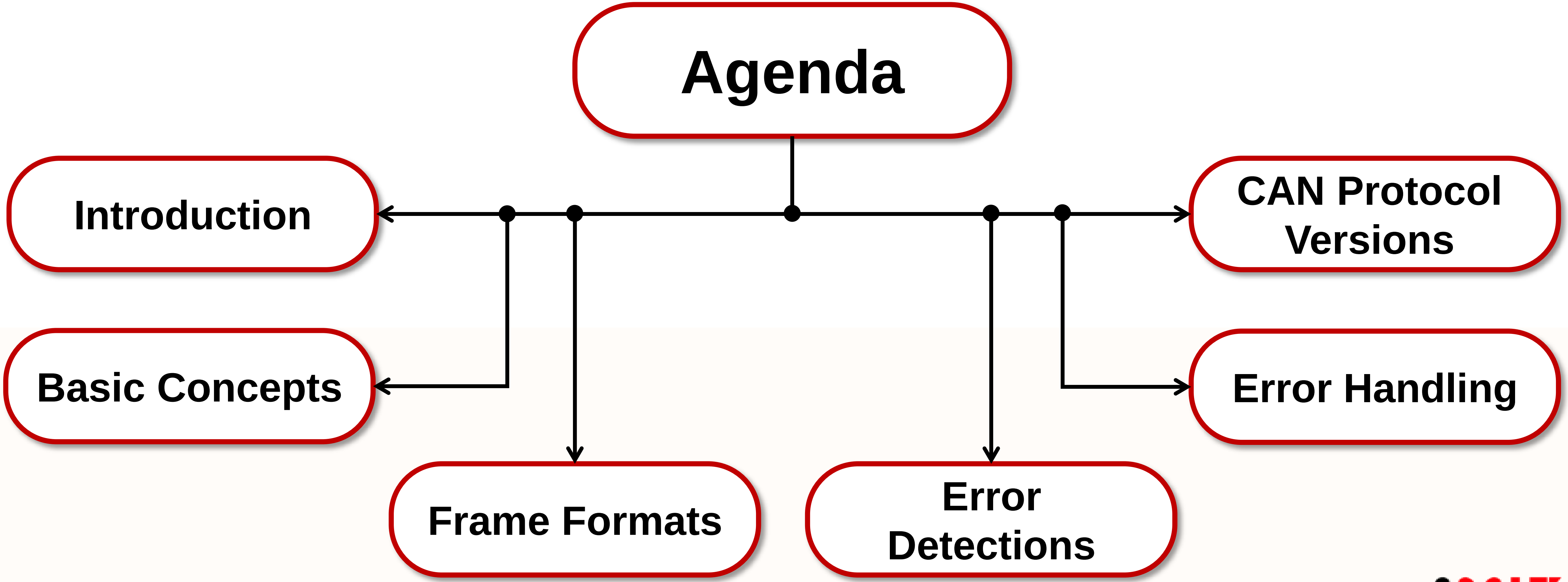
Learn Automotive Bus Technology

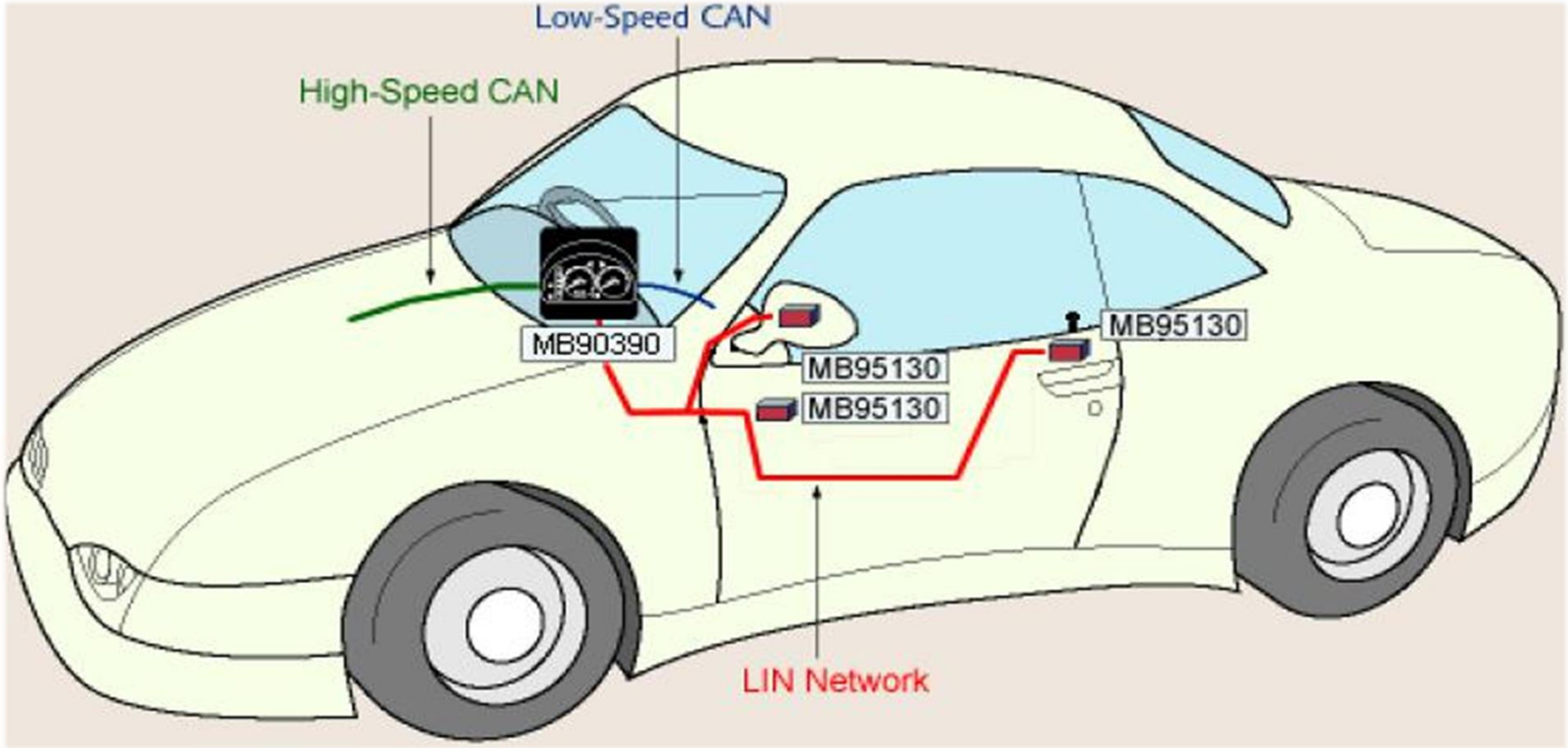
Prepared By: AMIT Learning

Agenda


Agenda
Introduction
Basic Concepts
Frame Formats
Error Detections
Error Handling
CAN Protocol Versions

➢ Introduction

- **How it began ...**



Introduction

Each type requires specific features:

- Conventional body and powertrain applications use protocols with known real-time properties
  - **CAN**

