receiver

- A node is called the receiver if it is not sending a message, and the bus is not idle.

CAN: Basic Concepts

Messages

- Information on the bus is sent in groups of bits using a fixed format called messages.
- Any node can start transmitting a message when the bus is free.

System Flexibility

- Nodes can be added to the CAN without requiring any change in the hardware or software of any node.

CAN: Basic Concepts

Message Routing

- The content of a message is described by an identifier.
- The identifier describes the meaning of the data, so that all nodes are able to decide whether or not the data is to be acted upon by them.

Multicast

- Multiple nodes may receive and act upon the message simultaneously.

CAN: Basic Concepts

Bit Rate

- The speed of CAN may be different for different systems. But, for a given system the bit rate is fixed and uniform.

Priorities

- The identifier defines a static message priority during bus access.

CAN: Basic Concepts

Remote Data Request

- A node may request another node to send data by sending a remote data frame.

Multi-master

- When the bus is free any node may start to send a message. The highest priority node will be successful in sending the message.

CAN: Basic Concepts

Arbitration

- When the bus is free any node may start to transmit a message.
- If multiple nodes start transmitting at the same time, the bus access conflict is resolved by bit-wise arbitration using the identifier.
- If a Data frame and a Remote frame with the same identifier are initiated at the same time, the Data frame prevails over the Remote frame.

CAN: Basic Concepts

Arbitration

- During arbitration every transmitting node compares the level of the bit transmitted with the level that is monitored on the bus.
- If these levels are equal, then the node will continue to transmit.
- When a recessive level is sent, but a dominant level is monitored, the node has lost arbitration and must withdraw without sending any further bits.

CAN: Basic Concepts

Data Integrity

- In order to achieve a high level of integrity of data transfer, powerful measures of error detection, signaling and self-checking are implemented in every CAN node.

Error Detection

The following techniques are used to detect errors:

- Monitoring: Each transmitter compares the bit levels detected on the bus with the bit levels being transmitted.
- Cyclic Redundancy Check (CRC)
- Bit Stuffing
- Message Frame Check

CAN: Basic Concepts

Error Signaling

- Corrupted messages are flagged by any node detecting an error.
- Such messages are aborted and are retransmitted automatically.

Fault Confinement

- CAN nodes are able to distinguish between short disturbances and permanent failures.
- Defective nodes are switched off.

CAN: Basic Concepts

Connections

- CAN serial communication link is a bus to which a number of nodes may be connected.
- This number has no theoretical limit.
- Practically, the total number of nodes will be limited by the delay times and/or electrical loads on the bus line.

CAN: Basic Concepts

Single Channel

- The bus consists of a single bi-directional channel that carries bits.
- From this data, resynchronization information can be derived.
- The way in which this channel is implemented is not fixed, e.g. *single wire plus ground*, *two differential wire*, *optical fibers*, etc.

CAN: Basic Concepts

Bus Values

- The bus can have one of two complementary values: dominant and recessive.
- During simultaneous transmission of dominant and recessive bits, the resulting bus value will be dominant.
- For example, in the wired-AND implementation of the bus, the dominant value will be '0' and the recessive value will be '1'.

CAN: Basic Concepts

Acknowledgement

- All receivers check the consistency of the message being received.
- The receivers acknowledge a consistent message and flag an inconsistent message.

CAN: Basic Concepts

Sleep mode/wakeup

- A CAN node can be put into sleep, in which there is no internal activity and the bus drivers are disconnected.
- The sleep mode is finished with a wake-up by any bus activity or by internal conditions of the system.
- On wake-up, the internal activity is restarted.

Message Frames

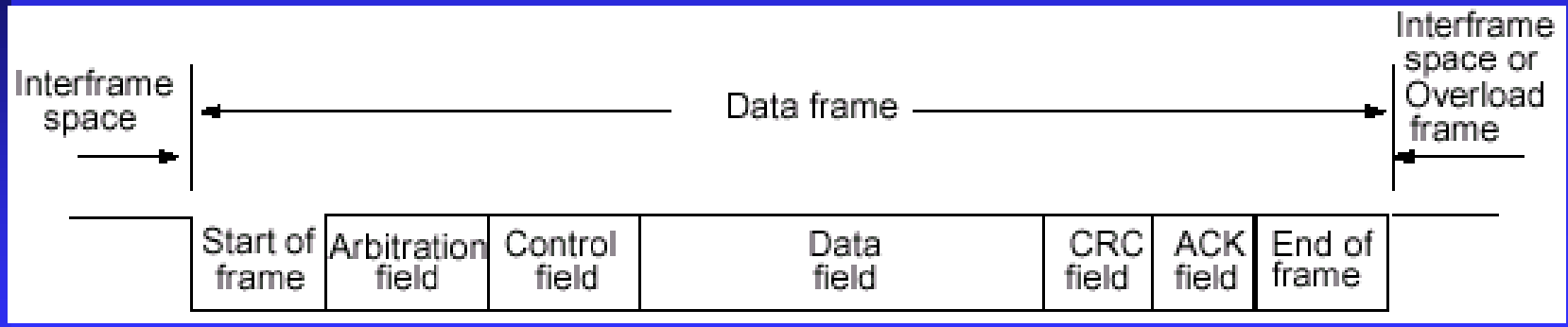
There four types of CAN message frames.

- A **Data** frame carries data from a transmitter to the receivers.
- A **Remote** frame is transmitted by a node to request the transmission of a Data frame with the same identifier.
- An **Error** frame is transmitted by any node after detecting a bus error.
- An **Overload** frame is used to provided additional delay between two data or remote frames.

Data Frame

There are seven fields in a Data frame.

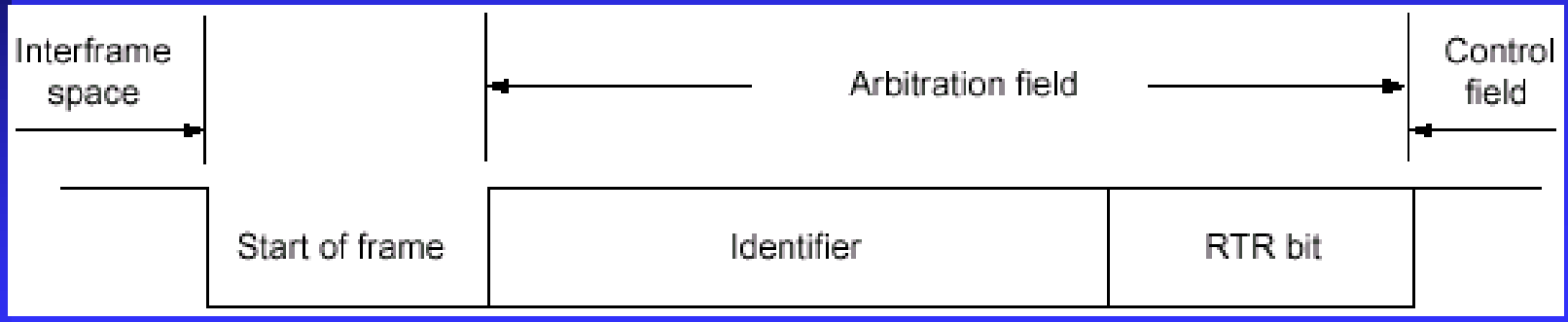
- Start of Frame: It marks the beginning of a data or remote frame. It consists of only one dominant bit. All nodes have to synchronize with the leading edge of the start of frame bit.
- Arbitration Field: This field consists of the identifier and the RTR bit.



Arbitration Field

Identifier: There are 11 bits in the identifier. These bits are transmitted in the order from ID10 to ID0. ID0 is the least significant bit. The most significant 7 bits must not be all recessive. The Identifier performs the following operations:

- Labels the content (type) of a message.
- Performs acceptance test of messages.
- Arbitrates & determines the priority of the message.



Arbitration

- **Carrier Sense, Multiple Access with Collision Detect (CSMA/CD)**
- **Method used to arbitrate and determine the priority of messages.**
- **Uses enhanced capability of non-destructive bitwise arbitration to provide collision resolution.**

Arbitration Technique

- Any potential bus conflicts are resolved by bit-wise arbitration.
- A dominant state (logic 0) has precedence over a recessive state (logic 1).

  Example: Assume that Node-1, Node-2 and Node-3 are going to compete for the bus with the following identifiers:

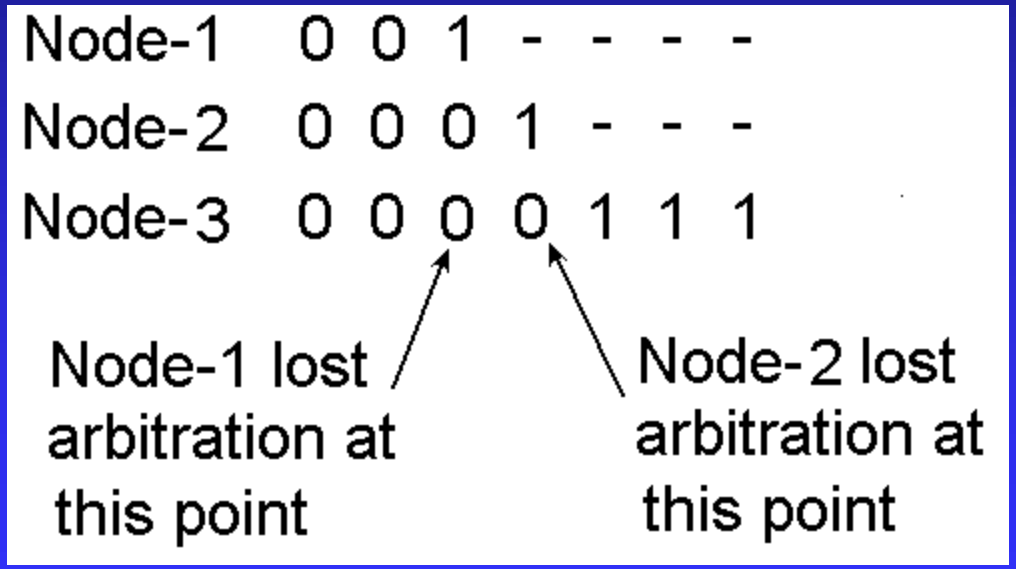
  Node-1 0010100 . .

  Node-2 0001100 . .

  Node-3 0000111 . .

Arbitration Technique (contd.)

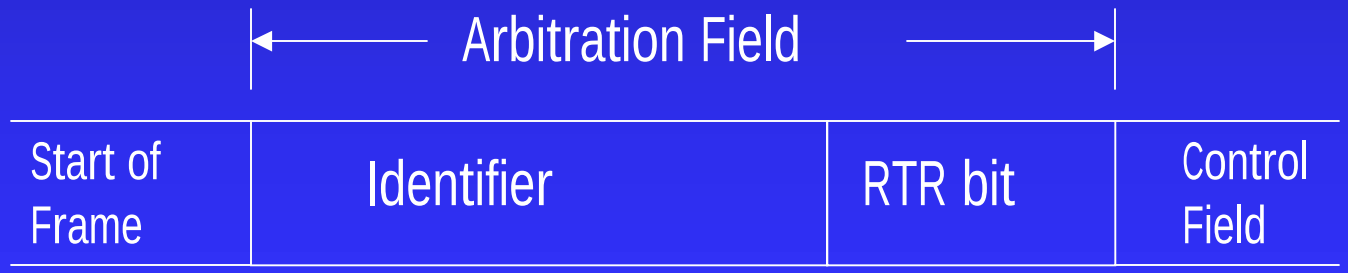
- Transmission of identifier bits by Nodes 1, 2 and 3 are as follows:



RTR Bit

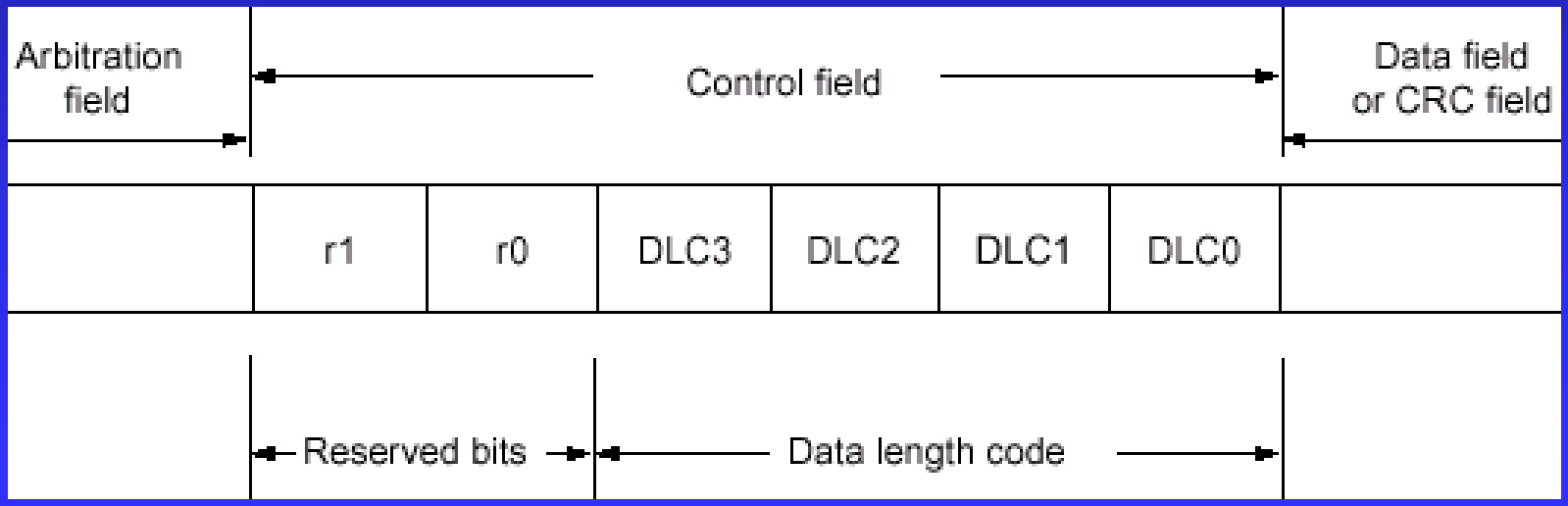
- RTR bit (Remote Transmission Request Bit): In Data frames the RTR bit must be dominant. In Remote frames it must be recessive.

RTR bit	Type of Frame
d	Data Frame
r	Remote Frame



Control Field

- There are 6 bits in the control field.
- Out of the 6 bits, 4 bits are used to indicate the length of the data and the other two bits are reserved for future expansion.
- The reserved bits must be sent as dominant.



Data Length Code

- The number of bytes in the data is indicated by this field.
- The DLC bits can code data lengths from 0 to 8 bytes; other values are not allowed.

d="dominant"

r="recessive"

Data Length Code				Length
d	d	d	d	0
d	d	d	r	1
d	d	r	d	2
d	d	r	r	3
d	r	d	d	4
d	r	d	r	5
d	r	r	d	6
d	r	r	r	7
r	d	d	d	8

Data and CRC Fields

Data Field

- The data field contains the data to be transmitted.
- The data field can contain 0 to 8 bytes.
- The most significant bit of a byte is sent first.

CRC Field

- The CRC field contains the CRC sequence followed by a CRC delimiter.
- The CRC sequence is calculated using a predefined algorithm. The CRC delimiter is a single recessive bit.

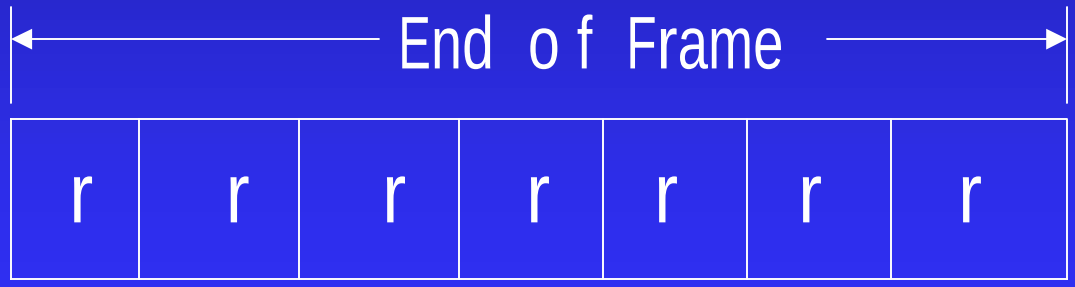
Acknowledge Field

ACK Field

- The ACK field consists of two bits.
- One bit for ACK slot and another bit for ACK delimiter.
- The TRANSMITTING node sends two recessive bits through the ACK field.
- The RECEIVING node sends a dominant bit through the ACK slot after receiving a valid message.
- Since the CRC and ACK delimiters are both recessive bits, the ACK slot is always surrounded by two recessive bits.

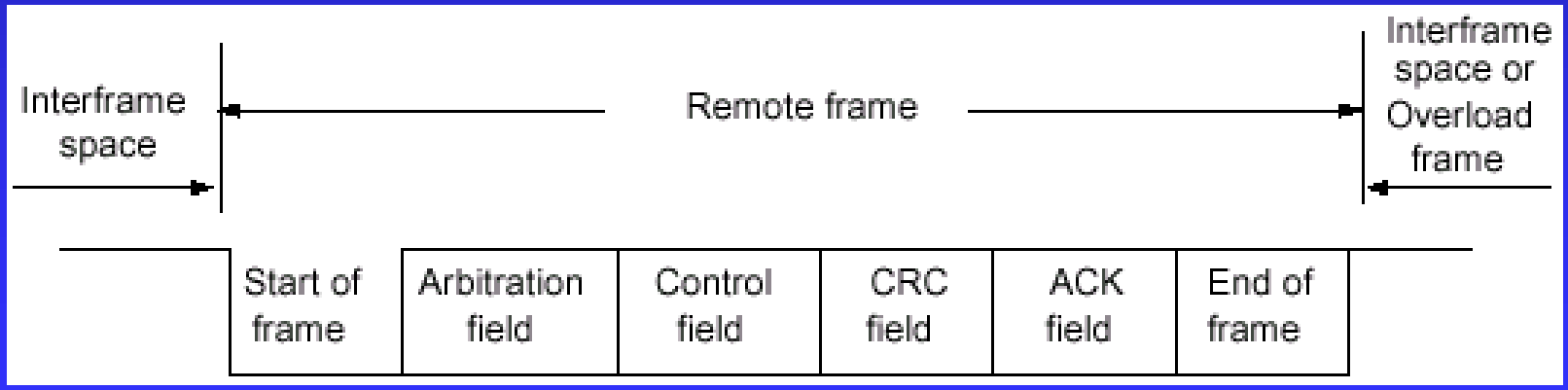
End of Frame

- Each data frame and remote frame is delimited by a flag sequence consisting of seven recessive bits.
- This is a contradiction to the Bit Stuffing technique.



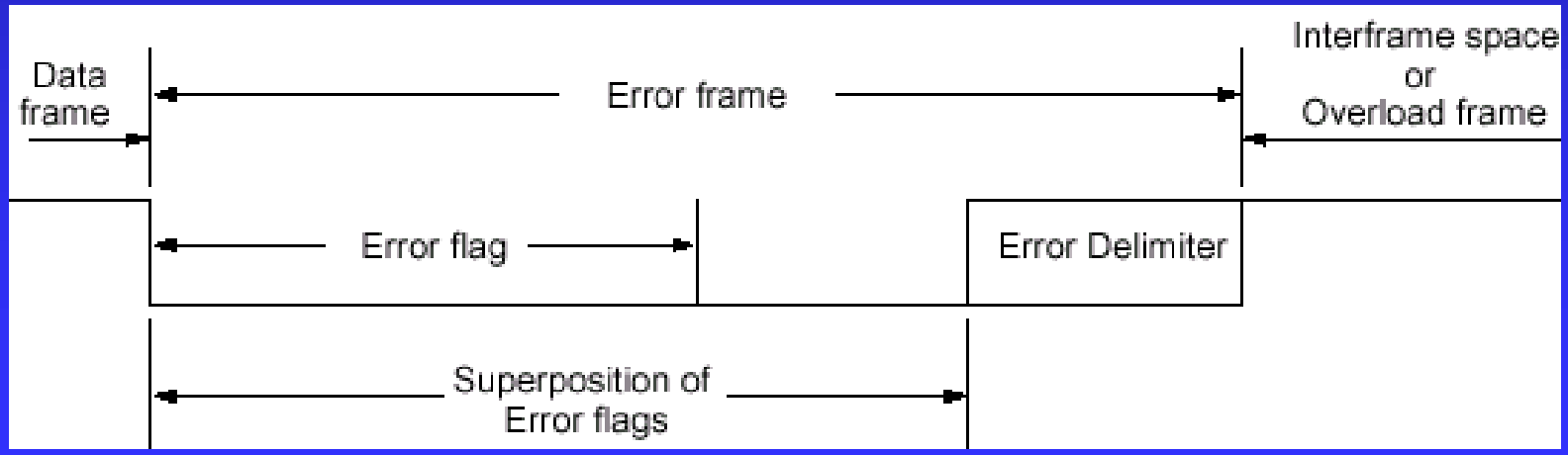
Remote Frame

- A node which needs data from another node (remote node) can request the remote node to transmit data by sending a remote frame.
- It has six fields: Start of frame, Arbitration, Control, CRC, ACK and End of frame.
- It has no Data field, regardless of the value of the Data length code.



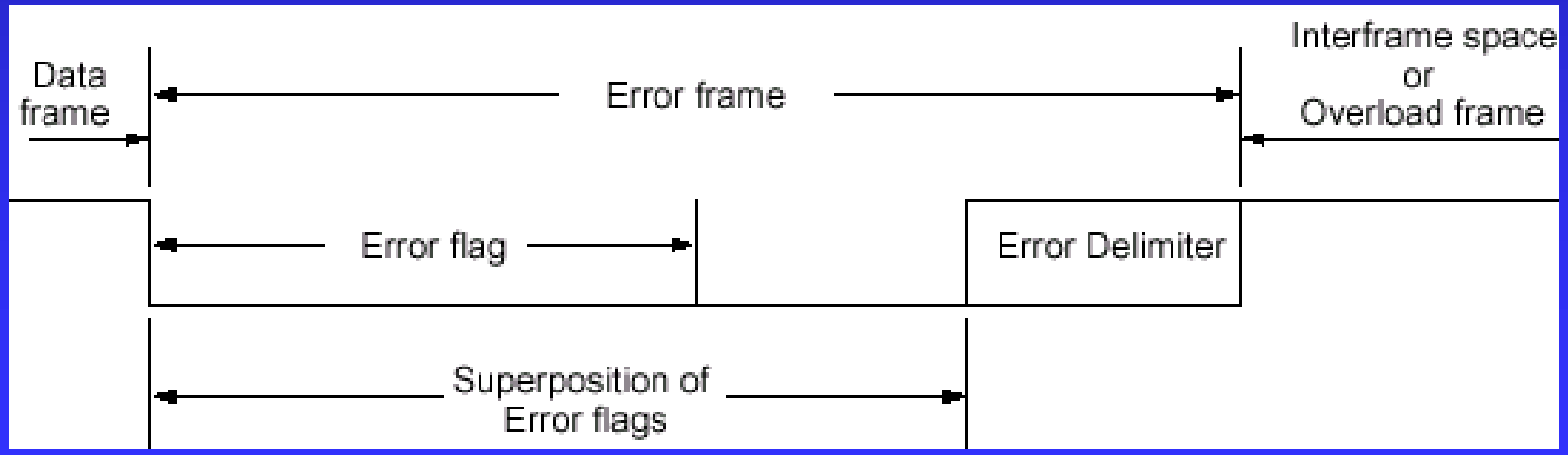
Error Frame

- The Error frame contains two distinct fields.
- The first field is given by the superposition of Error flags contributed from different nodes.
- The second field is the Error delimiter



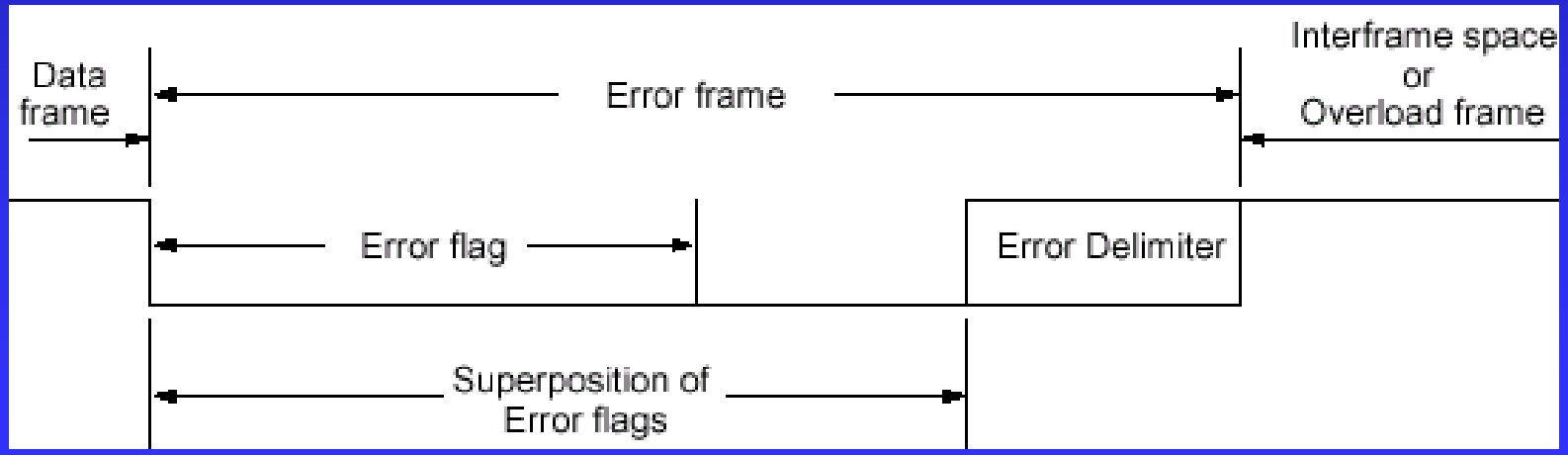
Error Flag

- There are two types of error flags: an Active error flag and a Passive error flag.
  - Active error flag consists of six consecutive dominant bits (d d d d d d).
  - Passive error flag consists of six consecutive recessive bits (r r r r r r) unless it is overwritten by dominant bits from other nodes.



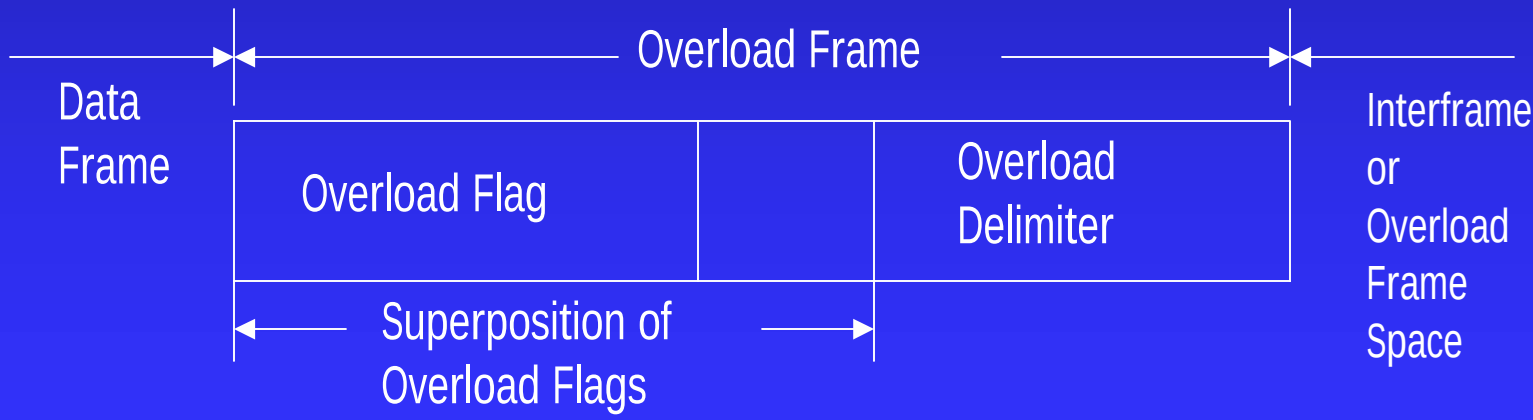
Error Delimiter

- The Error delimiter consists of *eight* recessive bits (r r r r r r r r).
- After transmission of an Error Flag each node sends recessive bits and monitors the bus until it detects a recessive bit. Afterwards it starts transmitting seven more recessive bits.



Overload Frame

- The Overload frame contains two bit fields, Overload flag and Overload delimiter.
- There are two types of Overload conditions, both of which lead to the transmission of an Overload flag:
  - Where the internal conditions of a receiver are such that the receiver requires a delay for the next Data frame or Remote frame.
  - On detection of a dominant bit during INTERMISSION.



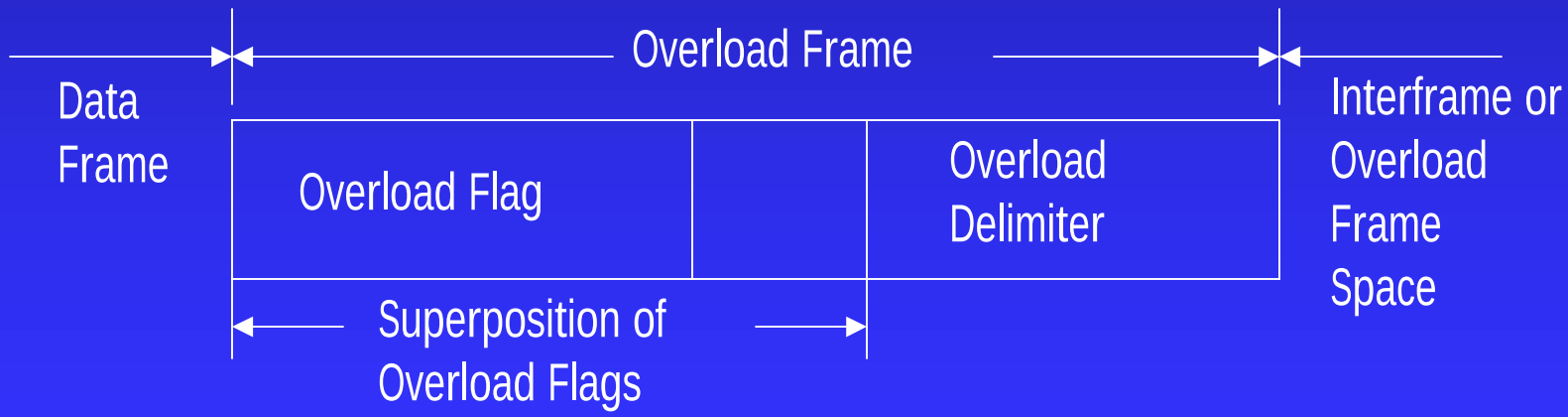
Overload Flag and Overload Delimiter

Overload Flag

- This flag consists of six dominant bits (d d d d d d). It is similar to that of an ACTIVE error flag.

Overload Delimiter

- The Overload delimiter consists of eight recessive bits (r r r r r r r r).

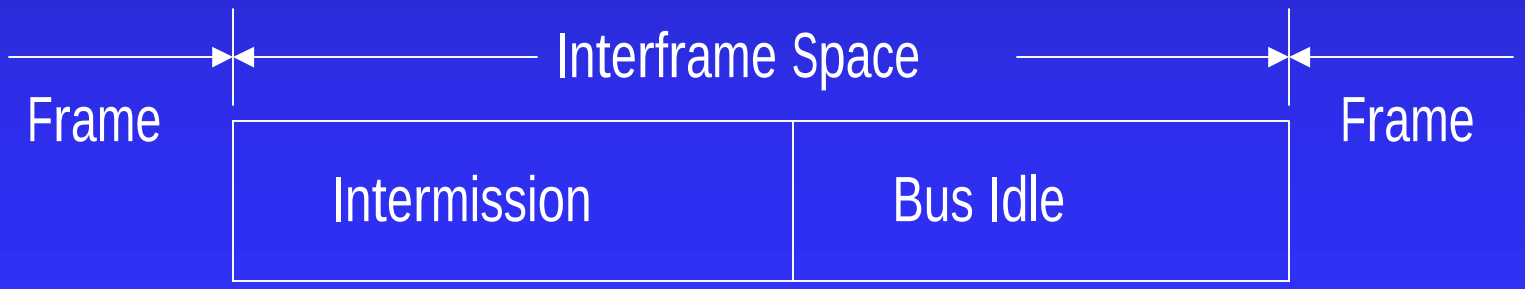


Inter-frame Space

- Data frames and Remote frames are separated from preceding frames by a field called Inter-frame space

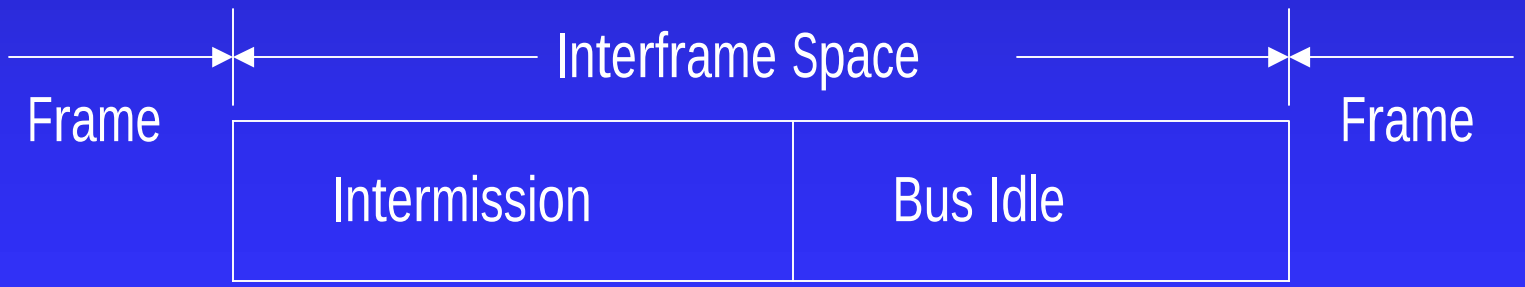
Intermission

- Intermission consists of three recessive bits (r r r).
- During Intermission no node is allowed to start transmission of a Data frame or Remote frame. The only action permitted is signaling of an Overload condition.



Bus Idle

- The bus idle period may be of arbitrary length.
- The bus is recognized to be free, and any node having something to transmit can start transmission.
- A message, pending during the transmission of another message, is started in the first bit following the Intermission.
- The detection of a dominant bit on the bus is considered as the Start of a Frame.



Bit-stream coding

- The frame segments: Start of frame, Arbitration field, Control field, Data field and CRC Sequence are coded by the method of bit stuffing.
- Whenever a transmitter detects five consecutive bits of identical value in the bit-stream to be transmitted, it automatically inserts a complementary bit in the actual transmitted bit-stream.

  Example:

  Original Bit Sequence: 0000000011111111

  Transmitted Bit Seq.: 0000010011111010111

Error Detection

CAN implements five error detection mechanisms.

- Three at the message level
  - Cyclic Redundancy Checks (CRC)
  - Frame Checks
  - Acknowledgment Error Checks
- Two at the bit level
  - Bit Error
  - Stuff Error

CRC Error

- The CRC sequence consists of the result of the CRC calculation by the transmitter.
- The receivers calculate the CRC in the same way as the transmitter. A CRC error is detected if the calculated result is not the same as that received in the CRC sequence.

Frame Error (Form or Format Error)

- If a receiver detects an invalid bit in one of the following positions, a Form Error (or Format Error) is flagged:
  - CRC Delimiter
  - ACK Delimiter
  - End of Frame Bit Field
  - Interframe Space (the 3 bit INTermission field and a possible Bus Idle time).

Acknowledge Error

- Each receiving node writes a dominant bit into the ACK slot
- If a transmitter determines that a message has not been ACKnowledged then an ACK Error is flagged.
- ACK errors may occur because of transmission errors (bits have been corrupted) or there is no operational receivers.

Bit Error

- A node which is sending a bit on the bus also monitors the bus.
- The node must detect, and interpret as a Bit error, the situation where the bit value monitored is different from the bit value being sent.
- An exception to this is the sending of recessive bit during the Arbitration field or during the ACK slot; in this case no Bit error occurs when a dominant bit is monitored.

Stuff Error

- Bit stuffing is used to guarantee enough edges in the NRZ bit stream to maintain synchronization.
- After five identical and consecutive bit levels have been transmitted, the transmitter will automatically inject (stuff) a bit of the opposite polarity into the bit stream.
- Receivers of the message will automatically delete (destuff) such bits.
- If any node detects six consecutive bits of the same level, a stuff error is flagged.

Error Signaling

- A node detecting an error condition signals this by transmitting an Error flag. An error-active node will transmit an ACTIVE error flag; an error-passive node will transmit a PASSIVE error flag.
- Whenever a Bit error is detected by any node, that node will start transmission of an Error flag at the next bit time.
- Whenever a CRC error is detected, transmission of an Error flag will start at the bit following the ACK delimiter, unless an Error flag for another error condition has already been started.

Fault Confinement

- A method for discriminating between temporary errors and permanent failures .

- Temporary errors may be caused by external conditions, voltage spikes, etc.

- Permanent failures may be caused by bad connections, faulty cables, defective transmitters or receivers, or long lasting external disturbances.

Error counts

- To facilitate fault confinement two counts are implemented in every bus node:
  - Transmit Error Count
  - Receive Error Count

- These counts are modified according to the following 12 rules (more than one rule may apply during a given message transfer)
  1. When a Receiver detects an error, the Receive Error Count is incremented by 1, except when the detected error was a Bit error during the sending of an Active error flag or an Overload.

Error Counting Rules

2. When a Receiver detects a dominant bit as the first bit after sending an error flag, the Receive Error Count is incremented by 8.
3. When a Transmitter sends an error flag, the Transmit Error Count is increased by 8. (But there are two exceptions to this rule).
4. An error-active Transmitter detects a Bit error while sending an Active error flag or an Overload flag, the Transmit Error Count is increased by 8.
5. An error-active Receiver detects a Bit error while sending an Active error flag or an Overload flag, the Receive Error Count is increased by 8.

etc.

CAN Node Status

- With respect to error confinement, a node may be in one of three states: Error-Active, Error-Passive, or Bus-Off.

Node Status	Value of either Transmit or Receive Error Count
Error-Active	1 to 127
Error-Passive	128 to 255
Bus-Off	> 255

Error-Passive → Error-Active → Normal

- Transmit Error Count is decremented by 1 after a successful transmission of a message.
- Receive Error Count is decremented by 1after a successful reception of a message.
- If both Transmit and Receive Error Counts go below 128, then an Error-Passive node becomes an Error-Active node
- If both Transmit and Receive Error Counts become 0, then an Error-Active node becomes a Normal node.

Activities of Error Nodes

- An error active node can normally take part in bus communication and send an ACTIVE error flag when an error has been detected.
- An error-passive node must not send an ACTIVE error flag. It takes part in bus communication, but when an error has been detected only a PASSIVE error flag is sent. Also after a transmission, an error-passive node will wait before initiating a further transmission.
- A bus-off node is not allowed to have any influence on the bus.

Requirements of a CAN Controller

- Simple user interface to the CPU
- Message filtering and buffering
- Protocol handling
- Physical layer interface

Full CAN versus Basic CAN

Full CAN (suitable for Powertrain)

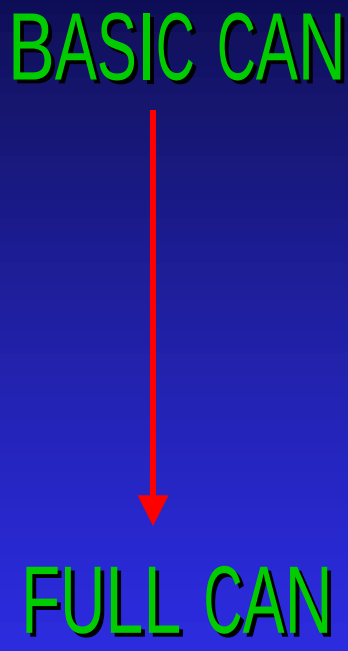
- Typically 16 message buffers, sometimes more.
- Global and Dedicated Message Filtering Masks.
- Dedicated Hardware for Reducing CPU workload.
- More Silicon → More Cost

Basic CAN (suitable for Car body)

- 1 or 2 Transmit and Receive Buffers
- Minimal Filtering
- More Software Intervention.
- Low Cost

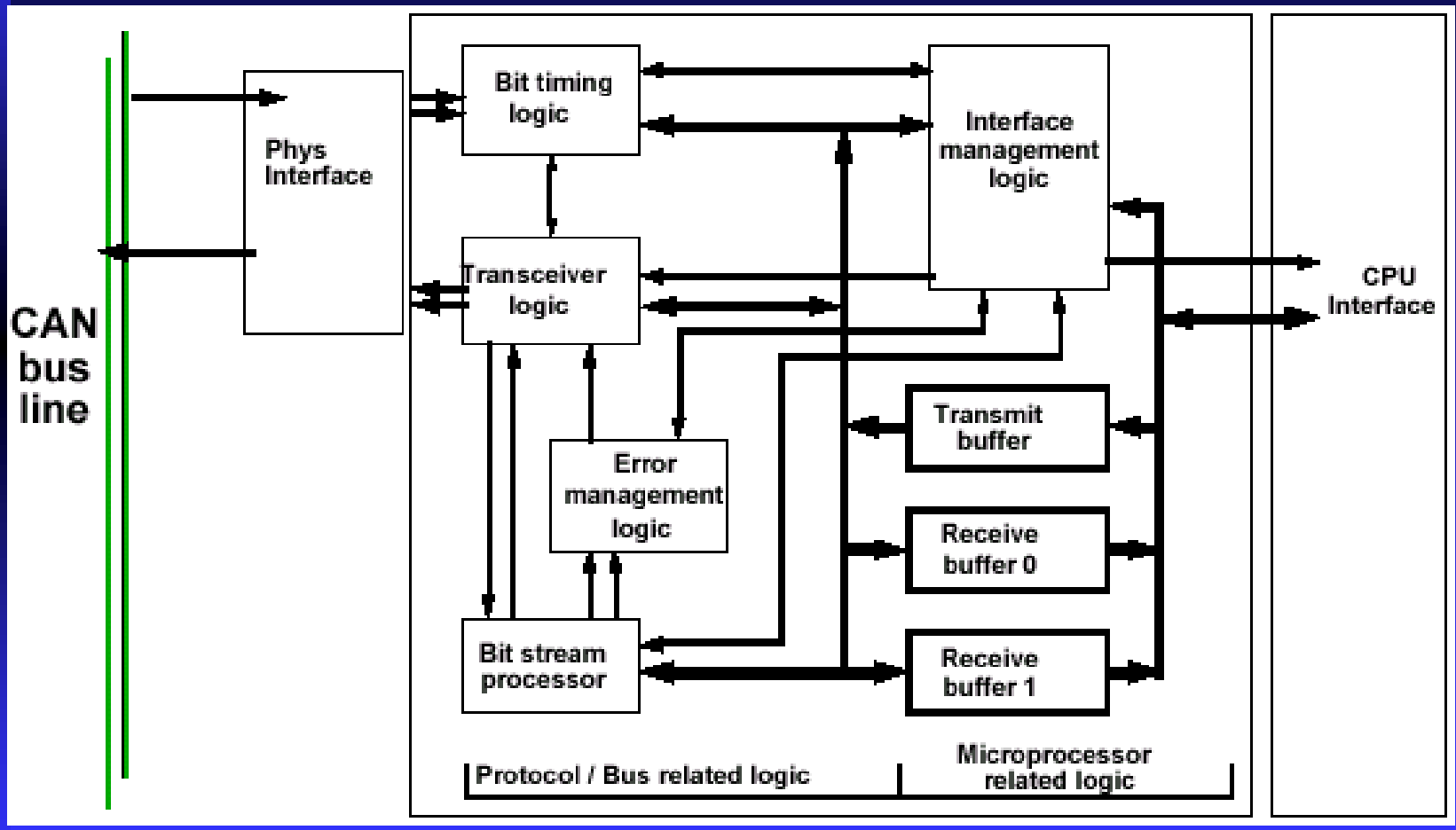
Motorola CAN Implementations

- MCAN (on HC05X family)
- MSCAN08
- MSCAN12
- TouCAN



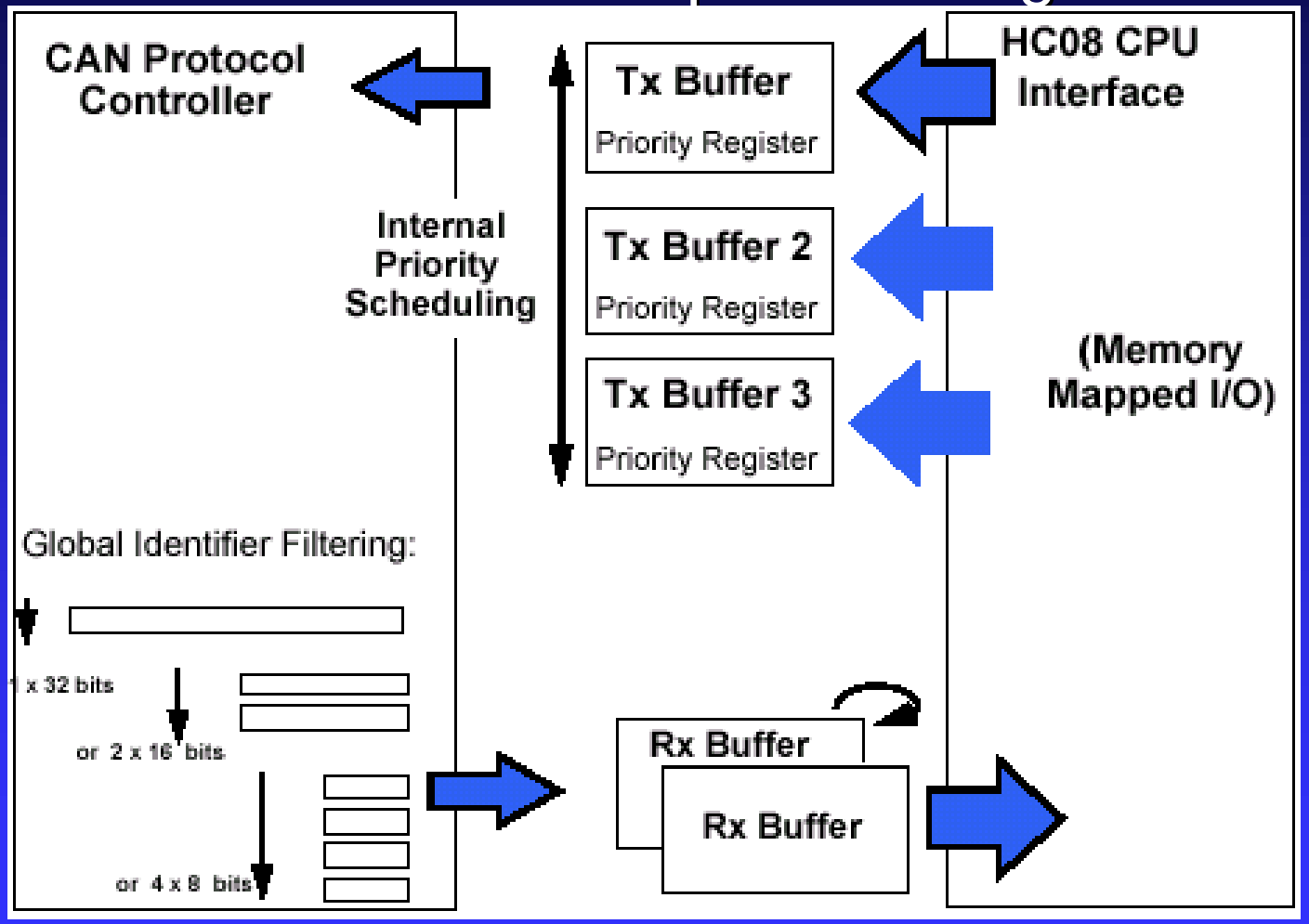
Motorola CAN (MCAN) Module HC05X Family

- 2 RX and 1 Tx buffers



MSCAN HC08 Family

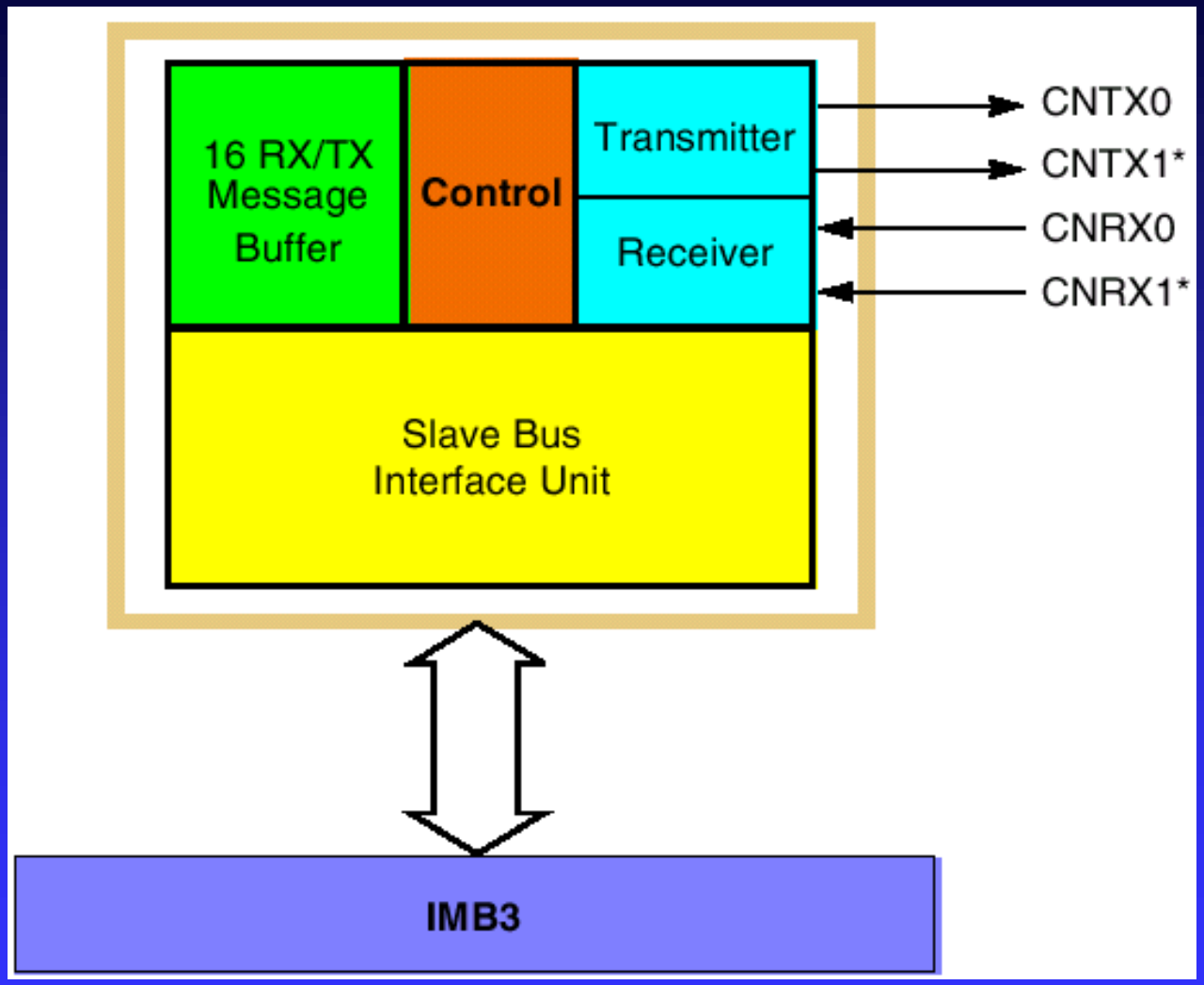
- 3 Tx and 2 Rx buffers plus filtering scheme.



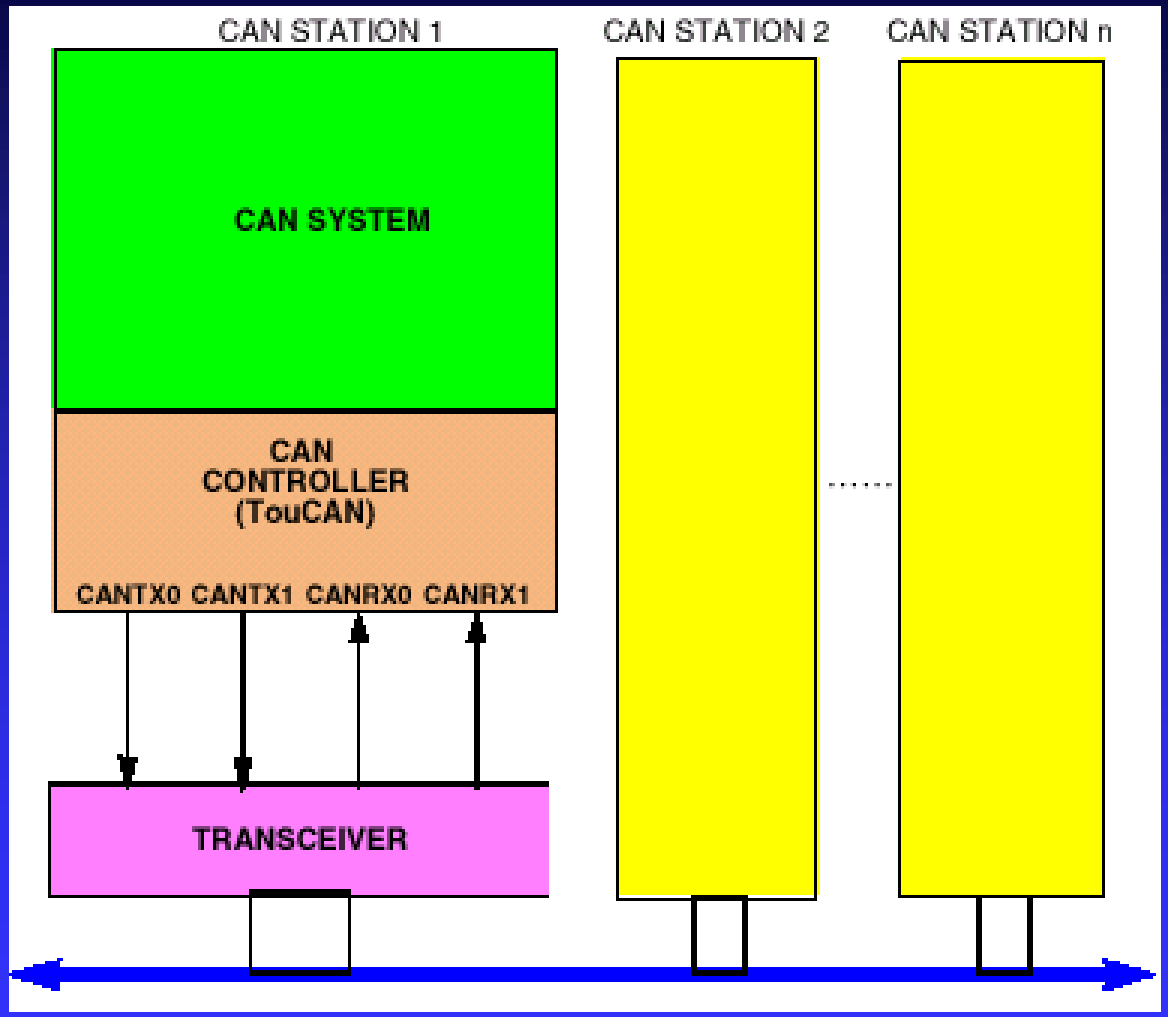
The High Performance TouCAN Module

- Full implementation of the CAN protocol - Version 2.0B *(29-bit identifier)*
- 16 Rx/Tx Message Buffers of up to 8 bytes Data Length
- Programmable Bit Rate up to 1 Mbit/sec
- Programmable Receive identifier mask
- Time stamping to allow network timing synchronization
- Low power “sleep” mode, with programmable “wake up”

TouCAN Module of MC68377



Typical CAN Network



Other Microcontrollers Supporting CAN

INTEL : 82527

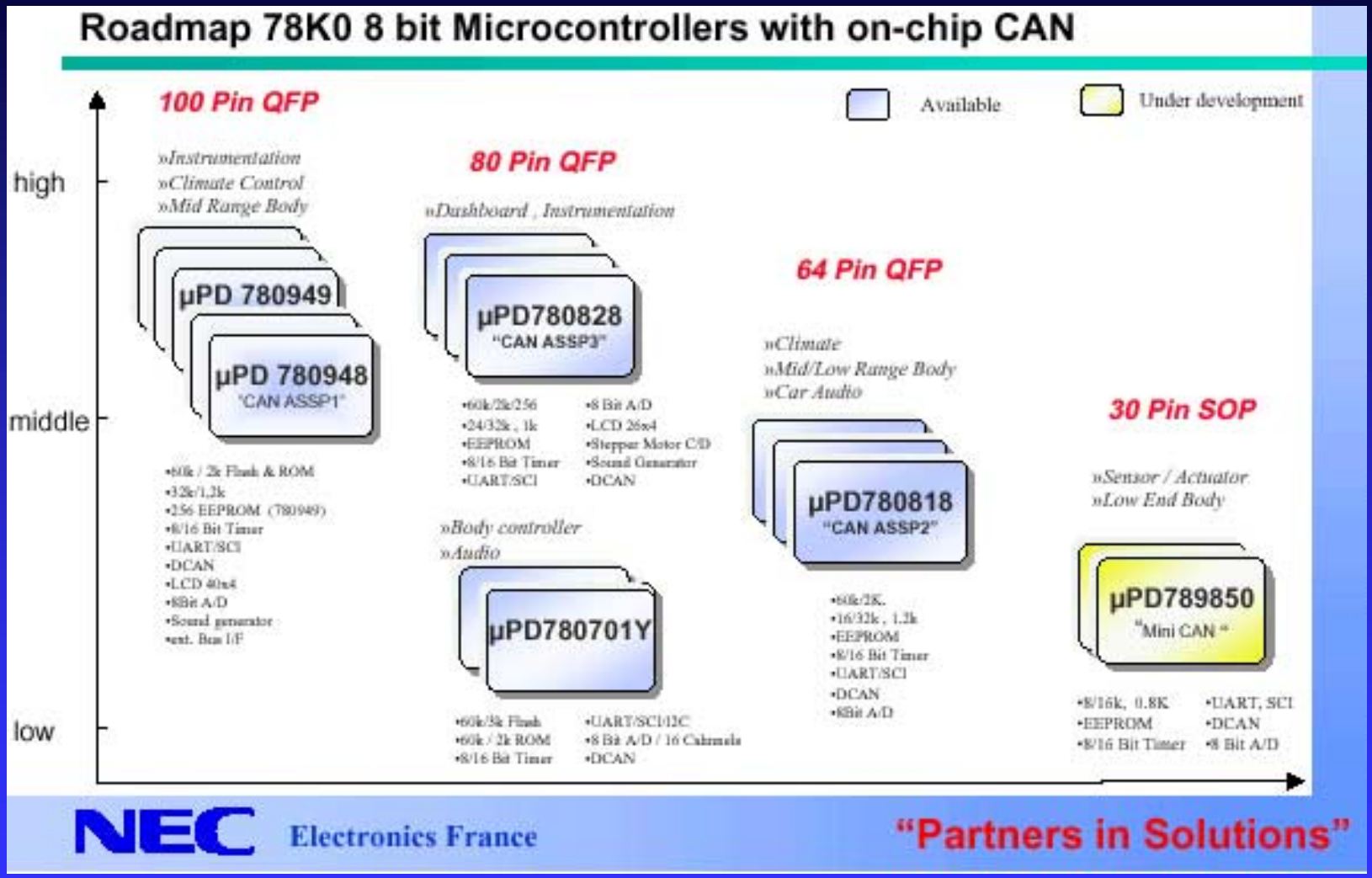
MICROCHIP : PIC 18 Family

http://www.microchip.com

ATMEL : T89C51CC01 CAN

http://www.atmel.com/

NEC Microcontrollers with CAN



CAN Tools & Products Manufacturers

http://www.vector-cantech.com/

Product	www.vector-cantech.com
CANalyzer	CAN network analysis and development tool.
CANoe	CAN system level message analysis and modeling for multiple modules.
CANape Graph	ECU monitor and calibration tool using CAN and CCP.
CANscope	Digitized oscilloscope of CAN message wave forms.

CAN Tools & Products Manufacturers

http://www.axiomatic.com/default.html

CANbus Hub Technology

- **Protocol exchanger** allows connection of three CANbuses with different protocols (i.e. CANOpen, SAEJ1939 and DeviceNet).
- **Applications** - interface to SAE J1939 engine control modules and gear boxes to other CAN buses in the vehicle (off-highway equipment, on-highway equipment, and agricultural equipment) or acts as a gateway in industrial automation applications

CAN Tools & Products Manufacturers

- ADVANCED VEHICLE TECHNOLOGIES, INC. http://www.avt-hq.com

- HIGHLANDER TECHNOLOGIES

  http://www.highlandertech.com/

Network Sizes

- The number of nodes that can exist on a single network is, theoretically, limited only by the number of available identifiers.
- However, the drive capabilities of devices imposes restrictions.
- Depending on the device types, up to 32 or 64 nodes per network is normal.

Data Rate vs Bus Length

- The bit rate depends on the total overall length of the bus and the delays associated with the transceivers.
- For all ISO11898 compliant devices running at 1Mbit/sec speed, the maximum bus length is specified as 40 Metres,
- For longer bus lengths it is necessary to reduce the bit rate.
- 500 K bits per second at 100 metres (328 ft)
- 250 K bits per second at 200 metres (656 ft)
- 125 K bits per second at 500 metres (1640 ft)

The above numbers are taken from M J Schofield's web site
http://www.mjschofield.com/implment.htm

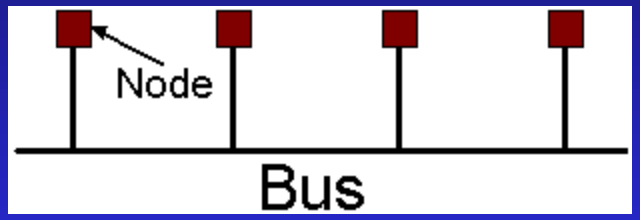
VAN : Vehicle Area Network

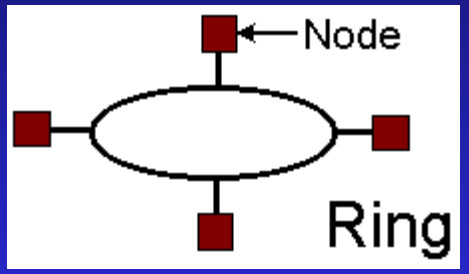
- An Overview of VAN
- Transfer modes
- Transfer features
- Coding
- Transmit/Receipt Error Management

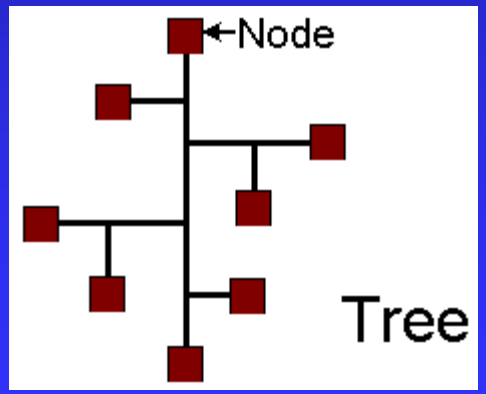
An Overview of VAN

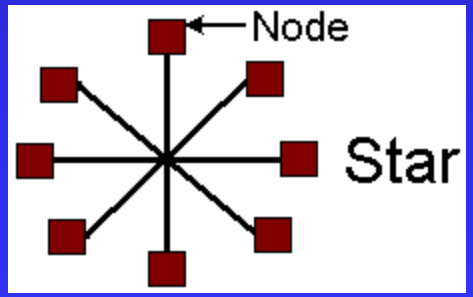
Topology

- Bus, ring, tree and star type topology









An Overview of VAN

Protocol Level

- Multi-masters, multi-slaves and mixed architectures.
- Communication type : broadcast or point to point with acknowledge.
- Access method based on a non-destructive bitwise arbitration on the whole frame.

An Overview of VAN

The VAN protocol in a few words

- In-Frame response (In-frame reply) mechanism.
- In-Frame acknowledge mechanism.
- 12 bits message Identification.
- Variable size of the data field, up to 28 bytes.
- Logical or physical addressing.
- Bit rate up to 1 Mbit/s.
- Systematic encoding method, patented and named E-Manchester (Enhanced Manchester).

An Overview of VAN

- Detection of possible transmission errors using a 15 bits CRC code (Cyclic Redundancy Code).
- A VAN module is able to have the following behaviors :
  - Autonomous module : it is able to send data or orders on the network, to receive data and to ask other modules.
  - Slave module : it is able to receive or send data when it is asked.
  - Synchronous access module : it is able to initiate a transmission when a frame header has been detected on the network.

An Overview of VAN

Particular features

- A symbol named preamble located at the start of the frame allows on the one hand to indicate the beginning of the frame, and on the other hand to set a temporal reference.
- Non destructive collision access, in-frame response and in-frame acknowledge are possible.
- The medium is typically composed of two data transmission lines similar to CAN.

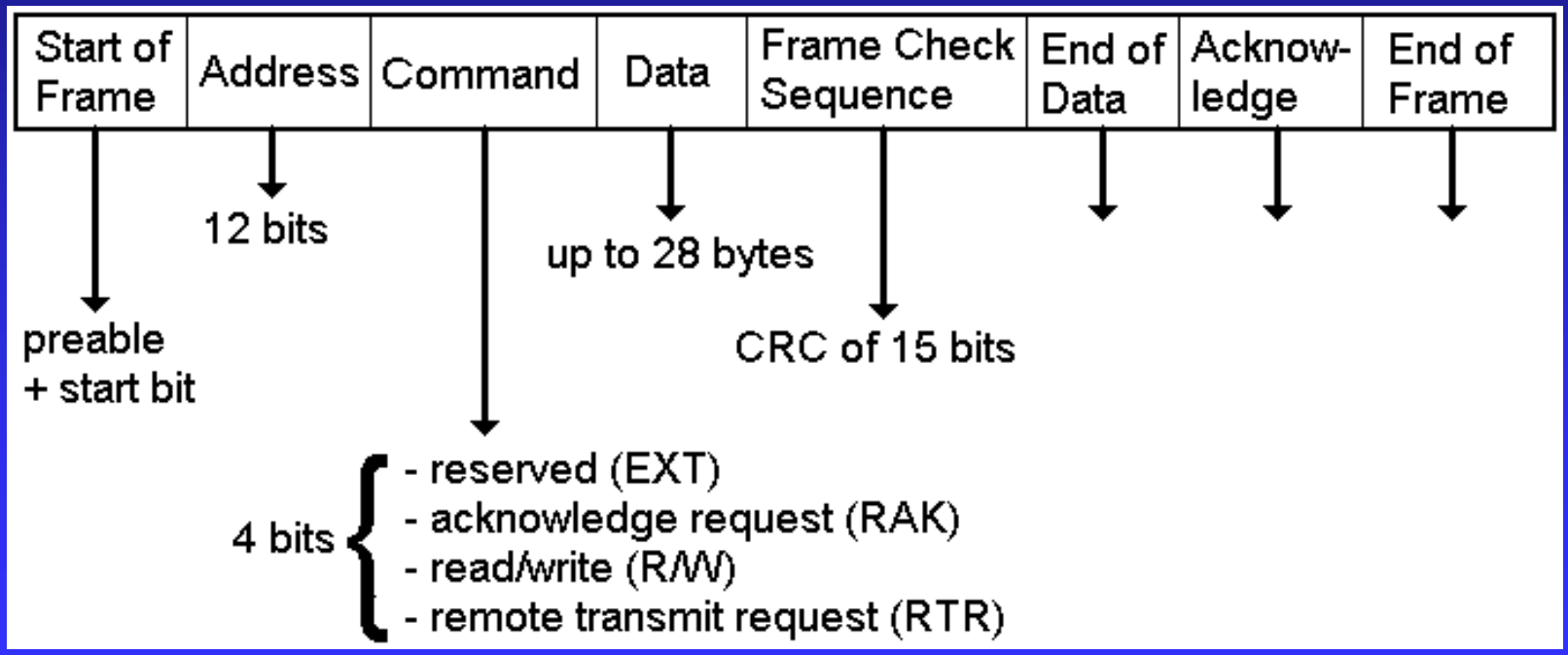
An Overview of VAN

Particular features

- The line diagnosis includes detecting a communication medium failure (short circuit and open circuit).
- An automatic transmission retry on error.
- The identification of communication errors given to the upper layers.
- An activity supervisor included in the slave components (Watchdog).

An Overview of VAN

VAN frame structure



VAN Transfer Modes

- VAN data transfer modes are based on the bus activity monitoring, which is the fundamental rule of the CSMA protocols (Carrier Sense Multiple Access).
- The basic rule to apply is :
  - a frame cannot be sent as long as the medium is not free for a certain duration.
  - The minimal length of idle bus is named Inter Frame Separation (IFS), which is equal to 128 local clock periods (8 Time Slot).

VAN Transfer Modes

The different transfer types are :

- Read or request data,
- Sending data to a single node (Point to point mode),
- Sending data without any specified destination (Broadcast mode),
- Asking data, implying an immediate (In Frame Response) or later (deferred response) response.

VAN Transfer Modes

There are two types of data consumers :

- Only one node pointed by the IDEN field (IDENtifier) : this mode is named point to point.It is the most important level of transmission security because the in-frame acknowledge is possible and advisable.

- One or several nodes : this mode is named broadcast. In this case, the in-frame acknowledge is not used because it has no meaning.

VAN Transfer Modes

Transfers are based on the following communication access principles :

- To be a frame initiator,
- To answer in a frame,
- To answer with a deferred reply,
- To transmit synchronously with the frame header sent by another node,
- To acknowledge a frame.

VAN Transfer Modes

Communication behaviors

- The VAN protocol defines three types of communication behaviors:
  - master type,
  - slave type, and
  - synchronous type behavior

Master type behavior

- A master type module includes all communication privileges allowed by the VAN protocol.
- It is able to achieve the following operations:
  - To initiate a frame as a data producer or a data consumer. This is called **Rank 0** access because it is triggered from the frame symbol number 0 which is the preamble.
  - To receive incoming data.
  - To acquire broadcasted data.

Slave type behavior

- A slave produces the requested data if asked by a master type module.
- This type of node is able to achieve the following operations :
  - To be a data producer, using the in-frame response mechanism. This is called **Rank 16** access because it is triggered from the frame symbol number 16 which is the RTR bit (Remote Transmit Request),
  - To receive incoming data,
  - To acquire the broadcasted data.

Synchronous type behavior

- A synchronous type module is similar to a slave module.
- The only difference is that this one can be producer of data without having been requested to.
- On the other hand, there is a restriction to this privilege : this type of node is only able to start the communication if a frame is present on the network.

Synchronous type behavior

- The synchronous transmission begins from the start bit.
- This type transmission is also known as **Rank 1** access because it starts from the frame symbol number 1.
- The main benefit is to allow a node, with a slave behavior, to produce data having priority associated with an internal event.

VAN Transfer Features

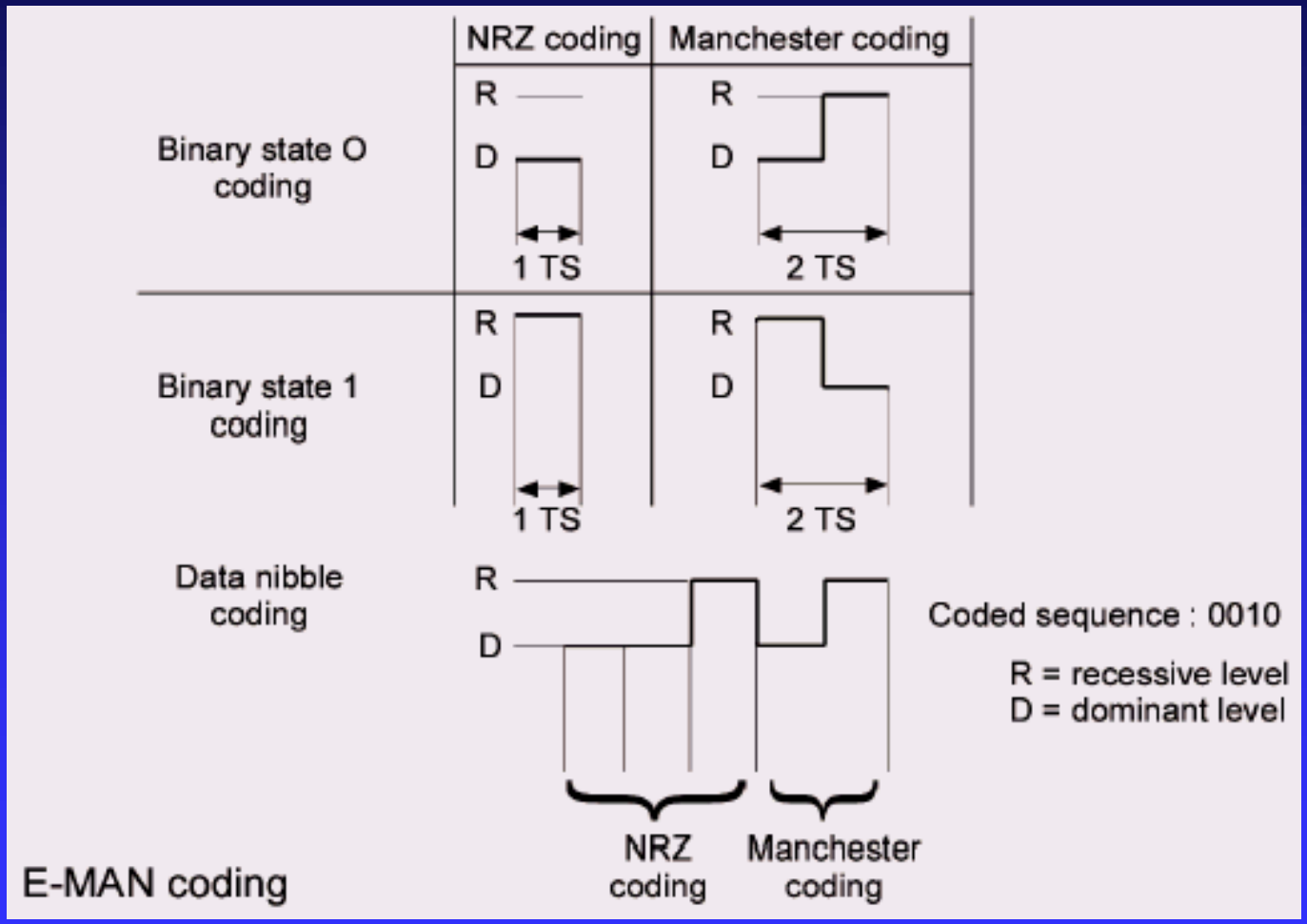
Arbitration Technique

- Same as CAN
- In the case of the VAN protocol, the arbitration process continues with the whole frame whereas for CAN, the arbitration is only allowed on the identification field.
- For CAN, if a collision occurs after the identification field, it will be considered as a transmit error.

VAN Coding

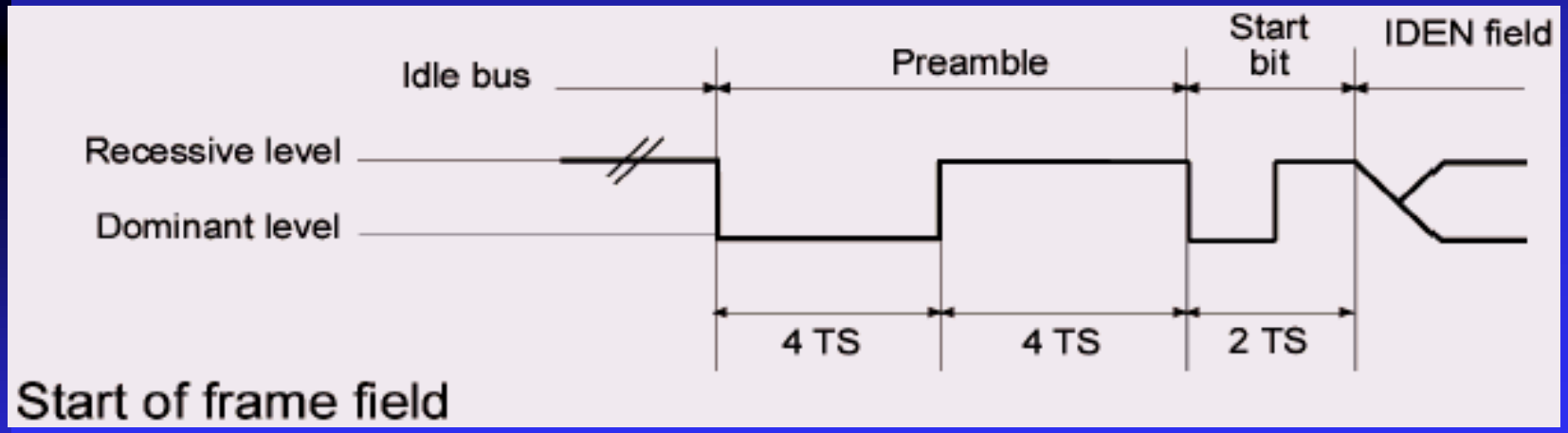
- In the VAN protocol, a block coding named E-Manchester has been adopted.
- Bits are grouped as blocks of four.
- NRZ coding is used for the first three bits, and E- Manchester coding is used for the fourth bit.

E- Manchester Coding



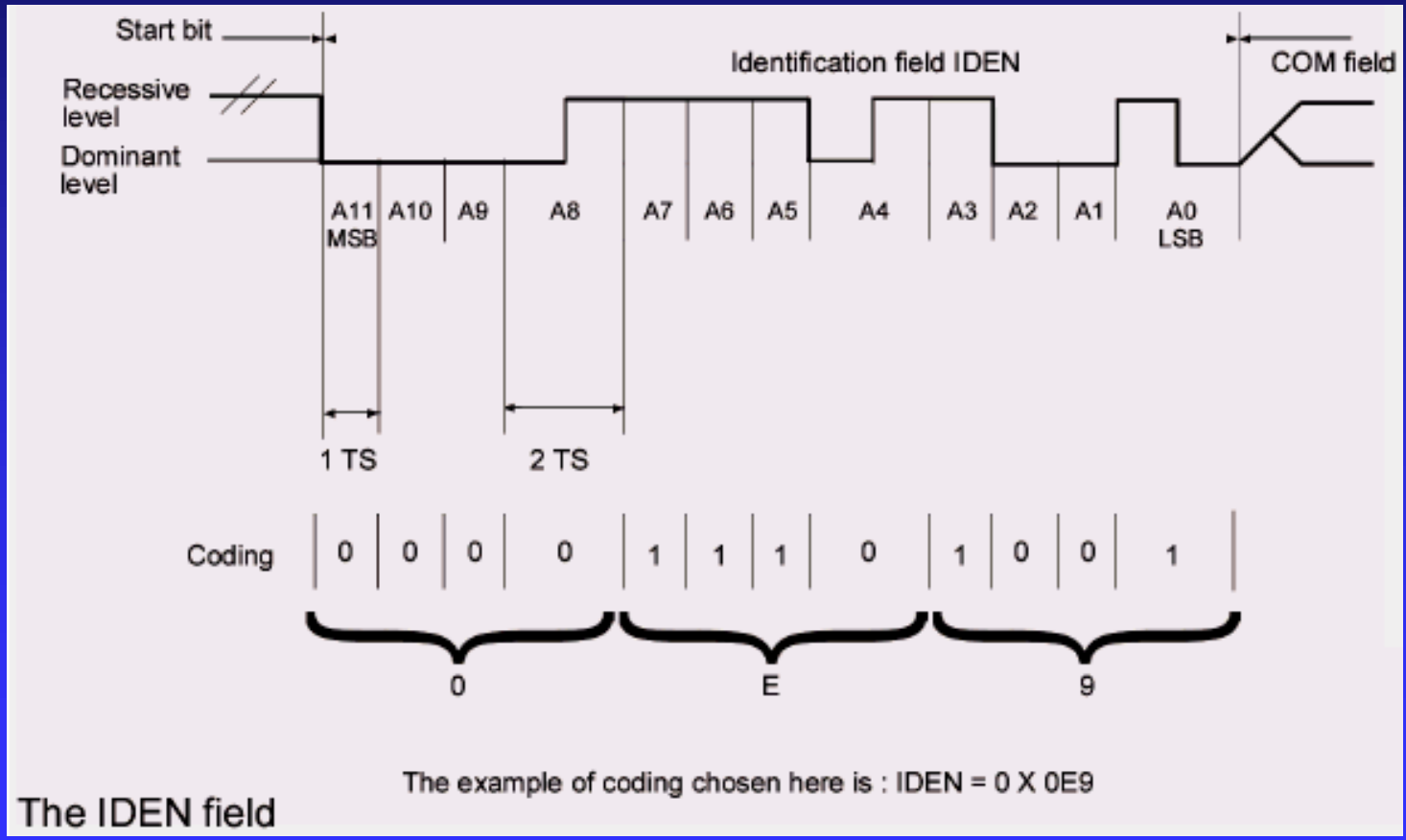
Start of Frame Field

This field is composed of two symbols: the preamble and the start bit.

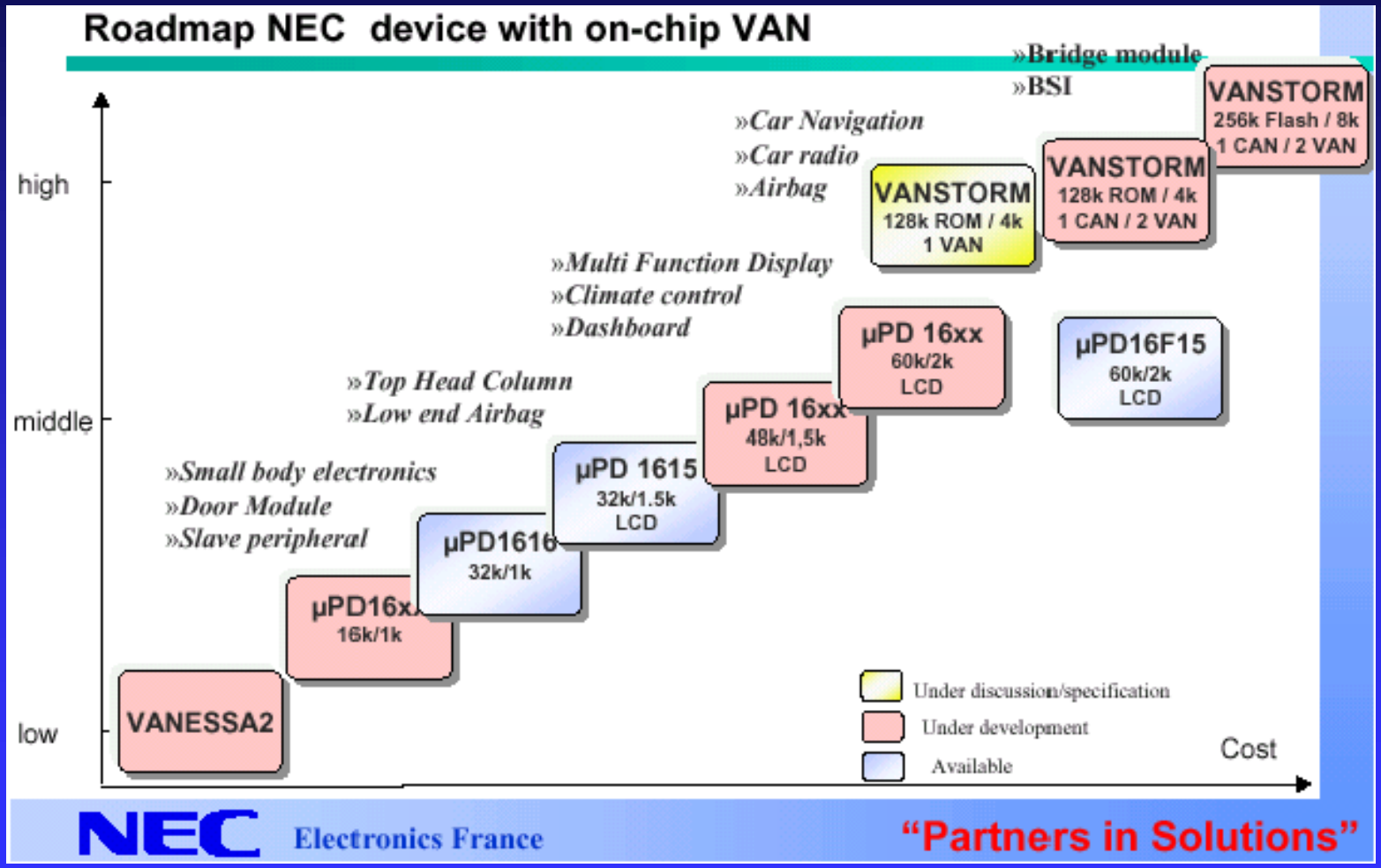


Identifier Field

- **This field contains 12 bits. The most significant bit immediately follows the start bit.**



Devices with on-chip VAN



VAN versus CAN

	VAN	CAN
Identification field	12 bits (possible extension in the data field)	11 bits in standard mode, 29 bits in extended mode
Frame length	28 bytes	8 bytes
Acknowledge	Selective	Non selective
In Frame Response	Yes	No
Bit Coding	Enhanced Manchester or Pulsed	NRZ bit stuffing
Other frames	No	Overload frame and Error frame

VAN versus CAN

	VAN	CAN
Standardization	Yes ISO 11519-3	Yes ISO 11519-2 (low speed) and 11898 (high speed)
Medium access	CSMA/CA	CSMA/CA
Bitwise arbitration	Yes (on the whole frame)	Yes (on Identification field+ Com field)
Hierarchical access	Rank 0 (autonomous), Rank 1 (synchronous), Rank 16 (slave)	Rank 0
Bit rate	standard 0 to 250 KTS/s (limitation of the line transceiver, protocol controllers can reach 1MTS/s)	Low speed standard 0 to 125 Kbits/s, High speed standard 0 to 1Mbits/s

LIN: Local Interconnection Network

Features of LIN

- Easy to Use
- Components available today
- Cheaper than CAN, VAN and J1850

Benefits of LIN

- More Reliable Cars (Diagnostics)
- More Functionality at Lower Price
- Standardization of Interfaces and Components
- Functional Extendibility

Typical LIN Applications

- **Roof**
  - Rain Sensor
  - Light Sensor
  - Light Control
  - Sun Roof
- **Seat**
  - Seat Position Motors
  - Occupancy Sensor

Typical LIN Applications

- **Steering Wheel**
  - Cruise Control
  - Wiper
  - Turning Light
  - Radio
  - Phone

Typical LIN Applications

- **Door**
  - Mirror
  - Mirror Switch
  - Window Lift
  - Seat Control Switch

LIN Concept

- Low cost single- wire implementation (enhanced ISO 9141)
- Speed up to 20Kbit/s
- Variable length of data frame (2, 4 and 8 bytes)
- Single Master / Multiple Slave Concept

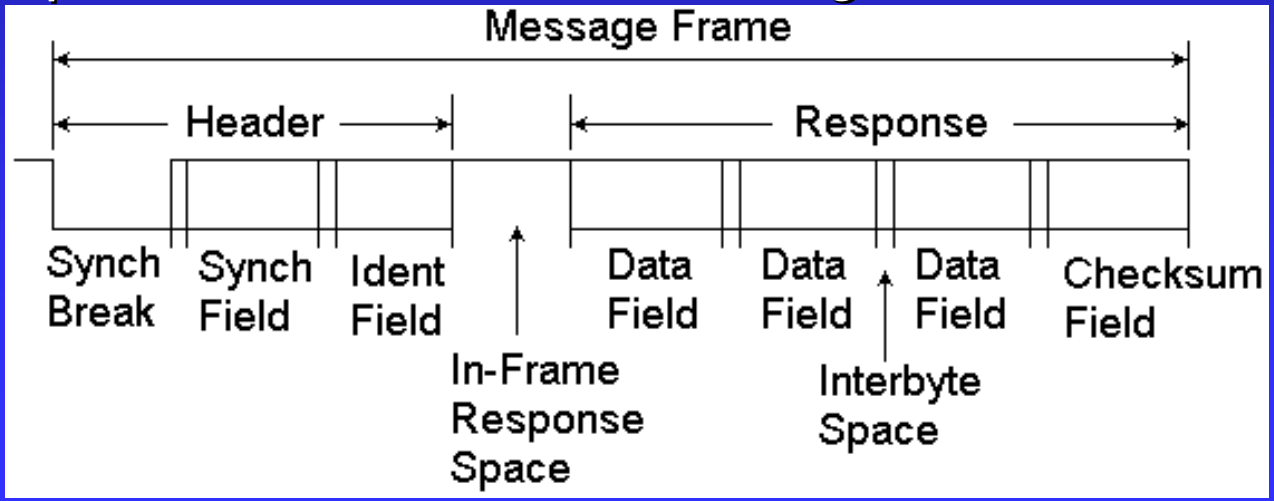
  $\rightarrow$ No arbitration necessary
- Low cost silicon implementation.
- Guaranteed latency times for signal transmission (Predictability)

Master Task

- It has control over the whole Bus and Protocol
- It also has control over error handling.
- To accomplish this the master
  - sends Sync Break
  - sends Sync Byte
  - sends ID- Field
  - monitors Data Bytes
  - receives WakeUp Break from slave nodes when the bus is inactive and they request some action.
  - serves as a reference with its clock base.

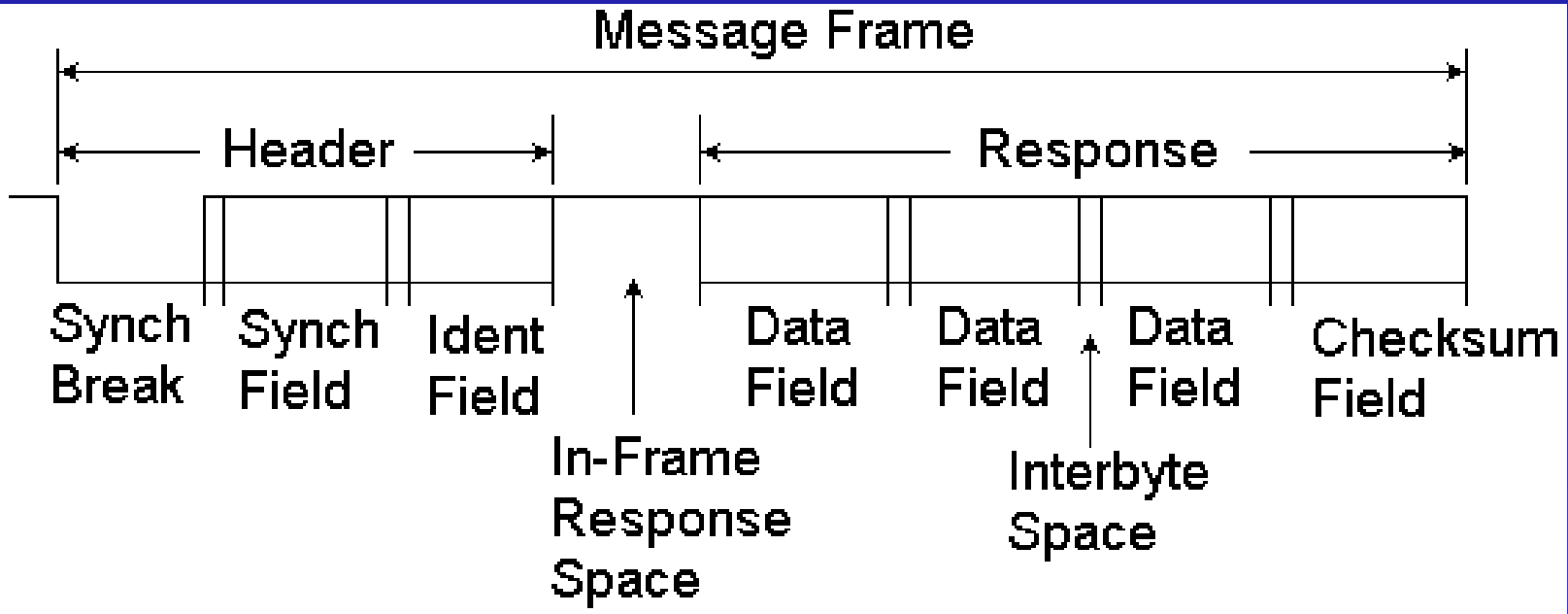
LIN: Message Frame Header

- Synch Break: Marks the Beginning of a Message Frame
- Synch Byte: Specific Pattern for Determination of Time Base (Determination of the time between two rising edges)
- ID- Field: Message Identifier: Incorporates Information about the sender, the receiver(s), the purpose, and the Data field length.



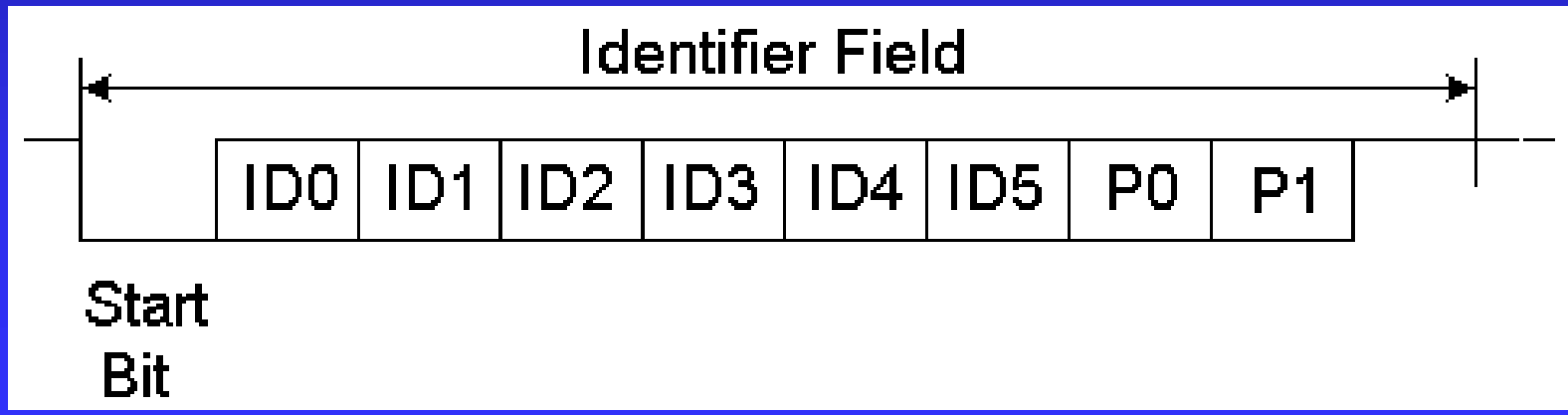
LIN: Message Frame Header

- The length coding is in the 2 MSB of the ID-Field.
- 2 linked Parity Bits protect this highly sensitive ID- Field.



LIN: Identifier Field

- There are 6 identifier bits and two parity bits
- The length coding is in the 2 MSB of the ID-Field.
- 2 Parity Bits protect this highly sensitive ID-Field.



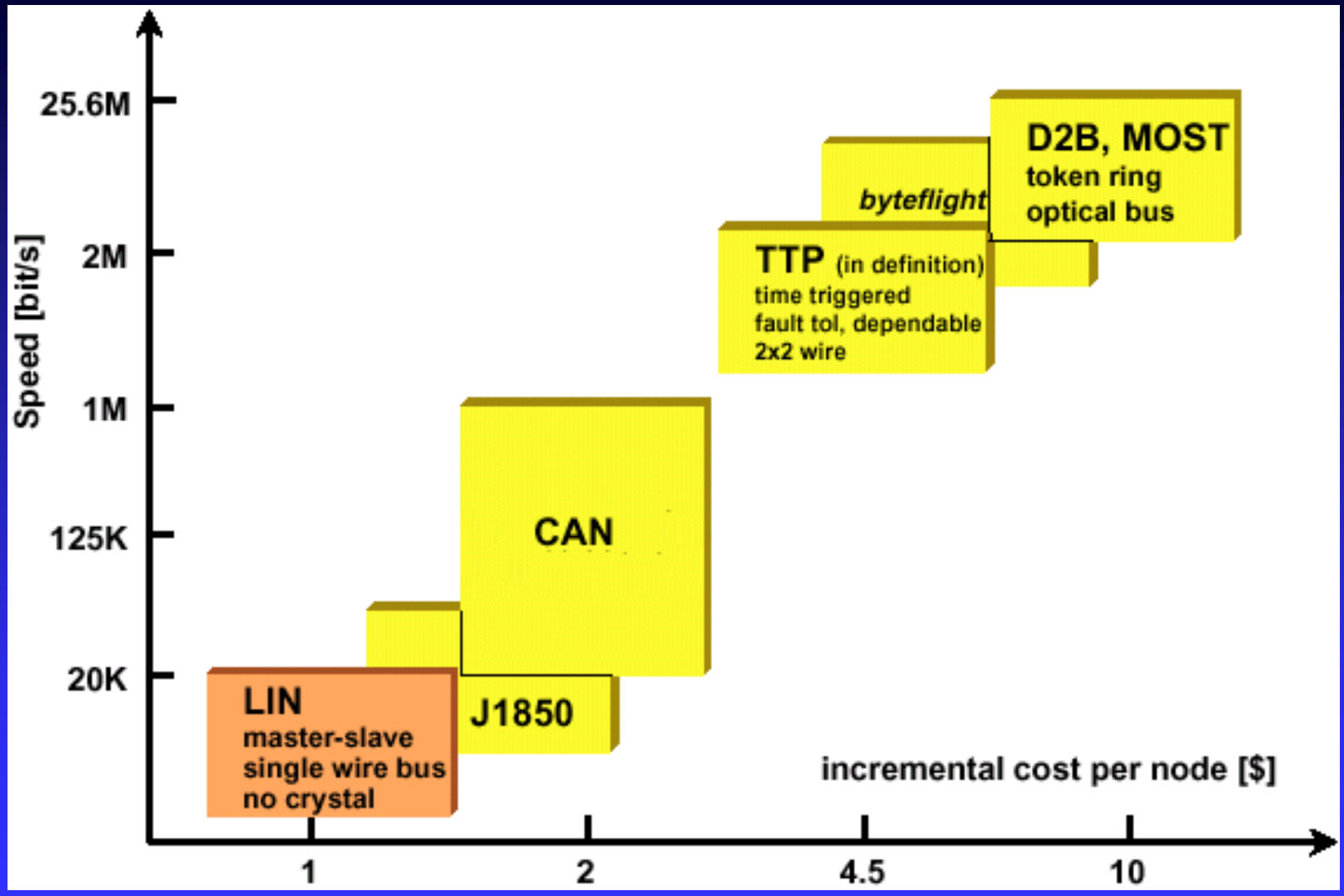
Slave Task

- **Is one of 2- 16 members on the bus and receives or transmits data when an appropriate ID is sent by the master.**
  - Slave waits for Sync Break
  - Slave synchronizes on Sync Byte
  - Slave snoops for ID.

Slave Task (contd.)

- **According to ID, slave determines what to do.**
  - either receive data
  - or transmit data
  - or do nothing.
- **When transmitting the slave**
  - sends 2, 4, or 8 Data Bytes
  - sends Check- Byte
- **The node serving as a master can be slave, too!**

Automotive Bus Systems



Courtesy of Hans-Chr. v. d. Wense, Motorola

CAN versus Other Protocols

	CAN 2.0A/B	SAE J1850	BEAN
Bit Encoding	NRZ	PWM or VPW	NRZ
Bus Wire Medium	Single or Dual	Single (10.4Kbps) or Dual (41.0Kbps)	Single
Data Rate	1Mbps	10.4 Kbps VPW or 41.7 Kbps PWM	10kbps
# of SOF Bits	1bit	unique symbol	1 bit
# of Identifier Bits	11/29 bits	8 to 24 bits	12 bits
Data Length Code	4 bits	none	4 bits
Message Length Field	0 to 24 bits	0 to 24 bits	1 to 88
CRC Field	15 bits	8 bits	8 bits
ACK Field	2 bits	none	2 bits
End of Frame	7 bits	unique symbol	6 bits
IFR	ACK	a) 1 byte from 1 receiver or- b) Multiple bytes from multiple receivers. c) Data bytes, with or without CRC, from a single receiver.	
EOF	1 bit	1 bit	1 bit

BLUETOOTH

A Protocol for Wireless Communications

- **Introduction**
- **History**
- **Bluetooth can save lives**
- **Piconet**
- **Technical Challenges**
- **Bluetooth Layers**

Introduction

- Bluetooth is a new technology standard using short-range radio links, intended to replace the cable(s) connecting portable and/or fixed electronic devices.
- Bluetooth implementations support a range of roughly 10 meters, and throughput up to 721 Kbps.
- A uniform structure for a wide range of devices to communicate with each other, with minimal user effort.
- Its key features are robustness, low complexity, low power and low cost.

Introduction

- Bluetooth is ideal for applications such as wireless headsets, wireless synchronization of PDAs with PCs, and wireless PC peripherals such as printers, keyboards, or mice.
- It also offers wireless access to LANs, the mobile phone network and the internet.

History

- Bluetooth was invented in 1994 by Ericsson of Sweden.
- The standard is named after Harald Blaatand "Bluetooth" II, king of Denmark 940-981A.D.
- The Bluetooth Special Interest Group (SIG) was founded by Ericsson,IBM,Intel,Nokia and Toshiba in February 1998, to develop specifications.
- More than 1900 companies have joined the SIG .

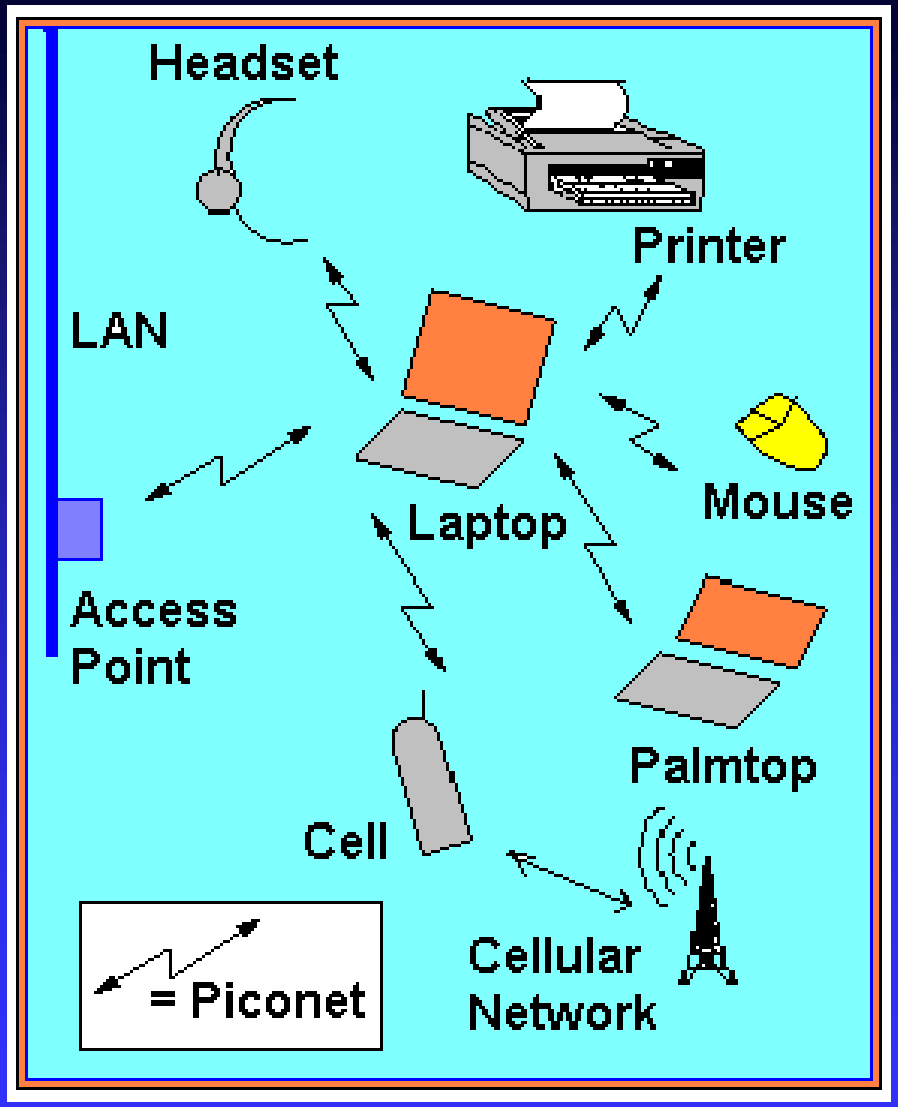
Death and Injury due to Tire Pressure Problem

- **According to the U.S. National Highway Traffic Safety Administration, if all 2003 model-year vehicles were equipped with tire-pressure monitoring systems, some 80 deaths and 10,000 injuries could be avoided every year...**

Bluetooth Can Save Lives

- **Nokian has developed the RoadSnoop Safety System that provides the driver with real-time updates on tire condition using Bluetooth technology.**
- **The system is housed entirely in a chip mounted in the tire, meaning no additional equipment is required in the car. Should the tire pressure drop, the driver is alerted to the fact by a simple phone call.**
- **This a practical use of the technology.**

Piconet


Headset
Printer
LAN
Mouse
Laptop
Access
Point
Palmtop
Cell
Cellular
Network
= Piconet

Piconet (continued)

- Bluetooth enabled electronic devices connect and communicate wirelessly via short-range, networks called piconets.
- One unit acts as the master of the piconet, whereas the other unit(s) acts as slave(s).

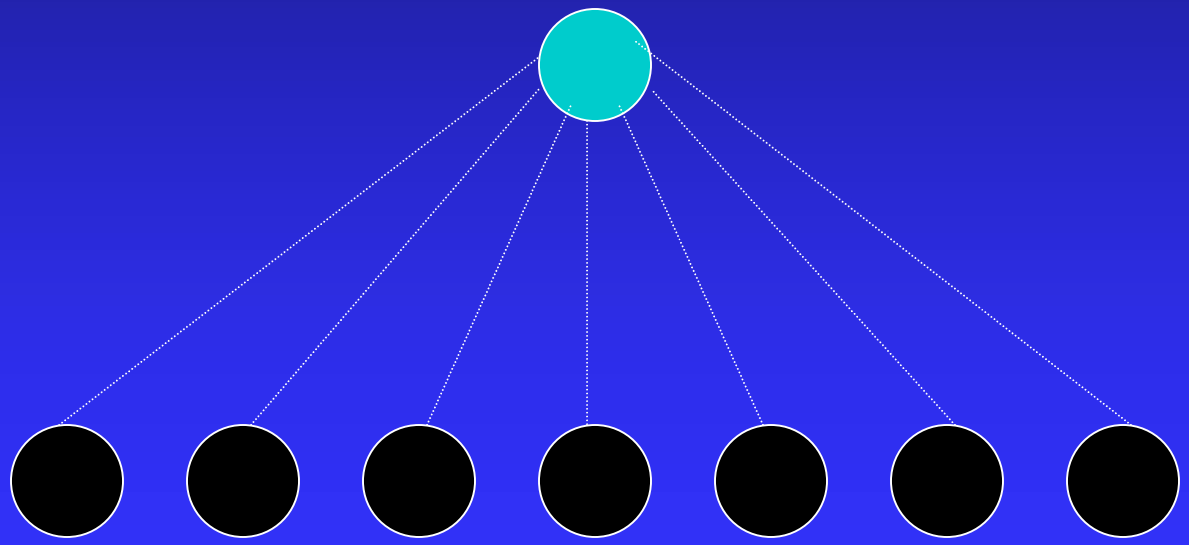
A Piconet with one Master and one Slave

Master Slave

Piconet (continued)

- **Each unit can simultaneously communicate with up to seven other units per piconet.**

A Piconet with one Master and seven Slaves

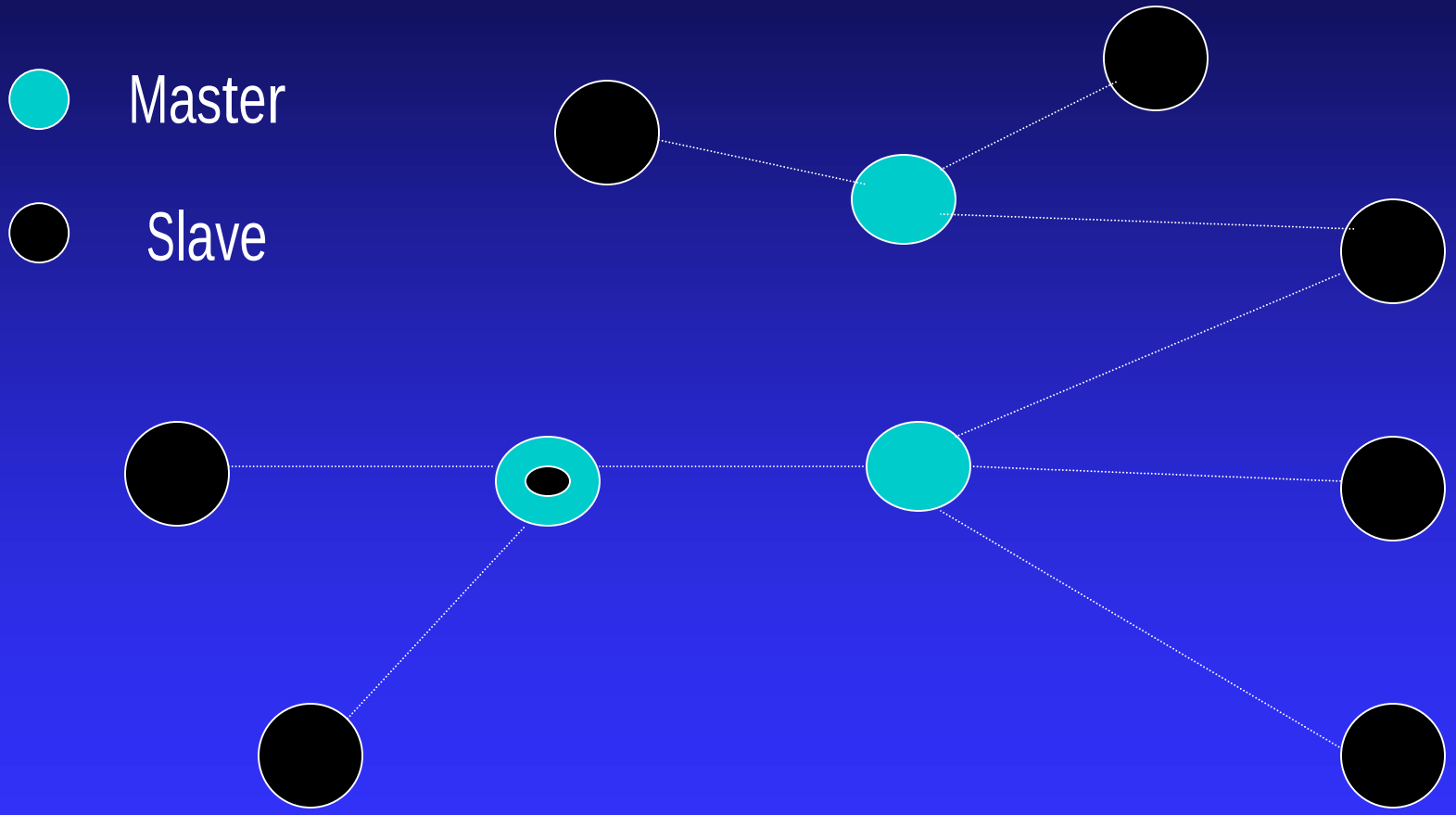


Piconet (continued)

- **Slaves can participate in different piconets on a time-division multiplex basis.**
- **A master in one piconet can be a slave in another piconet.**
- **The piconets are established dynamically and automatically as Bluetooth devices enter and leave the radio proximity**

Scatternet

Multiple piconets with overlapping coverage areas form a scatternet.



Technical Challenges

- **The system has to use an unlicensed band for universal acceptance and usage. Thus the Industrial, Scientific and Medical (ISM) band has been selected for Bluetooth. (2.4GHz)**

- **The challenge here is to make the system robust to interference from other sources in this band, which include not only ISM band communication systems but also microwave ovens.**

Technical Challenges

- The transceivers should be able to **adapt to a rapidly changing environment** as the devices will usually be mobile.
- The **multipath fading problem** must be handled.
- The **connection establishment and routing protocols** have to operate in an environment where the number, location and variety of Bluetooth devices will change dynamically.

Technical Challenges

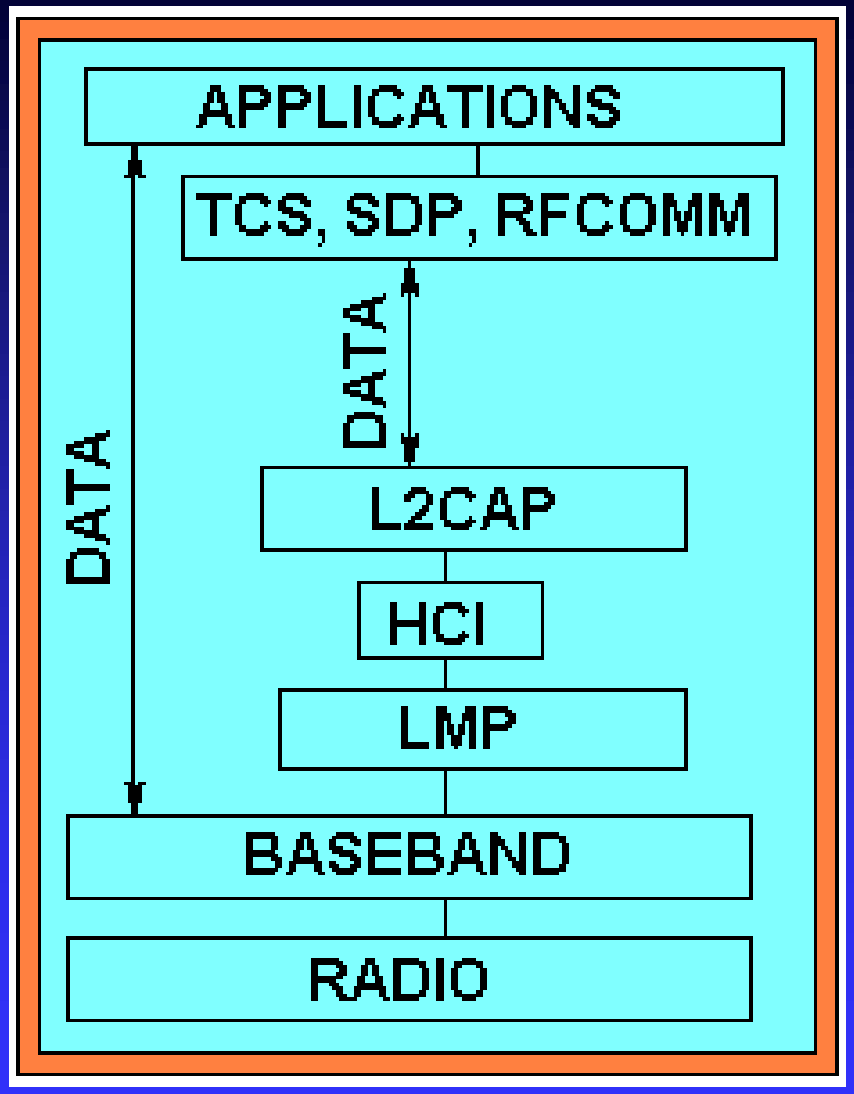
- The **size of the implementation** should be small for easy integration into handheld and mobile devices.
- The **power consumption** should not be more than a small fraction of the host device into which the Bluetooth capability is to be introduced.
- **Automatic connection establishment** must be provided, because the number of devices in proximity will change quite frequently and it will be inconvenient to establish connections manually each time.

Technical Challenges

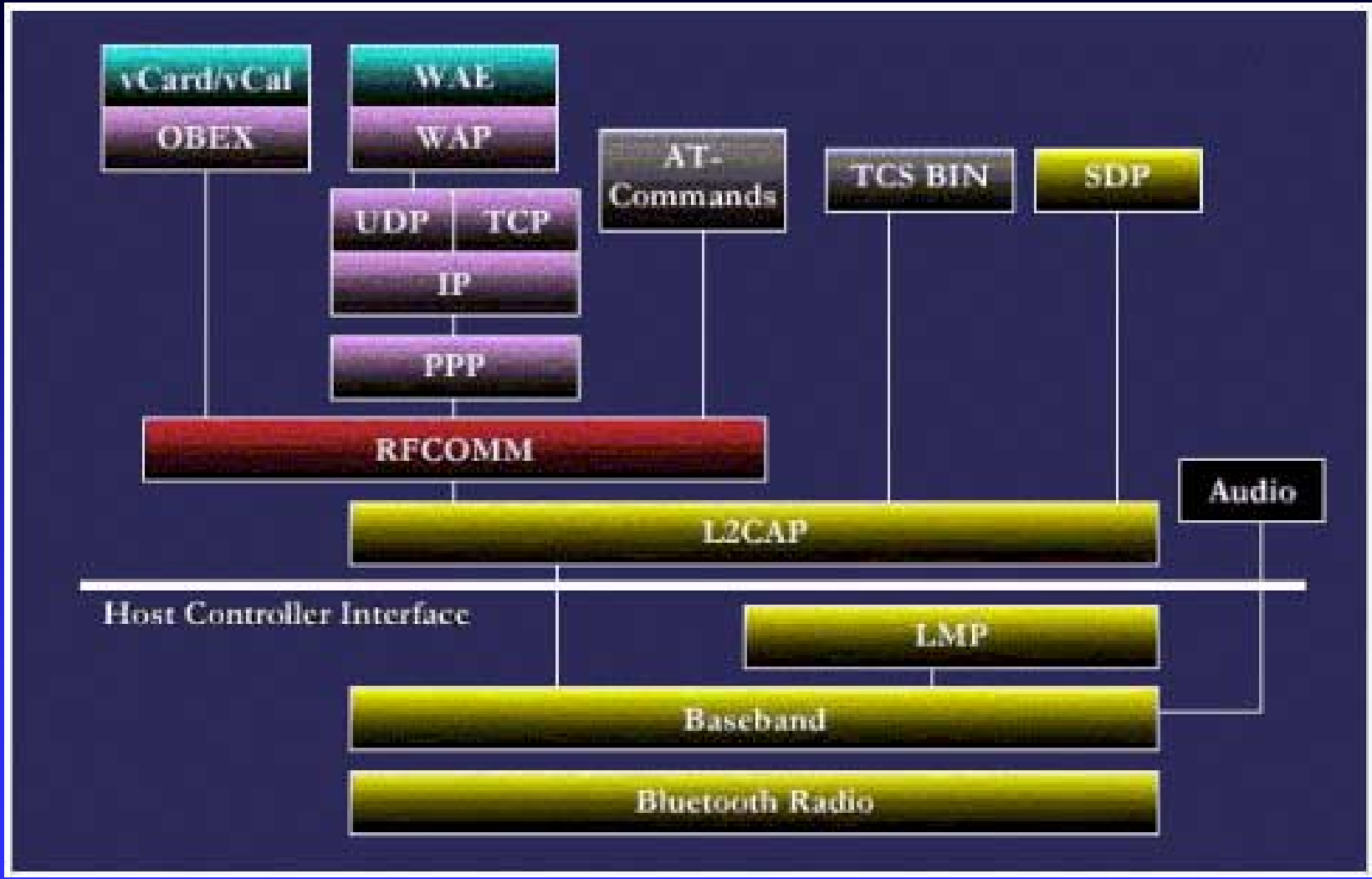
- **Synchronization of clocks** among the communicating units will have to be achieved, because each unit will have its own free running clock with its own drift.

- The Bluetooth devices will be part of people's personal usage and will contain and communicate their personal information. **Encryption facility must be provided**.

Bluetooth Protocol Stack



Bluetooth Protocol Stack



Courtesy of Palowireless

http://www.palowireless.com/infotooth/tutorial.asp

Bluetooth Layers

- Protocol stack consists of a **Radio layer** at the bottom which forms the physical connection interface.
- The **Baseband** and **Link Manager Protocol** (LMP) layers establish and control links between Bluetooth devices.
- Radio, Baseband and LMP layers are typically implemented in hardware/firmware.

Bluetooth Layers

- The **Host Controller layer** (HCI) is required to interface the Bluetooth hardware to **L2CAP** (Logical Link Control and Adaptation Protocol) layer.
- The host controller is required only when the L2CAP resides in software in the host.
- If the L2CAP is also on the Bluetooth module, HCI may not be required as then the L2CAP can directly communicate with the LMP and baseband.
- Applications reside above L2CAP.

Radio Layer

FREQUENCY BAND AND RF CHANNELS

- This link operates in the unlicensed ISM (Industrial, Scientific and Medicine) band around 2.4 GHz .
- Channel spacing is 1 MHz.
- There are 79 RF channels in USA.

Geography	Regulatory Range	RF Channels
USA, Europe and most other countries[1)]	2.400-2.4835 GHz	f=2402+k MHz, k=0,…,78
Spain[2)]	2.445-2.475 GHz	f=2449+k MHz, k=0,…,22
France[3)]	2.4465-2.4835 GHz	f=2454+k MHz, k=0,…,22

Guard Band

- In order to comply with out-of-band regulations in each country, a guard band is used at the lower and upper band edge.

Geography	Lower Guard Band	Upper Guard Band
USA	2 MHz	3.5 MHz
Europe (except Spain and France)	2 MHz	3.5 MHz
Spain	4 MHz	26 MHz
France	7.5 MHz	7.5 MHz
Japan	2 MHz	2 MHz

Radio Layer

- Frequency Hopping (FH) technique is used
  - As multiple uncoordinated networks may exist in this band and cause interference, fast FH and short data packets are used
  - As the error rate may be high, especially due to strong interference from microwave ovens which operate at this frequency, CVSD (Continuous Variable Slope Delta) coding has been adopted for voice, which can withstand high bit error rates

Baseband Layer

- Controls the Radio layer.
- Provides frequency hop sequences.
- Takes care of lower level encryption for secure links.
- Handles packet over the wireless link.
- Two types of links can be established:
  - SCO: Synchronous Connection Oriented, meant for synchronous data typically voice.
  - ACL: Asynchronous Connection Less, used for data transfer applications.

Baseband Layer

- Synchronizes devices' clocks and establish connections.
- Discovers devices' addresses in proximity.
- Handles error correction for packets.
- Performs data whitening.
- Defines functions for encryption keys and link keys.

Link Manager Protocol (LMP) Layer

Piconet management

- A piconet is a group of devices connected to a common channel, which is identified with its unique hop sequence
- One of the devices, usually the one which first initiated the connection is the master
- Upto seven other devices can be actively connected to this master, and many more could be connected in a low power "parked" state
- The devices on one piconet can communicate with each other over SCO or ACL links

LMP Layer (continued)

The LMP provides the functionality to

- attach/detach slaves,
- switch roles between a master and a slave

- LMP also handles the low power modes: hold, sniff and park, to save power when the device does not have data to send.

Link configuration tasks include

- setting link parameters,
- authentication of devices
- managing link keys

LMP Layer (continued)

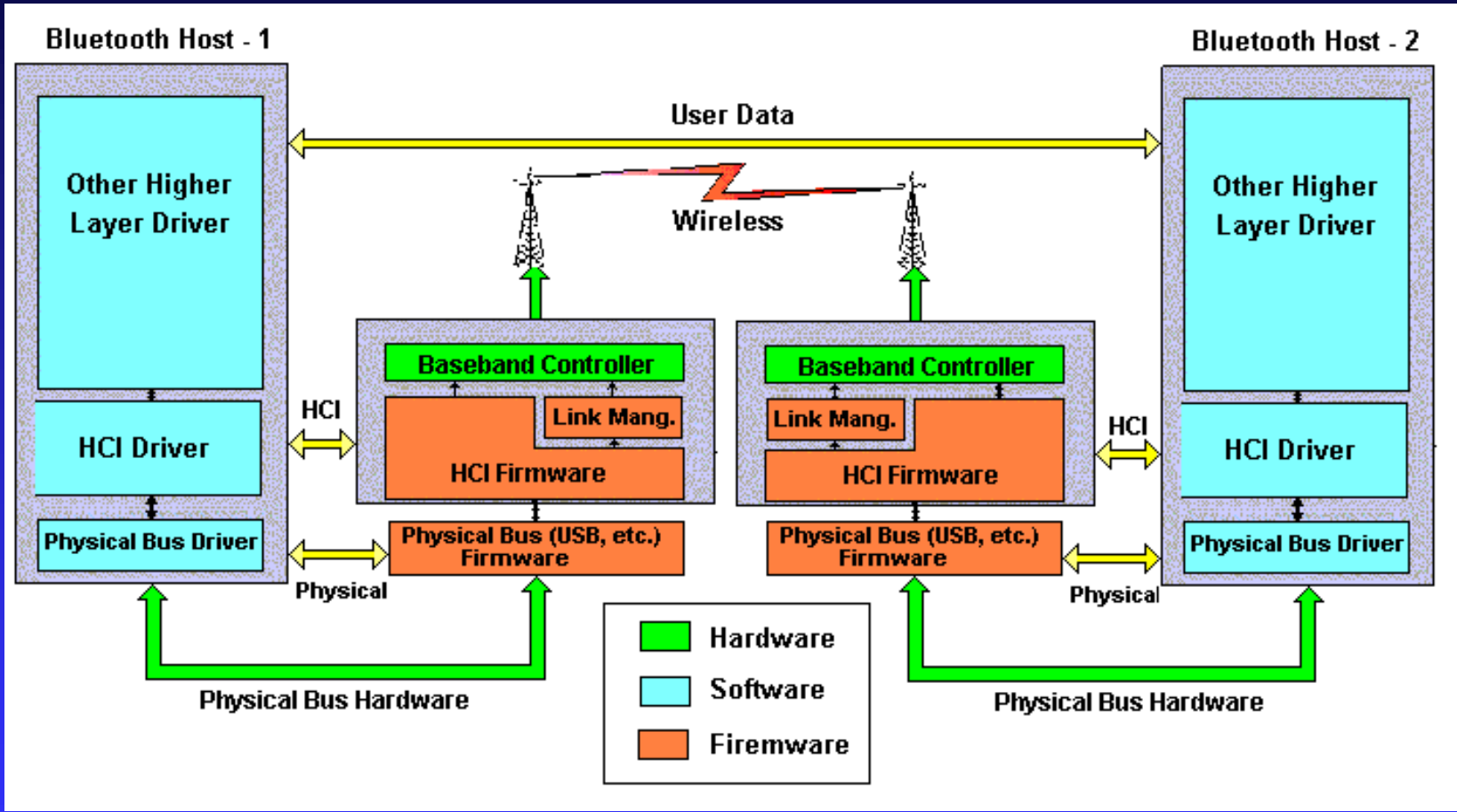
Link configuration includes tasks such as:

- **Setting link parameters**
- **Quality of Service**
- **Power control if the device supports it**

Security functions such as:

- **Authentication of devices to be linked**
- **Managing link keys**

Host Controller Interface (HCI)



HCI (continued)

- For many devices, Bluetooth enabling module may be added as a separate card
  - For instance, on a PC or a laptop, the Bluetooth hardware may be added as a PCI card or a USB adapter
- Hardware modules usually implement the lower layers: radio, baseband and LMP. Then the data to be sent to LMP and baseband travels over the physical bus like USB.
  - A driver for this bus is required on the "host", that is PC, and
  - A "host controller interface" is required on Bluetooth hardware card to accept data over the physical bus

HCI (continued)

- **If the higher Bluetooth layers, L2CAP and above are in software and the lower ones in hardware, the following extra layers are at least required:**

- **HCI driver**
  - **This is the driver for host controller interface.**
  - **It resides in the host, above the physical bus, and formats the data to be accepted by the Host Controller on the Bluetooth hardware.**

- **Host Controller Interface**
  - **This resides on the Bluetooth hardware and accepts communications over the physical bus.**

Logical Link Control and Adaptation Protocol (L2CAP)

- Multiplexing
  - Protocol must allow multiple applications to use a link between two devices simultaneously

L2CAP (continued)

- Segmentation and Reassembly
  - Protocol must reduce size of packets provided by applications to size of packets accepted by Baseband
  - L2CAP itself accepts packet sizes upto 64kb but baseband can accept a payload of at most 2745 bits
  - The reverse procedure, that of combining the segmented packets in the proper order, has to be carried out for received packets

L2CAP (continued)

- Quality of Service
  - Allow applications to demand QoS on certain parameters like peak bandwidth, latency and delay variation
  - Check if the link is capable of providing it and provides it if possible

- Basically, L2CAP provides network layer functions to applications and higher protocols

Application Layer

- L2CAP may be accessed directly by applications or through certain support protocols like RFCOMM, TCS and SDP.
- Applications may use other protocols like TCP-IP or WAP.
- Applications may themselves run PPP, FTP or other specific protocols as required by the application.
- An application may use SDP to discover whether the service it needs from a remote device is available.

Connection Steps

- Assume that a person walks in to a hotel lobby and wants to access his email through his Bluetooth device, which could be a laptop or a Personal Digital Assistant.
- Depending on the implementation, the person would be clicking on a menu or an email application icon.
- The device would then automatically carry out the following steps, (except perhaps for the authentication step if the device has come to the environment for the first time):

Connection Steps

Inquiry

- The device on reaching a new environment would automatically initiate an inquiry to find out what access points are within its range. This will result in the following events:
  - All nearby access points respond with their addresses
  - The device picks one out the responding devices

Connection Steps (continued)

Paging

- The device will invoke a baseband procedure called paging.
- This results in synchronization of the device with the access point, in terms of its clock offset and phase in the frequency hop, among other required initializations

Connection Steps (continued)

Link establishment

- The LMP will now establish a link with the access point. As the application in this case is email, an ACL link will be used

Service Discovery

- LMP uses SDP (Service Discovery Protocol) to discover what services are available from the access point, in particular whether email access or access to the relevant host is possible from this access point or not

Connection Steps (continued)

L2CAP channel

- With information obtained from SDP, the device will create an L2CAP channel to the access point. This may be directly used by the application or another protocol like RFCOMM.

RFCOMM channel

- Depending on need of email application an RFCOMM or other channel (in case of other applications) will be created over the L2CAP channel

Connection Steps (continued)

Security

- If the access point restricts its access to a particular set of users or otherwise offers secure mode communications to people having some prior registration with it, then at this stage, the access point will send a security request for "pairing”
- This will be successful if the user knows the correct PIN code to access the service

Connection Steps (continued)

Security

- Note that the PIN is not transmitted over the wireless channel but another key generated from it is used, so that the PIN is difficult to compromise. Encryption will be invoked if secure mode is used

Connection Steps (continued)

PPP

- **Assuming that a PPP link is used over serial modem as in dial up networking, the same application will now be able to run PPP over RFCOMM (which emulates the serial port)**

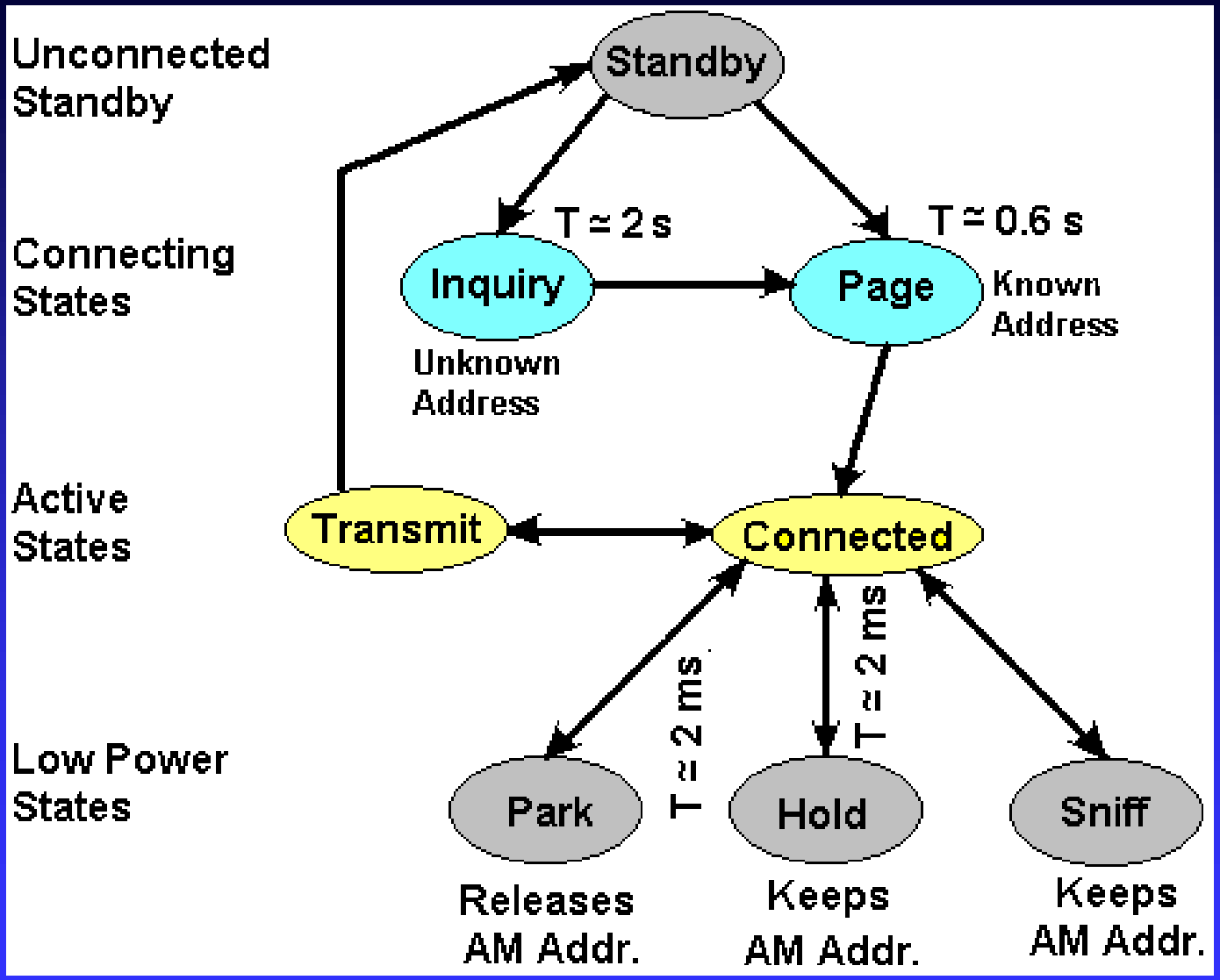
Network Protocols

- **The network protocols like TCP/IP, IPX, Appletalk can now send and receive data over the link**

States of Bluetooth Devices

- Standby
- Inquiry, Inquiry Scan, Inquiry Response
- Page, Page Scan, Page Response
- Connected
- Transmit
- Park
- Hold
- Sniff

States


Unconnected Standby
Connecting States
Active States
Low Power States
Standby
T ≃ 2 s
T ≃ 0.6 s
Inquiry
Page
Known Address
Unknown Address
Transmit
Connected
T ≃ 2 ms
T ≃ 2 ms
Park
Hold
Sniff
Releases AM Addr.
Keeps AM Addr.
Keeps AM Addr.

Standby State

- Before any connections in a piconet are created, all devices are in STANDBY mode.
- In this mode, an unconnected unit periodically "listens" for messages every 1.28 seconds.
- Each time a device wakes up, it listens on a set of 32 hop frequencies defined for that unit.
- The number of hop frequencies varies in different geographic regions; 32 is the number for most countries (except Japan, Spain and France).
- The connection procedure is initiated by any of the devices which then becomes master.

Inquiry, Inquiry Scan, Inquiry Response

- The INQUIRY message is used for finding devices, including public printers, fax machines, etc. with an unknown address.
- A device which that is looking for other devices goes to the ***Inquiry State***, and sends the inquiry message.
- If other devices are present within the range, then each one of those devices go to ***Inquiry Scan*** and ***Inquiry Response States*** to indicate their presence.
- After the inquiry process, the former device knows the addresses of the other devices.

Page, Page Scan, Page Response

- A connection is made by a PAGE message if the address is already known.
- In the initial PAGE state, the master unit will send a train of 16 identical page messages on 16 different hop frequencies defined for the device to be paged (slave unit).
- If no response, the master transmits a train of the remaining 16 hop frequencies in the wake-up sequence.
- The maximum delay before the master reaches the slave is twice the wakeup period (2.56 seconds) while the average delay is half the wakeup period (0.64 seconds).

Connected and Transmit States

- In these states a device remains active.

Sniff State

- A power saving mode can be used for connected units in a piconet if no data needs to be transmitted.
- In the sniff mode, the slave's listen activity is reduced.
- If a slave participates on an ACL link, it still listens to that link, in Sniff mode, but at a reduced rate.
- To enter the sniff mode, the master shall issue a sniff command via the LM protocol.

Hold State

- The master can put the ACL link to a slave in a hold mode.
- This means that the slave temporarily does not support ACL packets on the channel any more (note: possible SCO links will still be supported).
- Slave units can also demand to be put into HOLD mode.
- The HOLD is used when connecting several piconets or managing a low power device such as a temperature sensor.

Park State

- When a slave does not need to participate on the piconet channel, but still wants to remain synchronized to the channel, it can enter the park mode which is a low-power mode with very little activity in the slave.
- In the park mode, the slave gives up its active member address AM_ADDR.
- Instead, it receives two new addresses to be used in the park mode
  - PM_ADDR: 8-bit Parked Member Address
  - AR_ADDR: 8-bit Access Request Address

Park State (contd.)

- The PM_ADDR distinguishes a parked slave from the other parked slaves.
- This address is used in the master-initiated unpark procedure.
- In addition to the PM_ADDR, a parked slave can also be unparked by its 48-bit BD_ADDR.
- The AR_ADDR is used by the slave in the slave-initiated unpark procedure.
- All messages sent to the parked slaves have to be carried by broadcast packets (the all-zero AM_ADDR) because of the missing AM_ADDR.

Inquiry Process

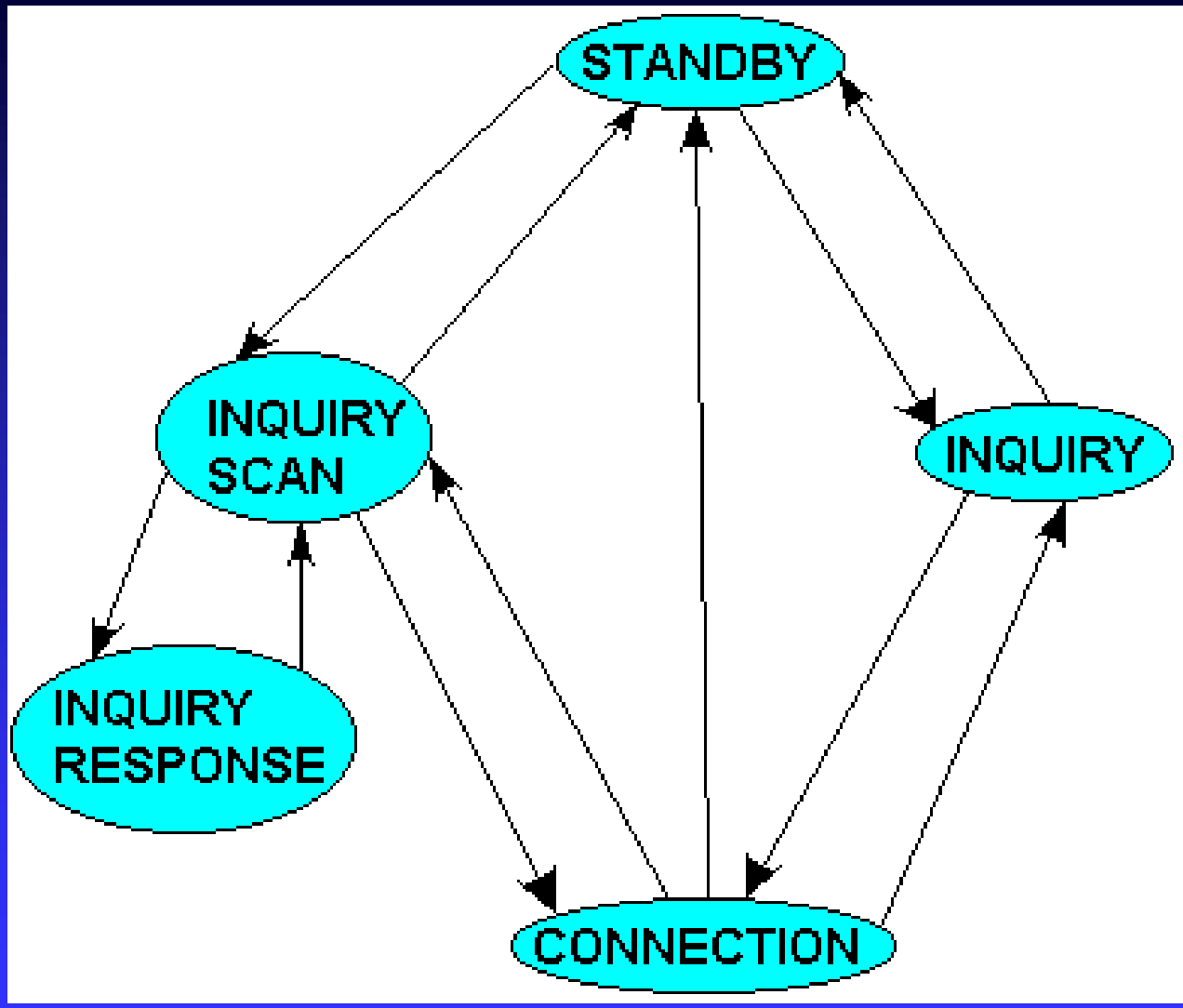
- If a device wants to discover new devices within the range, it goes to the INQUIRY state and sends an inquiry message.
- Another device which is present within the range, goes to Inquiry Scan state and performs the following operations in order to avoid collision.

1. Generates a random number (N) between 0 and 1023, and returns back to Standby or Connection state from where it came.
2. After at least N slots it comes back to the Inquiry Scan state again and waits for another Inquiry message.

Inquiry Process (contd.)

3. At the first inquiry message, it goes to the Inquiry Response state and sends a response signal. After that, it comes back to the Inquiry Scan state.
4. If this device is triggered again by another Inquiry Message, then it repeats Steps 1-3 again. But, if it is not triggered again within a time out period, then it returns to the Standby or Connected state.

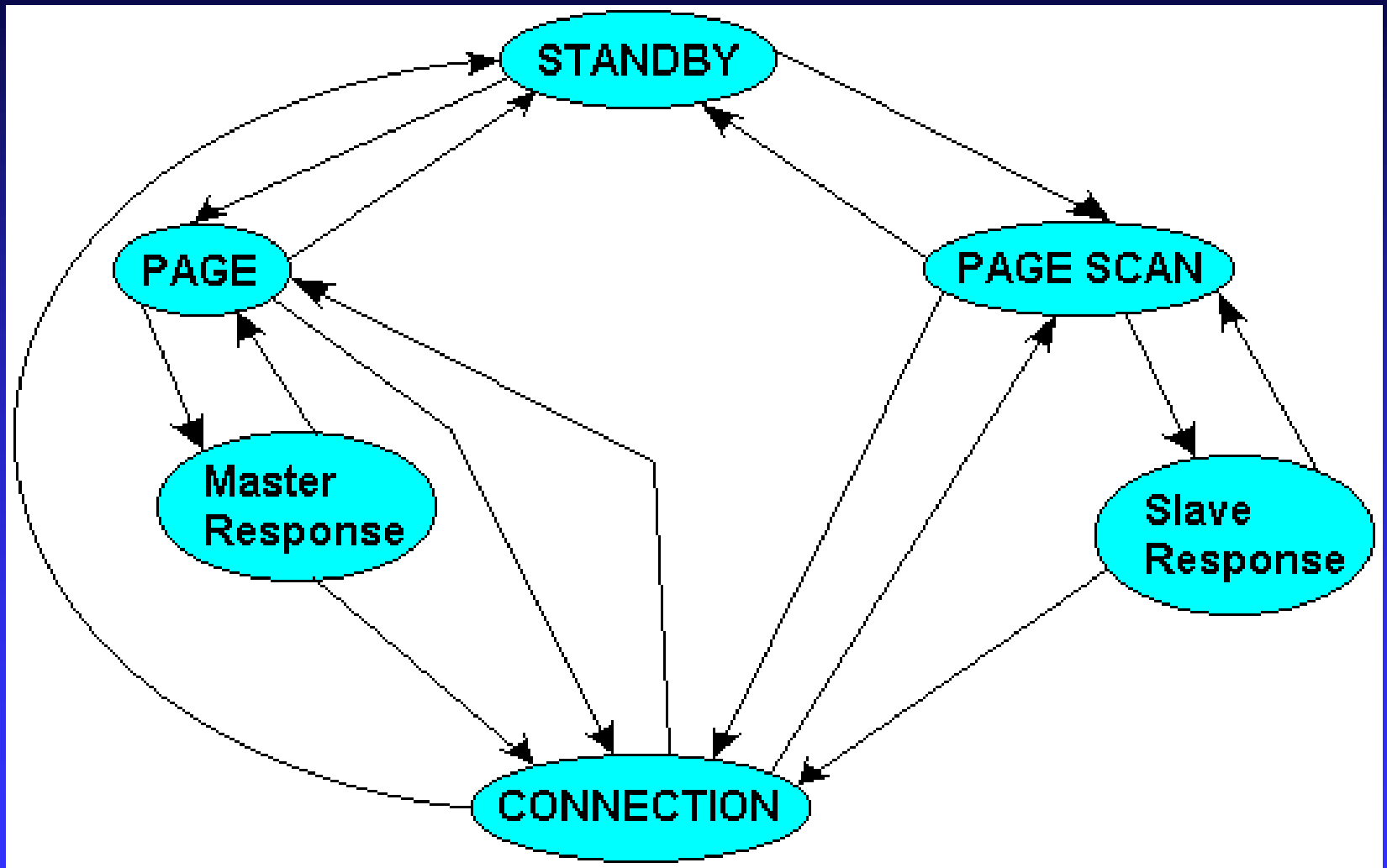
State Diagram of Inquiry Process



Paging Process

- The paging process is initiated by the master in order to connect a new slave to the piconet.

State Diagram of Bluetooth Link Controller



Baseband Specification

1. General Description
2. Physical Channels
3. Physical Links
4. Packets
5. Error Correction
6. Logical Channels
7. Data Whitening
8. Transmit/Receive Routines

1. GENERAL DESCRIPTION

- A frequency hop transceiver is applied to combat interference and fading.
- A shaped, binary FM modulation is applied to minimize transceiver complexity.
- A slotted channel is applied with a nominal slot length of 625 micro secs.
- For full duplex transmission, a Time-Division Duplex (TDD) scheme is used.

General Description (contd.)

- On the channel, information is exchanged through packets.
- Each packet is transmitted on a different hop frequency.
- A packet nominally covers a single slot, but can be extended to cover up to five slots.

2. PHYSICAL CHANNELS

CHANNEL DEFINITION

- The channel is represented by a pseudo-random sequence hopping through the 79 or 23 RF channels.
- The hopping sequence is unique for a piconet and is determined by the BD_Address of the master. ***(BD_Address : Bluetooth Device Address)***
- The phase in the hopping sequence is determined by the clock of the master.

Channel Definition (contd.)

- The channel is divided into time slots where each slot corresponds to an RF hop frequency.
- Consecutive hops correspond to different RF hop frequencies.
- The nominal hop rate is 1600 hops/s. All Bluetooth units participating in the piconet are time- and hop-synchronized to the channel.

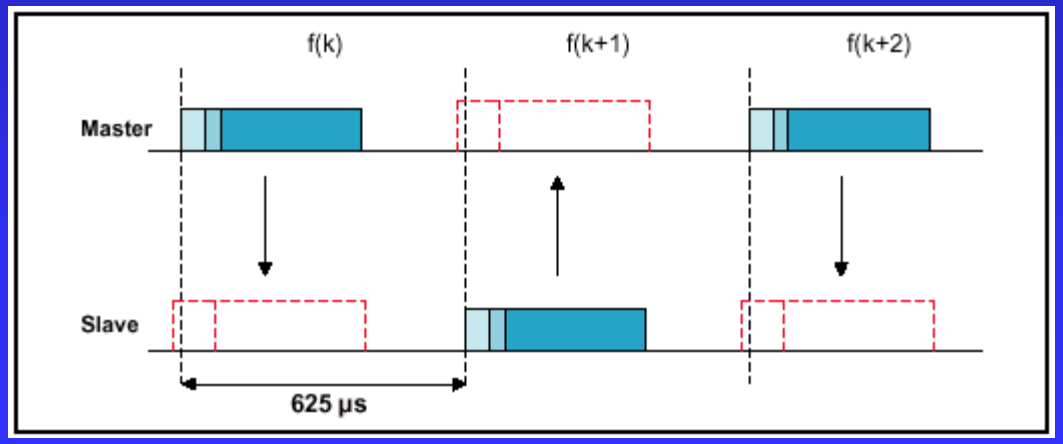
Time Slots

- The channel is divided into time slots, each 625 μs in length.
- The time slots are numbered according to the clock of the piconet master.
- The slot numbering ranges from 0 to $2^{27}-1$ and is cyclic with a cycle length of 2^{27}.
- In the time slots, master and slave can transmit packets.
- A TDD (Time-Division Duplex) scheme is used where master and slave alternatively transmit.

Time Slots (contd.)

- The master shall start its transmission in even-numbered time slots only, and the slave shall start its transmission in odd-numbered time slots only.
- The packet start shall be aligned with the slot start.

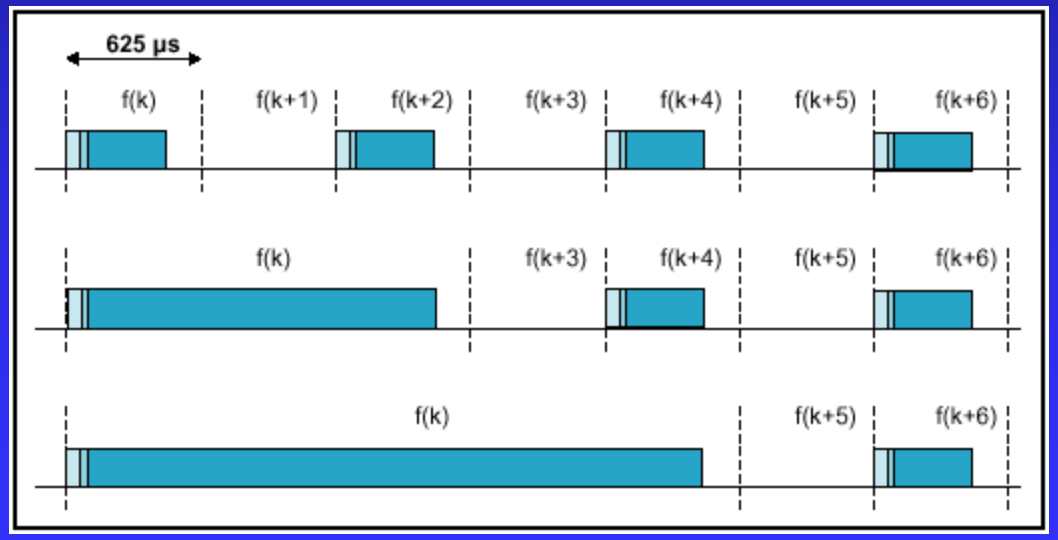
TDD and Timing



Time Slots (contd.)

- Packets transmitted by the master or the slave may extend over up to five time slots.
- The RF hop frequency shall remain fixed for the duration of the packet.

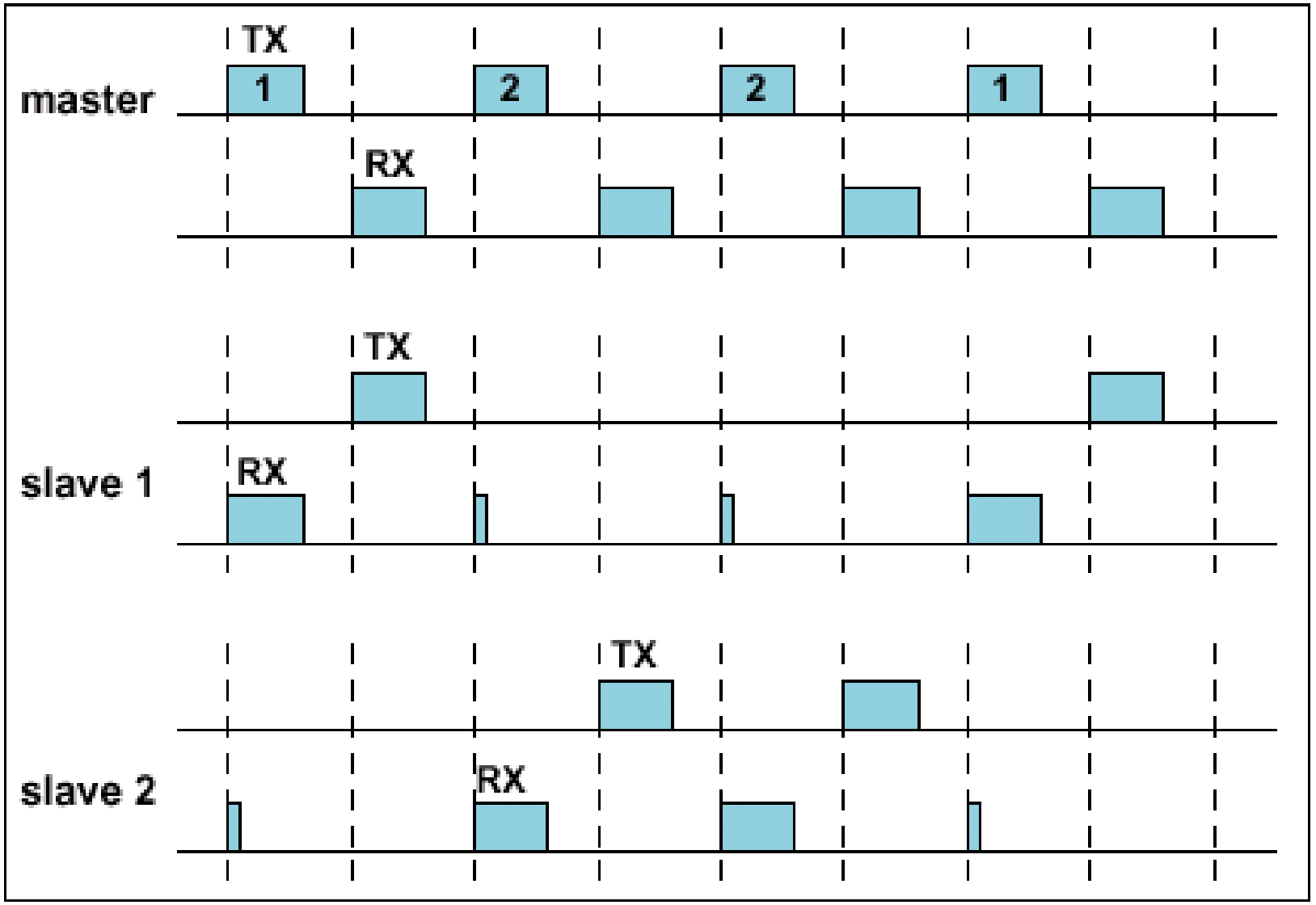
Multi-slot Packets



Time Slots (contd.)

- For a single packet, the RF hop frequency to be used is derived from the current clock value.
- For a multi-slot packet, the RF hop frequency to be used for the entire packet is derived from the clock value in the first slot of the packet.
- The RF hop frequency in the first slot after a multi-slot packet shall use the frequency as determined by the current clock value.

RX/TX Timing in Multi-Slave Configuration

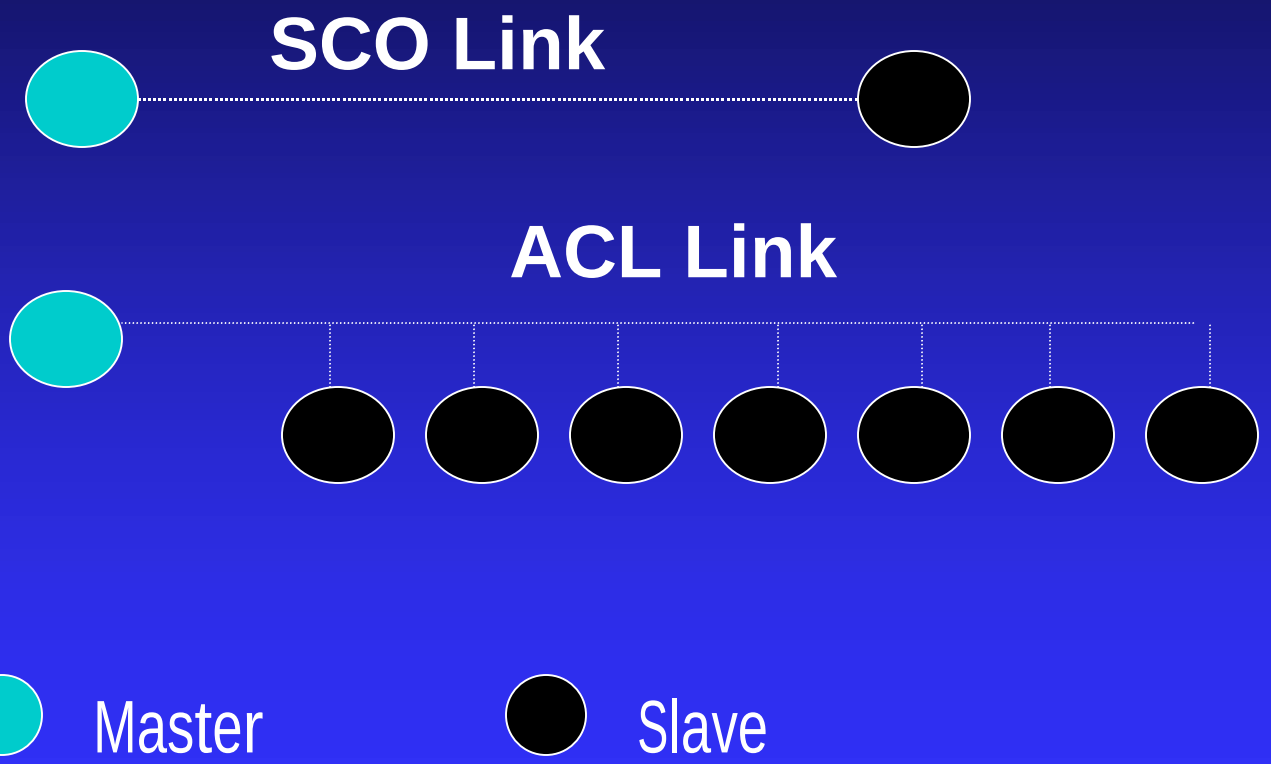


3. PHYSICAL LINKS

GENERAL

- Two link types have been defined:
  - Synchronous Connection-Oriented (SCO) link
  - Asynchronous Connection-Less (ACL) link
- The SCO link is a point-to-point link between a master and a single slave in the piconet.
- The ACL link is a point-to-multipoint link between the master and all the slaves participating on the piconet.

PHYSICAL LINKS



SCO LINK

- The SCO link is a point-to-point link between the master and a specific slave.
- The SCO link reserves slots and can therefore be considered as a circuit-switched connection between the master and the slave.
- The SCO link typically supports time-bounded information like voice.
- The master can support up to **three** SCO links to the same slave or to different slaves.
- A slave can support up to **three** SCO links from the same master, or **two** SCO links if the links originate from different masters.

SCO Links

SCO Links

SCO Links

SCO LINK (contd.)

- SCO packets are never retransmitted.
- The master will send SCO packets at regular intervals T_{SCO} (counted in slots) in the reserved master-to-slave slots.
- The SCO slave is always allowed to respond with an SCO packet in the following slave-to-master slot unless a different slave was addressed in the previous master-to-slave slot.
- If the SCO slave fails to decode the slave address in the packet header, it is still allowed to return an SCO packet in the reserved SCO slot.

SCO LINK (contd.)

- The SCO link is established by the master sending an SCO setup message via the LM protocol.
- This message will contain timing parameters such as the SCO interval T_{SCO} and the offset D_{SCO} to specify the reserved slots.

ACL LINK

- In the slots not reserved for SCO links, the master can exchange packets with any slave on a per-slot basis.
- The ACL link provides a packet-switched connection between the master and all active slaves participating in the piconet.
- Both asynchronous and isochronous services are supported.
- Between a master and a slave only a **single** ACL link can exist.
- For most ACL packets, packet retransmission is applied to assure data integrity.

ACL LINK (contd.)

- A slave is permitted to return an ACL packet in the slave-to-master slot if and only if it has been addressed in the preceding master-to-slave slot.
- If the slave fails to decode the slave address in the packet header, it is not allowed to transmit.
- ACL packets not addressed to a specific slave are considered as **broadcast** packets and are read by every slave.
- If there is no data to be sent on the ACL link and no polling is required, no transmission shall take place.

4. PACKETS

- GENERAL FORMAT
- ACCESS CODE
- PACKET HEADER
- PACKET TYPES
- PAYLOAD FORMAT
- PACKET SUMMARY

GENERAL FORMAT

- The bit ordering when defining packets and messages in the Baseband Specification, follows the ***Little Endian*** format, i.e., the following rules apply:
  - The least significant bit (LSB) corresponds to b_0;
  - The LSB is the first bit sent over the air;
  - In illustrations, the LSB is shown on the left side;
- For example, a 3-bit parameter X=3 is sent as $b_0b_1b_2$=110 over the air where 1 is sent first and 0 is sent last.

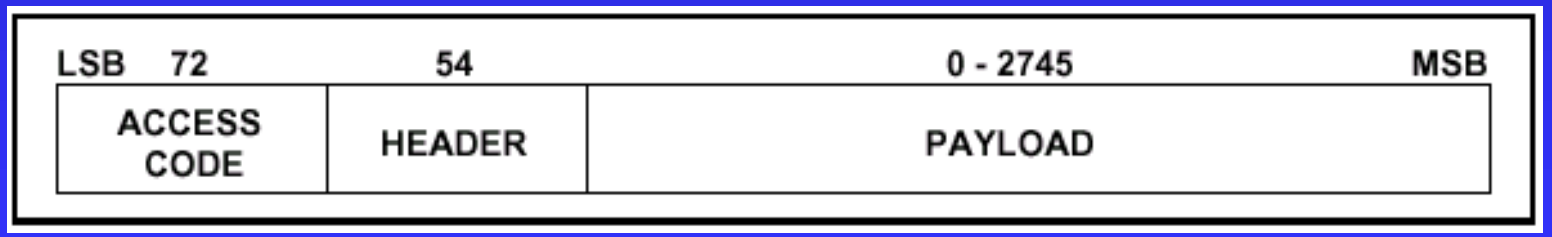
GENERAL FORMAT

- The data on the piconet channel is conveyed in packets.
- Each packet consists of 3 entities: the ***access code***, the ***header***, and the ***payload***.

Access code: 72 bits, Header: 54 bits

Payload: 0-2745 bits

Standard Packet Format



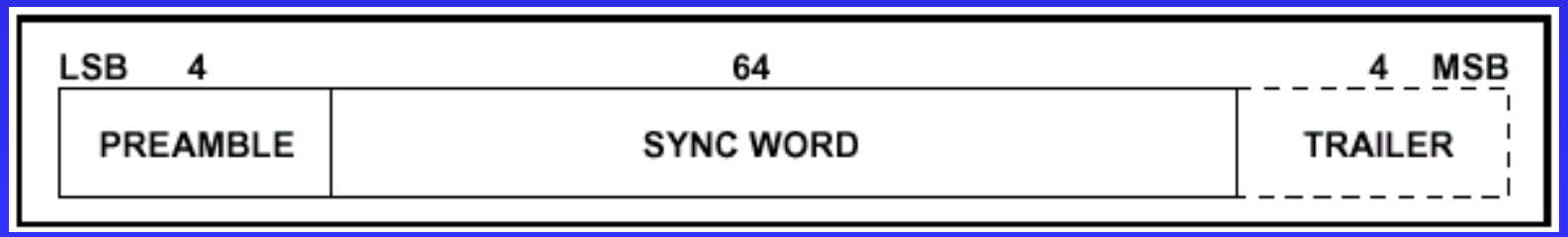
ACCESS CODE

- Each packet starts with an access code.
- If a packet header follows, the access code is 72 bits long, otherwise the access code is 68 bits long.
- This access code is used for synchronization, DC offset compensation and identification.
- The access code identifies all packets exchanged on the channel of the piconet: all packets sent in the same piconet are preceded by the same channel access code.

ACCESS CODE

- The access code is also used in paging and inquiry procedures. In this case, the access code itself is used as a signaling message and neither a header nor a payload is present.
- The access code consists of a ***preamble***, a ***sync word***, and possibly a ***trailer***.

Access Code Format



Access Code Types

- There are three different types of access codes defined:
  - **Channel Access Code (CAC)**
  - **Device Access Code (DAC)**
  - **Inquiry Access Code (IAC)**
- The channel access code identifies a piconet. This code is included in all packets exchanged on the piconet channel.
- The device access code is used for special signaling procedures, e.g., paging and response to paging.

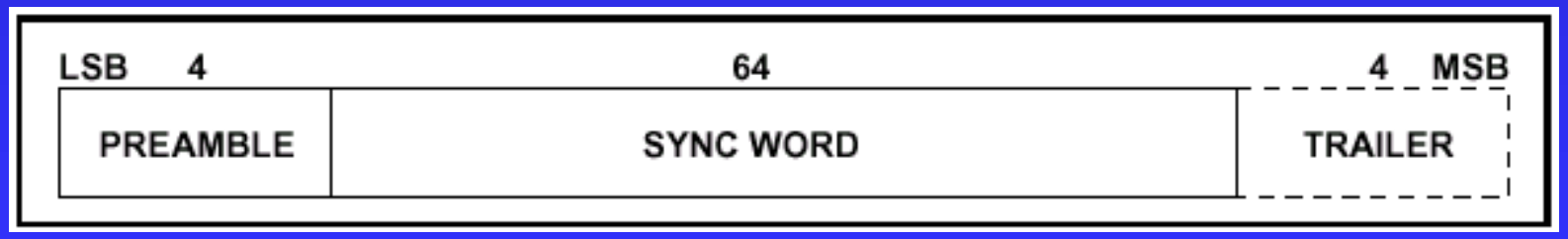
Access Code Types

- For the inquiry access code there are two variations.
- A general inquiry access code (GIAC) is common to all devices.
- The GIAC can be used to discover which other units are in range.
- The dedicated inquiry access code (DIAC) is common for a dedicated group of units that share a common characteristic.
- The DIAC can be used to discover only these dedicated units in range.

Access Code Types (contd.)

- **The CAC consists of a preamble, sync word, and trailer and its total length is 72 bits.**
- **When used as self-contained messages without a header, the DAC and IAC do not include the trailer bits and are of length 68 bits.**

Access Code Format



Preamble

- **The preamble is a fixed zero-one pattern of 4 symbols used to facilitate DC compensation.**
- **The sequence is either 1010 or 0101, depending whether the LSB of the following sync word is 1 or 0, respectively.**

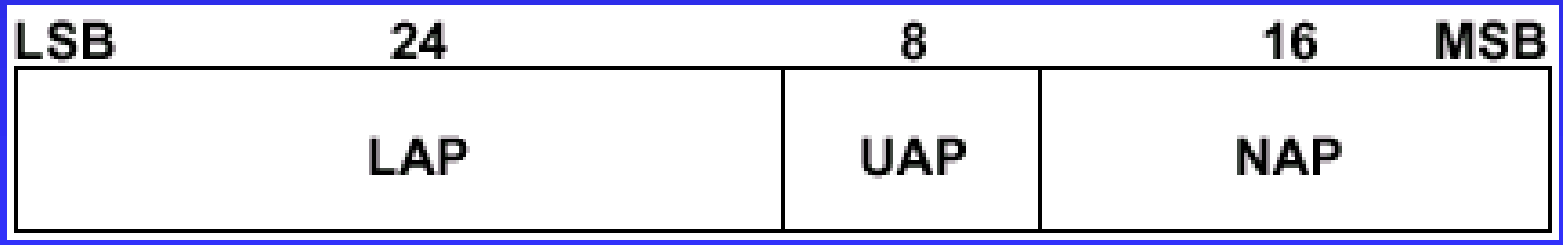
Sync Word

- The sync word is a 64-bit code word derived from a 24 bit address (LAP).

Bluetooth Device Address

Bluetooth Device Address (BD_Address)

- A 48-bit unique address is assigned to each Bluetooth device.
- LAP : Lower Address Part (24 bits)
- UAP: Upper Address Part (8 bits)
- NAP: Non-significant Address Part (16 bits)



Trailer

- **The trailer together with the three MSBs of the sync word form a 7-bit pattern of alternating 1 and 0 which may be used for extended DC compensation.**
- **The trailer sequence is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively.**

Packet Header

The header contains link control (LC) information and consists of 6 fields:

- **AM_ADDR: 3- bit *Active Member Address***
- **TYPE: 4-bit type code**
- **FLOW: 1-bit flow control**
- **ARQN: 1-bit acknowledge indication**
- **SEQN: 1-bit sequence number**
- **HEC: 8-bit header error check**

Packet Header (contd.)

- **The total header, including the HEC, consists of 18 bits, and is encoded with a rate 1/3 FEC (described later) resulting in a 54-bit header.**

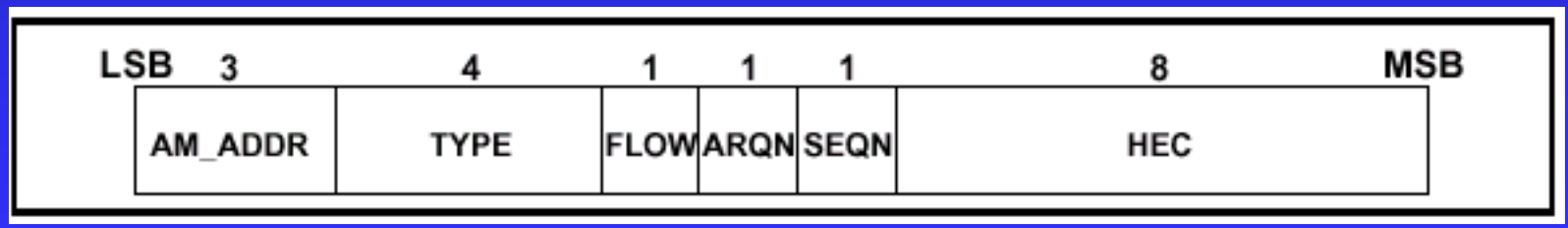
Header Format

AM-ADDR

- The AM_ADDR represents a member address and is used to distinguish between the active members participating on the piconet.
- To identify each slave separately, each slave is assigned a temporary 3-bit address to be used when it is active.
- Packets exchanged (both way) between the master and the slave all carry the AM_ADDR of this slave.
- The all-zero address is reserved for ***broadcasting*** packets from the master to the slaves.

AM-ADDR (contd.)

- An exception is the FHS packet which may use the all-zero member address but is not a broadcast message.
- Slaves that are disconnected or parked give up their AM_ADDR.
- A new AM_ADDR has to be assigned when they re-enter the piconet.



TYPE

- The 4-bit TYPE code specifies which packet type is used.
- The interpretation of the TYPE code depends on the physical link type (SCO or ACL) associated with the packet.
- TYPE code also reveals how many slots the current packet will occupy.
- This allows the non-addressed receivers to refrain from listening to the channel for the duration of the remaining slots.

FLOW

- This bit is used for flow control of packets over the ACL link.
- When the RX buffer of the recipient is full and is not emptied, a STOP indication (FLOW=0) is returned to stop the transmission of data temporarily.
- The STOP signal only concerns ACL packets.
- Packets including only link control information (ID, POLL and NULL packets) or SCO packets can still be received.
- When the RX buffer is empty, a GO indication (FLOW=1) is returned.

ARQN

- The 1-bit acknowledgment indication ARQN is used to inform the source of a transfer of payload data with CRC,
- The acknowledge can be positive or negative.
- If the reception was successful, an ACK (ARQN=1) is returned, otherwise a NAK (ARQN=0) is returned.
- When no return message regarding acknowledge is received, a NAK is assumed implicitly.

SEQN

- The SEQN bit provides a sequential numbering scheme to order the data packet stream.
- For each new transmitted packet that contains data with CRC, the SEQN bit is inverted.
- This is required to filter out retransmissions at the Destination.
- If a retransmission occurs due to a failing ACK, the destination receives the same packet twice.

SEQN (contd.)

- By comparing the SEQN of consecutive packets, correctly received retransmissions can be discarded.
- The SEQN has to be added due to a lack of packet numbering in the ***unnumbered ARQ scheme***.
- For broadcast packets, a modified sequencing method is used (explained later).

HEC

- Each header has a header-error-check to check the header integrity.

PACKET TYPES

- The packet types have been divided into four segments.
- The ***first segment*** is reserved for the four control packets common to all physical link types (SCO and ACL)
- The ***second segment*** is reserved for packets occupying a single time slot.
- The ***third segment*** is reserved for packets occupying three time slots.
- The ***fourth segment*** is reserved for packets occupying five time slots.

First Segment Packets

NULL Packet

- This packet has no payload.
- It is used to return link information to the source regarding the success of the previous transmission (ARQN), or the status of the RX buffer (FLOW).
- The NULL packet itself does not have to be acknowledged.

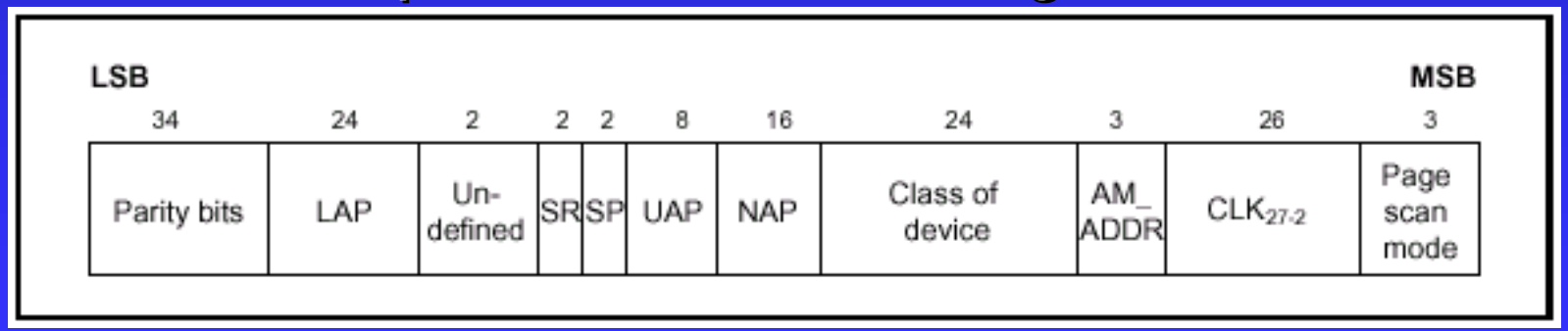
TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
0000	1	NULL	NULL
0001	1	POLL	POLL
0010	1	FHS	FHS
0011	1	DM1	DM1

POLL Packet

- The POLL packet is very similar to the NULL packet. It does not have a pay-load either.
- In contrast to the NULL packet, it requires a confirmation from the recipient.
- Upon reception of a POLL packet the slave must respond with a packet.
- This return packet is an implicit acknowledgement of the POLL packet.
- This packet can be used by the master in a piconet to poll the slaves.

FHS Packet

- The FHS packet is a special control packet revealing, among other things, the Bluetooth device address and the clock of the sender.
- The payload contains 144 information bits plus a 16-bit CRC code. The payload is coded with a rate 2/3 FEC which brings the gross payload length to 240 bits.
- The FHS packet covers a single time slot.



FHS Packet (contd.)

- The payload consists of eleven fields. The FHS packet is used in page master response, inquiry response and in master slave switch.
- In page master response or master slave switch, it is retransmitted until its reception is acknowledged or a timeout has exceeded.
- In inquiry response, the FHS packet is not acknowledged.
- The FHS packet contains real-time clock information.

FHS Packet (contd.)

- This clock information is updated before each retransmission.
- The retransmission of the FHS payload is thus somewhat different from the retransmission of ordinary data payloads where the same payload is used for each retransmission.
- The FHS packet is used for frequency hop synchronization before the piconet channel has been established, or when an existing piconet changes to a new piconet.

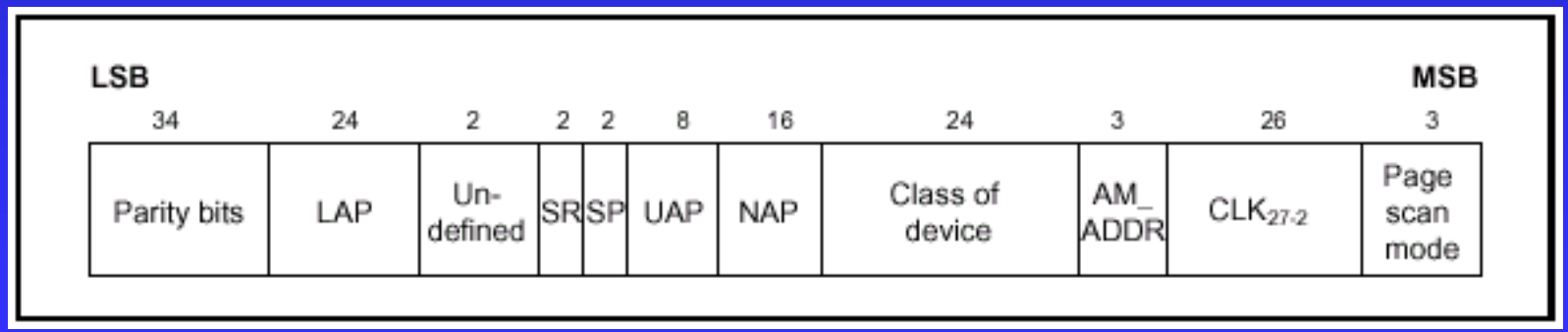
FHS Packet (contd.)

- In the former case, the recipient has not been assigned an active member address yet, in which case the AM_ADDR field in the FHS packet header is set to all-zeroes; however, the FHS packet should not be considered as a broadcast packet.
- In the latter case the slave already has an AM_ADDR in the existing piconet, which is then used in the FHS packet header.

FHS Packet (contd.)

Parity bits: This 34-bit field contains the parity bits that form the first part of the sync word of the access code of the unit that sends the FHS packet.

LAP: This 24-bit field contains the lower address part of the unit that sends the FHS packet.

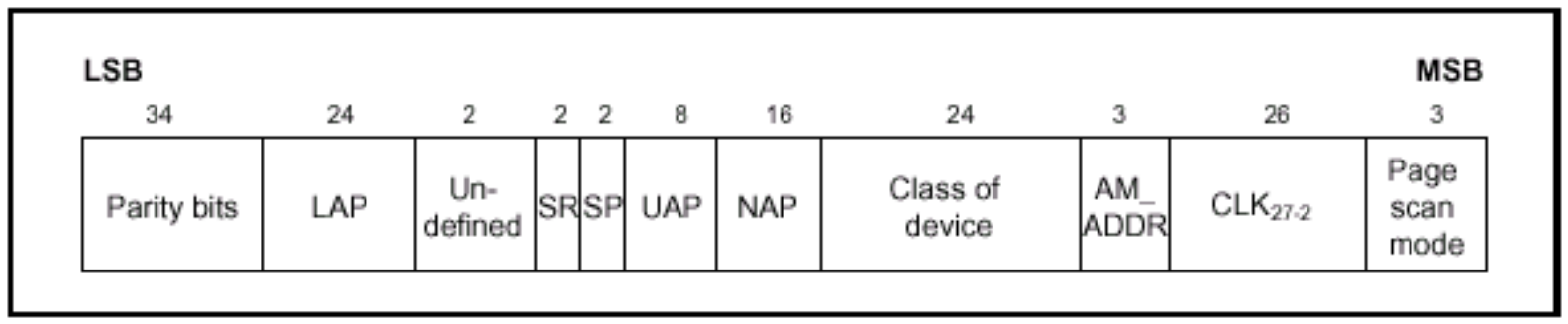


FHS Packet (contd.)

Undefined: This 2-bit field is reserved for future use and shall be set to zero.

SR: This 2-bit field indicates (scan repetition) the interval between two consecutive page scan windows.

SP: This 2-bit field indicates the period in which the mandatory page scan mode is applied after transmission of an inquiry response

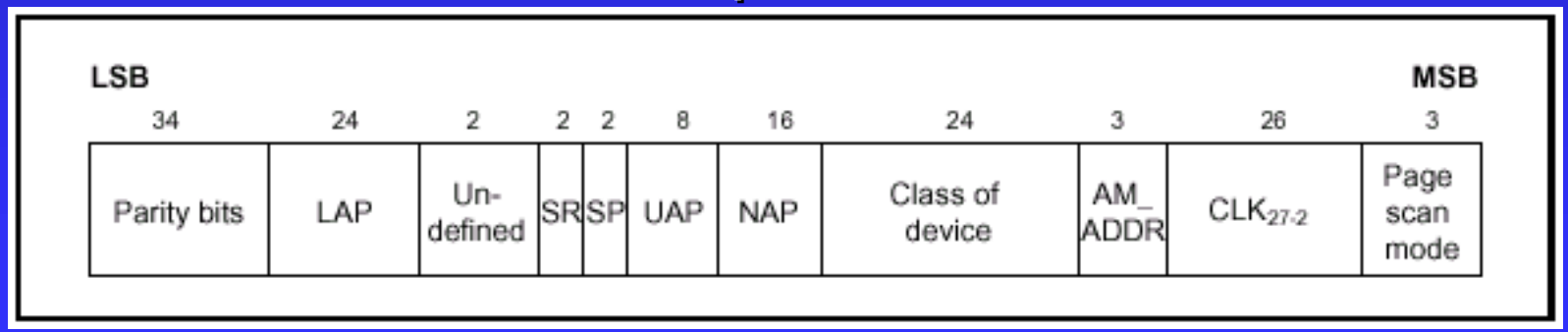


FHS Packet (contd.)

UAP: This 8-bit field contains the upper address part of the unit that sends the FHS packet.

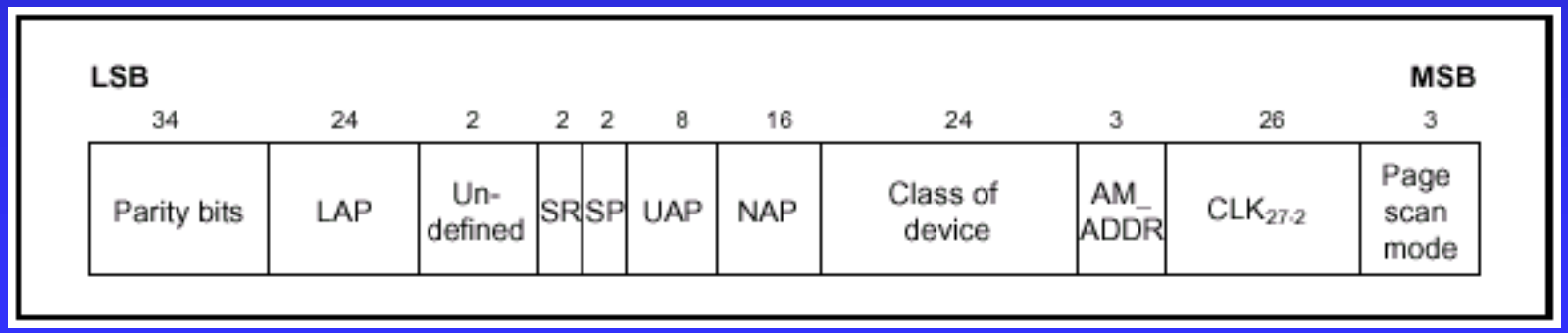
NAP: This 16-bit field contains the non-significant address part of the unit that sends the FHS packet

Class of device: This 24-bit field contains the class of device (explained later) of the unit that sends the FHS packet.



FHS Packet (contd.)

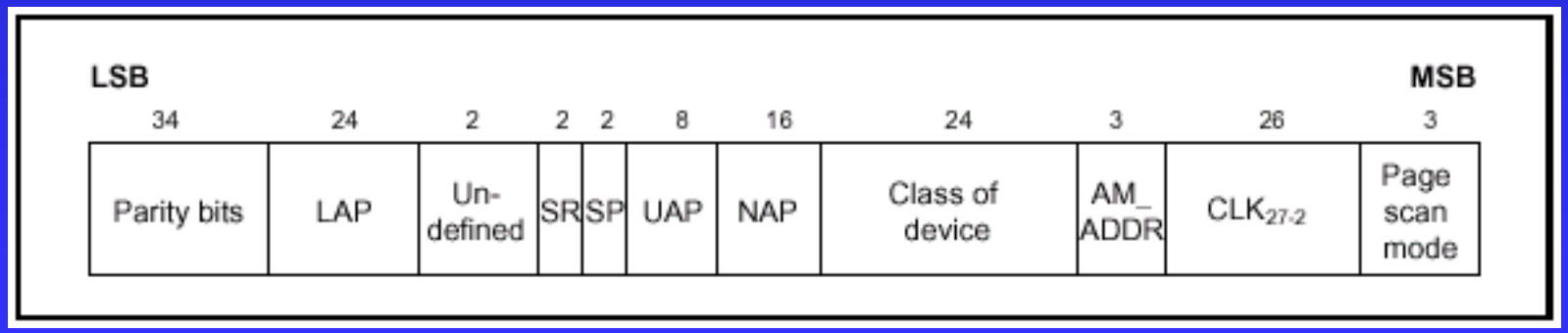
AM_ADDR: This 3-bit field contains the member address the recipient shall use if the FHS packet is used at call setup or master-slave switch. A slave responding to a master or a unit responding to an inquiry request message shall include an all-zero AM_ADDR field if it sends the FHS packet.



FHS Packet (contd.)

CLK $_{27\text{-}2}$: This 26-bit field contains the value of the native system clock of the unit that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet.

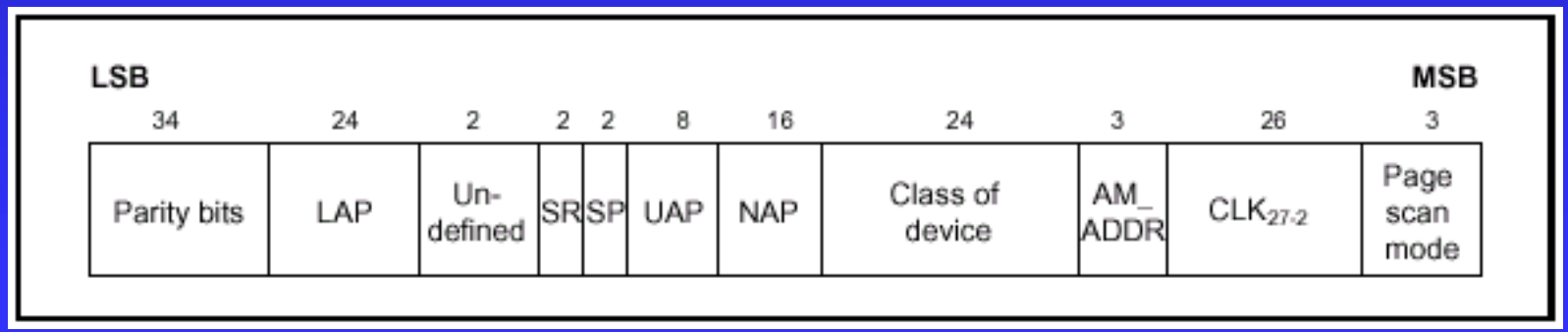
- This clock value has a resolution of 1.25ms (two-slot interval). For each new transmission, this field is updated to send new clock value.



FHS Packet (contd.)

Page scan mode: This 3-bit field indicates which scan mode is used by default by the sender of the FHS packet.

- Currently, the standard supports one mandatory scan mode and up to three optional scan modes.



DM1 Packet

- DM1 controls messages in any link type.
- It can also carry regular user data.

Segment 1 Packets

TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
0000	1	NULL	NULL
0001	1	POLL	POLL
0010	1	FHS	FHS
0011	1	DM1	DM1

Segment 2, 3 and 4 Packets

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
2	0100	1	undefined	DH1
	0101	1	HV1	undefined
	0110	1	HV2	undefined
	0111	1	HV3	undefined
	1000	1	DV	undefined
	1001	1	undefined	AUX1
3	1010	3	undefined	DM3
	1011	3	undefined	DH3
	1100	3	undefined	undefined
	1101	3	undefined	undefined
4	1110	5	undefined	DM5
	1111	5	undefined	DH5

SCO packets

- The SCO packets do not include a CRC and are never retransmitted.
- SCO packets are routed to the synchronous I/O (voice) port.
- Up to now, three pure SCO packets have been defined.
- Another SCO packet is defined which carries an asynchronous data field in addition to a synchronous (voice) field.
- The SCO packets are used for 64 kb/s speech transmission.

Types SCO Packets

- **HV1 Packet**
- **HV2 Packet**
- **HV3 Packet**
- **DV Packet**

HV1 Packet

- HV stands for ***High-quality Voice***.
- The HV1 packet carries 10 information bytes.
- The bytes are protected with a rate 1/3 FEC.
- The payload length is fixed at 240 bits.
- There is no payload header present.
- An HV1 packet can carry 1.25ms of speech at a 64 kb/s rate.
- In that case, an HV1 packet has to be sent every two time slots (T_{SCO} =2).

HV1 Packet (contd.)

Here is the Math for HV1 Packets

Bit Rate = 64 kbit/sec

Number of bits in 1.25msec of Speech

= (64000bits/sec)*(1 sec/1000 msec)*1.25msec

= 80 bits

= 10 bytes

HV2 Packet

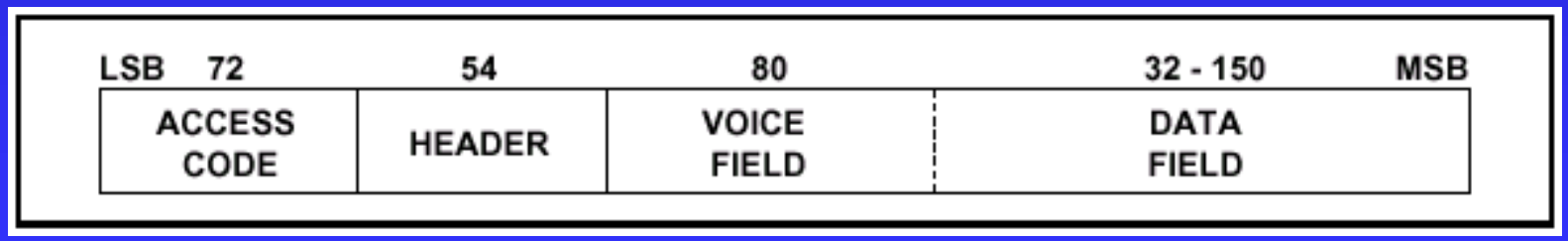
- The HV2 packet carries 20 information bytes.
- The bytes are protected with a rate 2/3 FEC.
- The payload length is fixed at 240 bits.
- There is no payload header present.
- An HV2 packet can carry 2.5ms of speech at a 64 kb/s rate.
- In that case, an HV2 packet has to be sent every four time slots ($T_{SCO} = 4$).

HV3 Packet

- The HV3 packet carries 30 information bytes.
- The bytes are not protected by FEC.
- The payload length is fixed at 240 bits.
- There is no payload header present.
- An HV3 packet can carry 3.75ms of speech at a 64 kb/s rate.
- In that case, an HV3 packet has to be sent every four time slots ($T_{SCO} = 6$).

DV Packet

- The DV packet is a combined data - voice packet.
- The payload is divided into a voice field of 80 bits and a data field containing up to 150 bits.
- The voice field is not protected by FEC. The data field contains up to 10 information bytes (including the 1-byte payload header) and includes a 16-bit CRC.



DV Packet (contd.)

- The data field is encoded with a rate 2/3 FEC.
- If necessary, extra zeroes are appended to assure that the total number of payload bits is a multiple of 10 prior to FEC encoding.
- The voice and data fields are treated completely separate.
- The voice field is handled like normal SCO data and is never retransmitted.
- The data field is checked for errors and is retransmitted if necessary.

Segment 2, 3 and 4 Packets

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
2	0100	1	undefined	DH1
	0101	1	HV1	undefined
	0110	1	HV2	undefined
	0111	1	HV3	undefined
	1000	1	DV	undefined
	1001	1	undefined	AUX1
3	1010	3	undefined	DM3
	1011	3	undefined	DH3
	1100	3	undefined	undefined
	1101	3	undefined	undefined
4	1110	5	undefined	DM5
	1111	5	undefined	DH5

ACL packets

- ACL packets are used on the asynchronous links.
- The information carried can be user data or control data.
- Including the DM1 packet, there are seven ACL packets.
- Six of the ACL packets contain a CRC code and retransmission is applied if no acknowledgement of proper reception is received.
- The 7th ACL packet, the AUX1 packet, has no CRC and is not retransmitted.

ACL packets

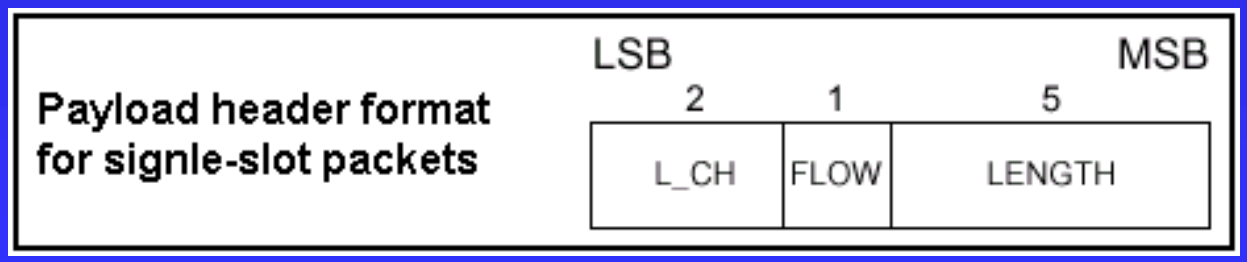
- **DM1 Packet**
- **DH1 Packet**
- **DM2 Packet**
- **DH3 Packet**
- **DM5 Packet**
- **DH5 Packet**
- **AUX1 Packet**

DM1 Packet

- DM stands for Data - Medium rate.
- The DM1 packet is a packet that carries data information only.
- The payload contains up to 18 information bytes (including the 1-byte payload header) plus a 16-bit CRC code.
- The DM1 packet may cover up to a single time slot.
- The information plus CRC bits are coded with a rate 2/3 FEC which adds 5 parity bits to every 10-bit segment.

DM1 Packet (contd.)

- If necessary, extra zeros are appended after the CRC bits to get the total number of bits (information bits, CRC bits, and tail bits) equal a multiple of 10.
- The payload header in the DM1 packet is only 1 byte long.
- The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).



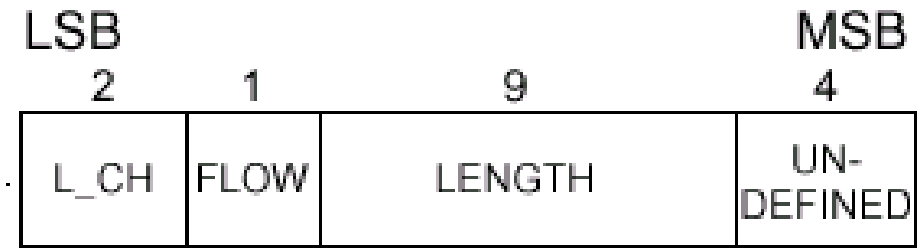
DH1 Packet

- DH stands for Data - High rate.
- This packet is similar to the DM1 packet, except that the information in the pay-load is not FEC encoded.
- As a result, the DH1 packet can carry up to 28 information bytes plus a 16-bit CRC code.
- The DH1 packet uses a single time slot.

DM3 Packet

- The DM3 packet is a DM1 packet with an extended payload.
- The DM3 packet covers up to three time slots.
- The payload contains up to 123 information bytes (including the 2-bytes payload header) plus a 16-bit CRC code.
- The pay-load header in the DM3 packet is 2 bytes long.

Payload header format for multi-slot packets



DM3 Packet

- The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).
- When a DM3 packet is sent or received, the RF hop frequency shall not change for a duration of three time slots (the first time slot being the slot where the channel access code was transmitted).

DH3 Packet

- This packet is similar to the DM3 packet, except that the information in the pay-load is not FEC encoded.
- As a result, the DH3 packet can carry up to 185 information bytes (including the two bytes payload header) plus a 16-bit CRC code.

DM5 Packet

- The DM5 packet is a DM1 packet with an extended payload.
- The DM5 packet may cover up to five time slots.
- The payload contains up to 226 information bytes (including the 2-bytes payload header) plus a 16-bit CRC code.
- The pay-load header in the DM5 packet is 2 bytes long.

DM5 Packet (contd.)

- The length indicator in the pay-load header specifies the number of user bytes (excluding payload header and the CRC code).
- When a DM5 packet is sent or received, the hop frequency shall not change for a duration of five time slots (the first time slot being the slot where the channel access code was transmitted).

DH5 Packet

- This packet is similar to the DM5 packet, except that the information in the pay-load is not FEC encoded.
- As a result, the DH5 packet can carry up to 341 information bytes (including the two bytes payload header) plus a 16-bit CRC code.

AUX1 Packet

- This packet resembles a DH1 packet but has no CRC code.
- The AUX1 packet can carry up to 30 information bytes (including the 1-byte payload header).
- The AUX1 packet covers up to a single time slot.

PAYLOAD FORMAT

- **In the payload, two fields are distinguished: the (synchronous) voice field and the (asynchronous) data field.**
- **The ACL packets only have the data field.**
- **The SCO packets only have the voice field - with the exception of the DV packets which have both.**

Voice field

- **The voice field has a fixed length.**
- **For the HV packets, the voice field length is 240 bits.**
- **For the DV packet the voice field length is 80 bits.**
- **No payload header is present.**

Data field

- The data field consists of three segments:
  - a payload header,
  - a payload body,
  - a CRC code
- Only the AUX1 packet does not carry a CRC code.

Data Field Format

Payload Header	Payload Body	CRC

Payload header (contd.)

- The payload header is one or two bytes long.
- Packets in segments one and two have a 1-byte payload header.
- Packets in segments three and four have a 2-bytes payload header.

Payload header format for single-slot packets

L_CH 2 bits	FLOW 1 bit	Length 5 bits

Payload header format for multi-slot packets

L_CH 2 bits	FLOW 1 bit	Length 9 bits	Undefined 4 bits

Fields of Payload header

- **L_CH Field**: It specifies a logical channel.
- **FLOW Field**: It controls the flow on the logical channels.

  FLOW = 1 means “OK to send”

  FLOW = 0 means “stop sending”

Meaning of L_CH bits

L_CH code b_1b_0	Logical Channel	Information
00	NA	undefined
01	UA/UI	Continuation fragment of an L2CAP message
10	UA/UI	Start of an L2CAP message or no fragmentation
11	LM	LMP message

Fields of Payload header (contd.)

- **LENGTH**: It indicates the length of the payload.

  In the case of a 2-byte payload header, the 4-bit Undefined Field is reserved for future use and shall be set to zero.

Payload header format for multi-slot packets

L_CH 2 bits	FLOW 1 bit	Length 9 bits	Undefined 4 bits

PACKET SUMMARY

Link Control Packets

Type	User Payload (bytes)	FEC	CRC
ID	na	na	na
NULL	na	na	na
POLL	na	na	na
FHS	18	2/3	yes

PACKET SUMMARY (contd.)

SCO Packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC
HV1	na	10	1/3	no
HV2	na	20	2/3	no
HV3	na	30	no	no
DV*	1 D	10+(0-9) D	2/3 D	yes D

*. Items followed by 'D' relate to data field only.

PACKET SUMMARY (contd.)

ACL Packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC
DM1	1	0-17	2/3	yes
DH1	1	0-27	no	yes
DM3	2	0-121	2/3	yes
DH3	2	0-183	no	yes
DM5	2	0-224	2/3	yes
DH5	2	0-339	no	yes
AUX1	1	0-29	no	no

5. ERROR CORRECTION

- There are three error correction schemes defined for Bluetooth:
  - 1/3 rate FEC
  - 2/3 rate FEC
  - ARQ scheme for the data
- The purpose of the FEC scheme on the data payload is to reduce the number of retransmissions.
- However, in a reasonable error-free environment, FEC gives unnecessary overhead that reduces the throughput.

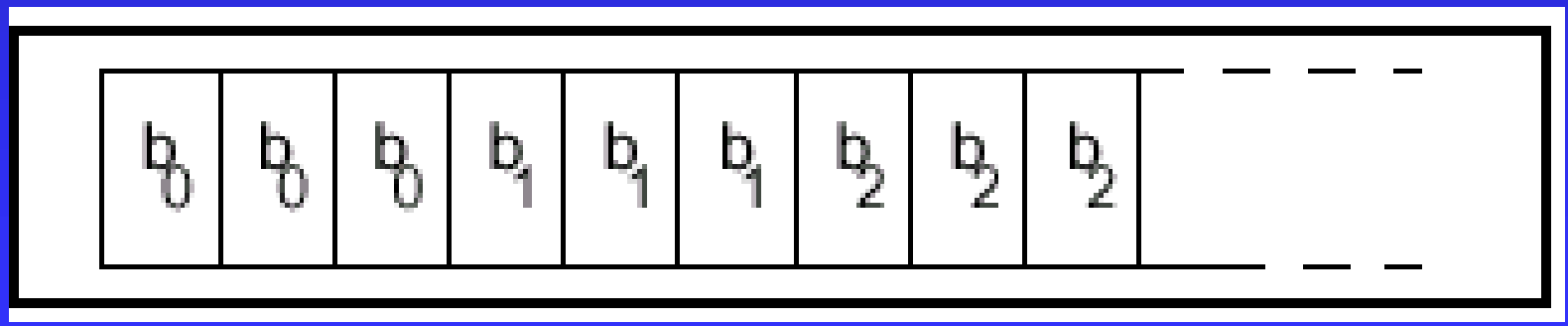
ERROR CORRECTION (contd.)

- Therefore, the packet definitions have been kept flexible to use or not to use FEC in the payload.
- The packet header is always protected by a 1/3 rate FEC.

FEC CODE: RATE 1/3

- A simple 3-times repetition FEC code is used for the header.
- The repetition code is implemented by repeating the bit three times.
- Length of the actual information is 1/3 of the length of the coded message.

Bit-repetition encoding scheme for $b_0 b_1 b_2$



FEC CODE: RATE 2/3

- In this coding each block of 10 information bits is encoded into a 15-bit codeword by appending 5 parity bits.
- Thus, the length of information bits is 2/3 the length of the codeword.
- This code can correct all single errors and detect all double errors in each codeword.

6. LOGICAL CHANNELS

In the Bluetooth system, five logical channels are defined:

- LC control channel
- LM control channel
- UA (asynchronous) user channel
- UI (isochronous) user channel
- US (synchronous) user channel

The control channels LC and LM are used at the link control level and link manager level, respectively.

LOGICAL CHANNELS (contd.)

- The LC channel is carried in the packet header.
- The LM, UA, and UI channels are indicated in the L_CH field in the payload header.
- The US channel is carried by the SCO link only.
- The UA and UI channels are normally carried by the ACL link, and also by the data in the DV packet.

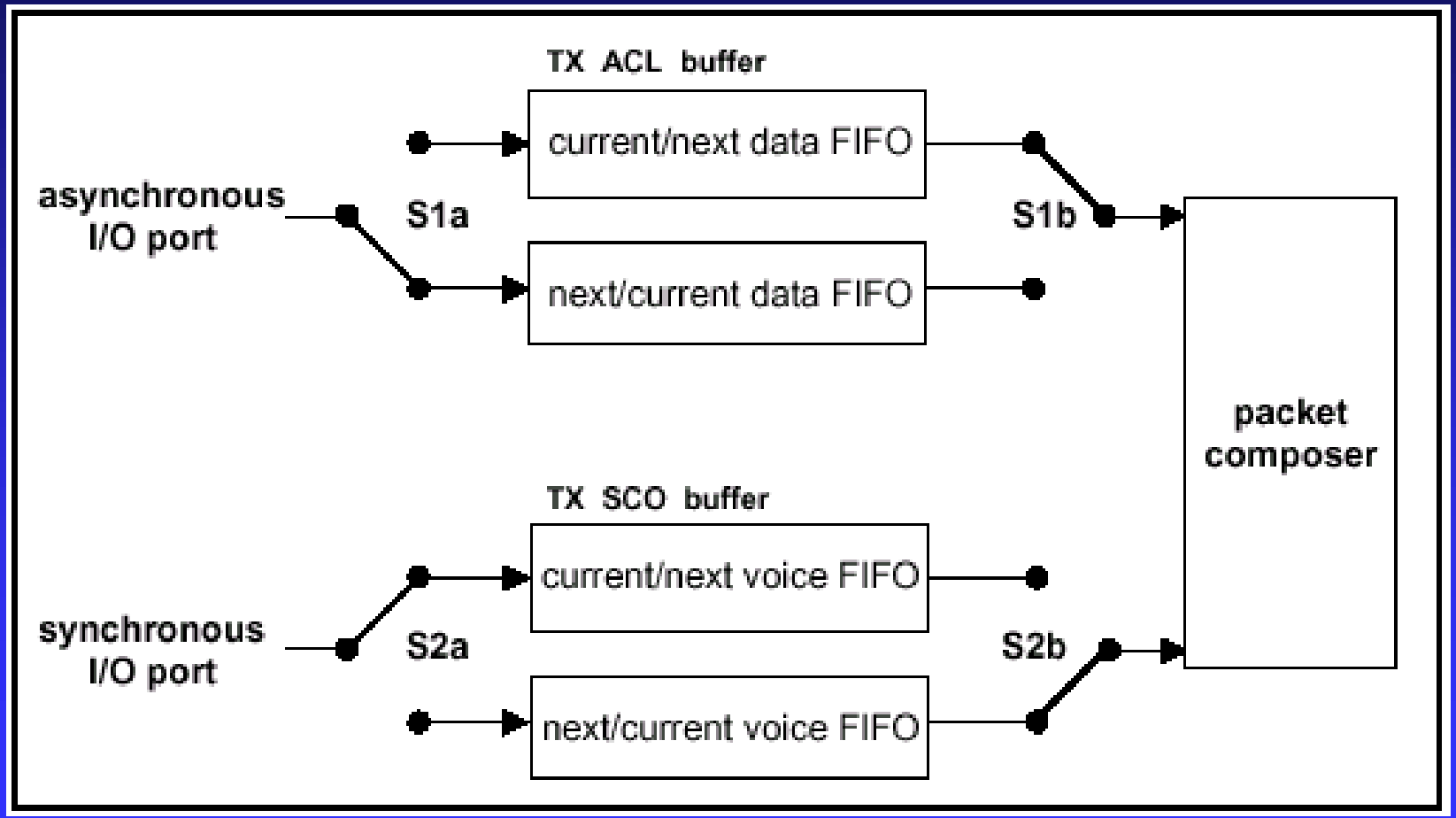
7. DATA WHITENING

- Before transmission, both the header and the payload are scrambled with a data whitening word in order to randomize the data from highly redundant patterns and to minimize DC bias in the packet.
- The scrambling is performed prior to the FEC encoding.
- At the receiver, the received data is descrambled using the same whitening word generated in the recipient.

8. Transmit/Receive Buffers

I/O Ports → Link Manager → One Tx Buffer

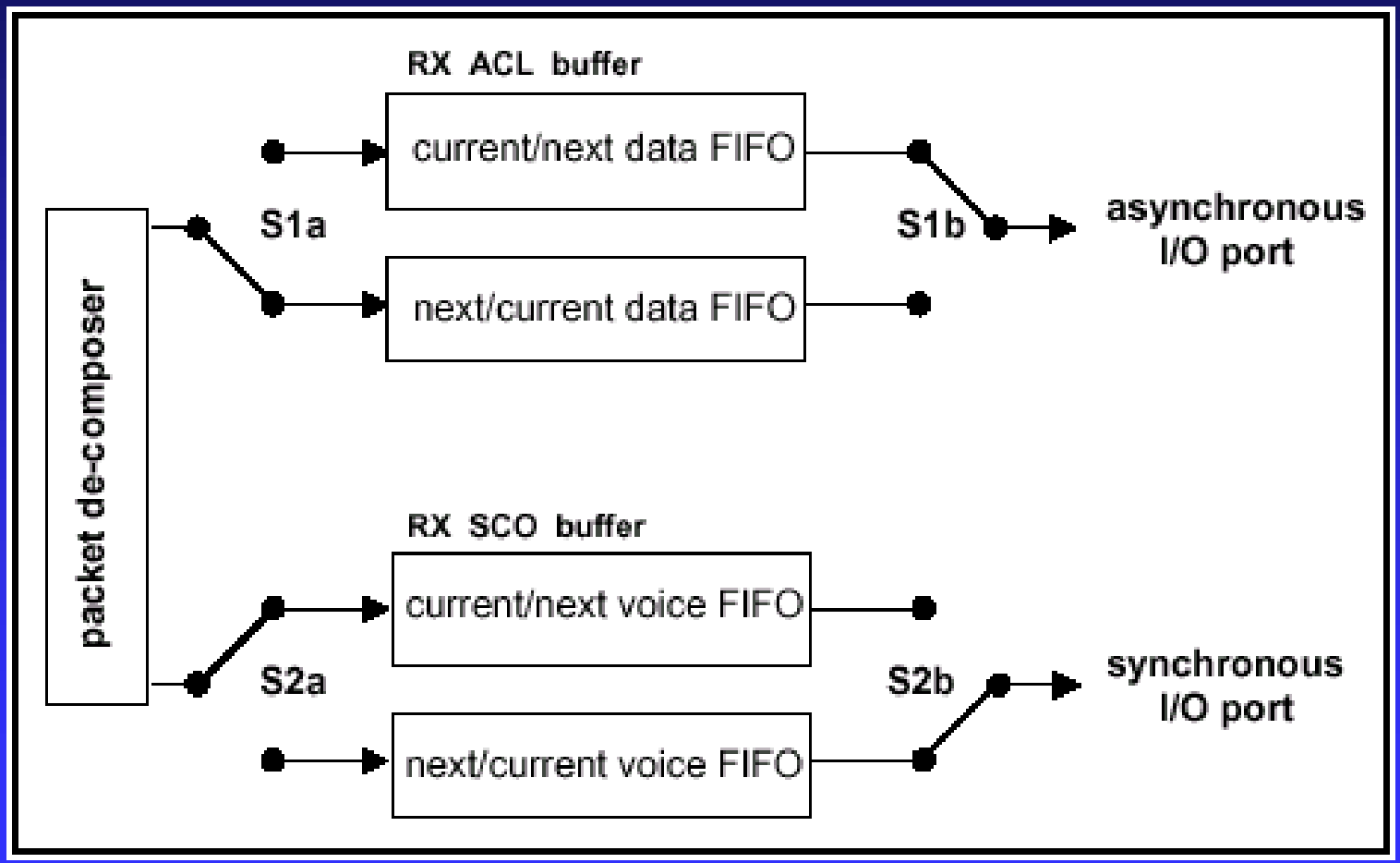
Another Tx buff. → BT Controller → Packets



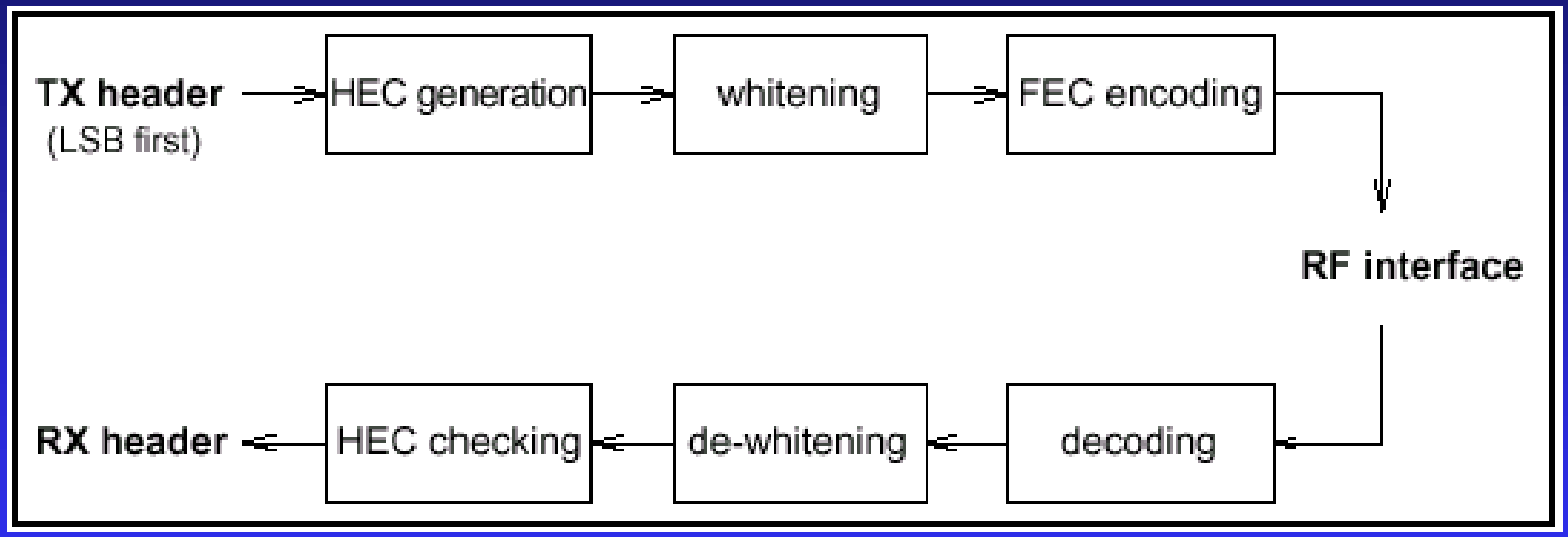
Trans./Rec. Buffers (contd.)

Current Payload → BT Controller → One RX buffer

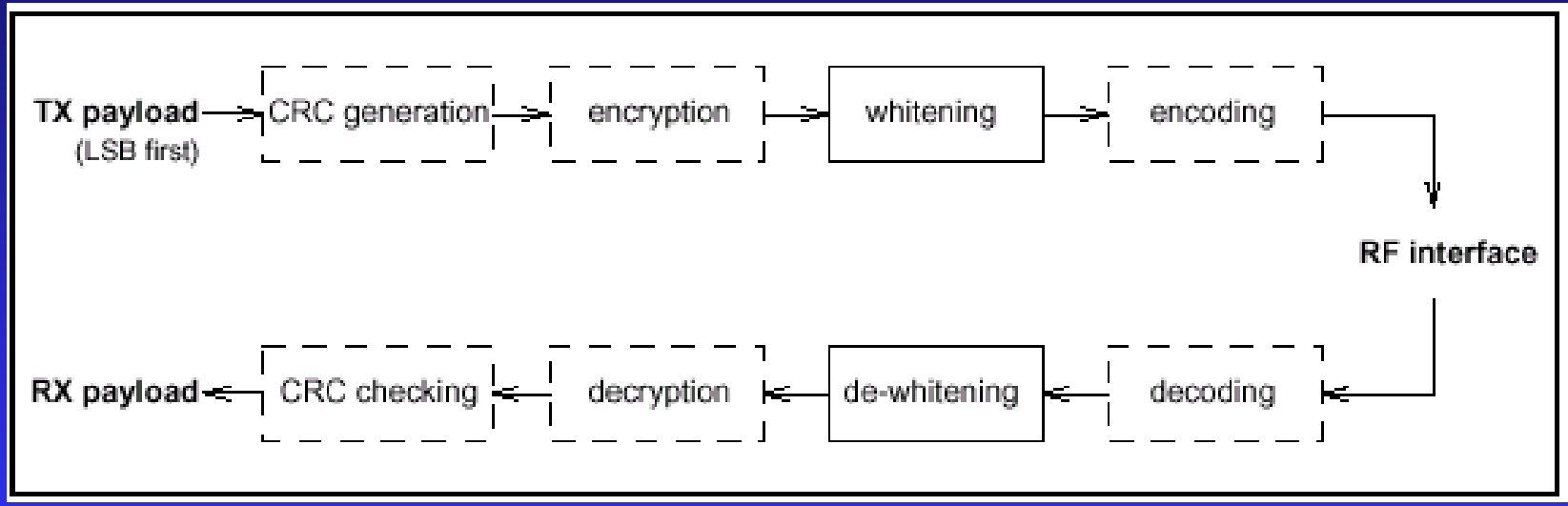
Another Rx Buff. → Link Manager → I/O Ports



Header Bit Processes



Payload Bit Processes



Applications

- **Remote networking using a Bluetooth cellular phone.**
- **Speakerphone applications using a Bluetooth cellular phone**
- **Business card exchange between Bluetooth notebooks, handhelds, and phones.**
- **Calendar synchronisation between Bluetooth notebooks, handhelds, and phones.**

Applications

- Wireless hands-free operation using a Bluetooth headset.
- Cable-free remote networking with a Bluetooth notebook or handheld computer.
- Automatic address book synchronisation with trusted Bluetooth notebooks or handheld computers.

Bluetooth Products

- Bluetooth PCMCIA Card
- Bluetooth Platform solution
- Bluetooth Enabled Cell phone

I-BAEN versus BLUETOOTH

- The Bluetooth standard uses the 2.4GHz license-free ISM band, and it implements the frequency-hopping spread spectrum technique for multiple access communication.

- The i-Bean uses 916MHz in the U.S., 434MHz in Europe and 429.5MHz in Japan, which are all license-free ISM bands in those areas.

I-BAEN versus BLUETOOTH

- The i-Bean doesn’t rely on any spread spectrum techniques for multiple access communication; instead, it uses a proprietary protocol to achieve robust and power efficient network communication.
- The 2.4GHz carrier frequency used by Bluetooth is substantially higher than that of the i-Bean and the processing tasks involved in the Bluetooth hardware are much more complex than the i-Bean design.

I-BAEN versus BLUETOOTH

- **Bluetooth chip set (radio transceiver and baseband controller) on the average draws 30 – 40 milli-ampere of current, which is considerably higher than the 1 milli-ampere current consumption of the i-Bean.**

- **The main advantage of Bluetooth over the i-Bean is in data rate: 721 Kbits/sec for Bluetooth vs. 115 Kbits/sec for i-Bean. However, i-Bean's 115 Kbits/sec data rate is comparable to that of a PC's RS232 port, and is sufficiently fast to support a large number of wireless applications.**

I-BAEN Specification

Function/Feature	Specifications
Wireless Communication	Yes
Communication Distance	Up to 100 ft (30 m)
Frequency Band	Selectable from 300 MHz to 900 MHz
Maximum Bandwidth	115kbits/sec
Power Supply Type	Battery, 3VDC
Supply Current	< 2 mA
Network Mode	Point-to-point; point-to-multipoint; multipoint-to-multipoint

I-BAEN Specification

Function/Feature	Specifications
Flash Memory	8 KB
A/D Function	8 channels with 8 bits resolution
D/A Function	2 channels with 8 bits resolution
Digital I/O	8 channels
PWM Signal	2 channels
Sampling Frequency	50 Hz max.
Dimensions	0.93 in. X 1.00 in. X 0.20 in

I-BEAN Applications

- Applications for i-Beans span a broad range including manufacturing, home automation, construction, consumer goods, appliances and security.
- i-Beans are ideal for a variety of short-range data acquisition, monitoring and control applications.
- An i-Bean can be connected to a low-level signal source (e.g. a thermocouple or a strain gauge), embedded in a hand held or wearable device or programmed to command another device.

I-BEAN Applications

- i-Beans have immediate utility in “personal area networking”, wearable computing, security, equipment monitoring and other close range wireless applications in the consumer, industrial, medical and military markets.

- For example, with a simple touch of a button, a truck driver with a mobile phone or a PDA can monitor the temperature of fresh food packages carried in the trailer at any time.

I-BEAN Applications

- A parking meter attendant or utility meter reader can drive through an entire street without stopping while inspecting all meters with a hand held device.
- A foreman can wear a hardhat embedded with an i-Bean to collect critical data from instruments and equipment scattered throughout the construction site.
- An air conditioner can deliver air to a room based on the temperature sensed by an i-Bean where the users actually are, not by the wall-mounted thermostat 20 ft away.

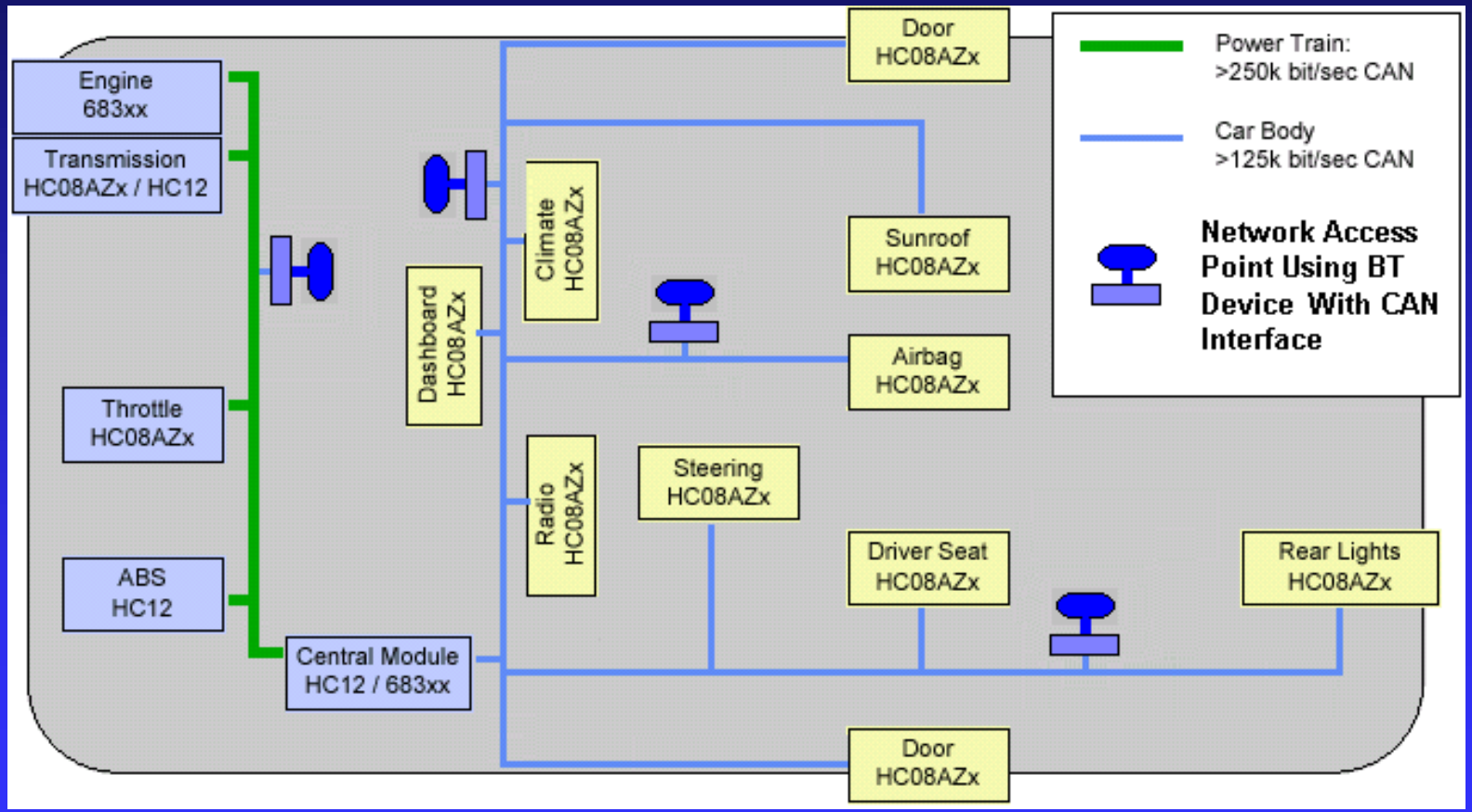
I-BEAN Applications

Other examples of potential applications include:

- Ambulatory health monitoring / Medical care;
- Remote sensing and metering;
- Building and environment monitoring;
- Industrial equipment monitoring;
- Distribution and retail systems management;
- Quality Control;
- Supply Chain Management;
- Home and factory automation; and,
- Wearable computers.

Hybrid Network

Original Source of the Figure is Motorola



Hybrid Network (contd.)

- The BT device at each Network Access Point could work in a different piconet in order to have the maximum throughput for the wireless network.
- The portable devices in the wireless network in a vehicle could be a laptop, cell phone, equipments for monitoring different operations of the vehicle, etc.
- Some of the monitoring parameters could be engine temperature, vehicle speed, emission, tire pressure, road condition, etc.

Hybrid Network (contd.)

- If the portable devices work in different piconets and all of them together need significantly high bandwidth from the wired network, then an optical bus could be used instead of a dual-wire network like CAN.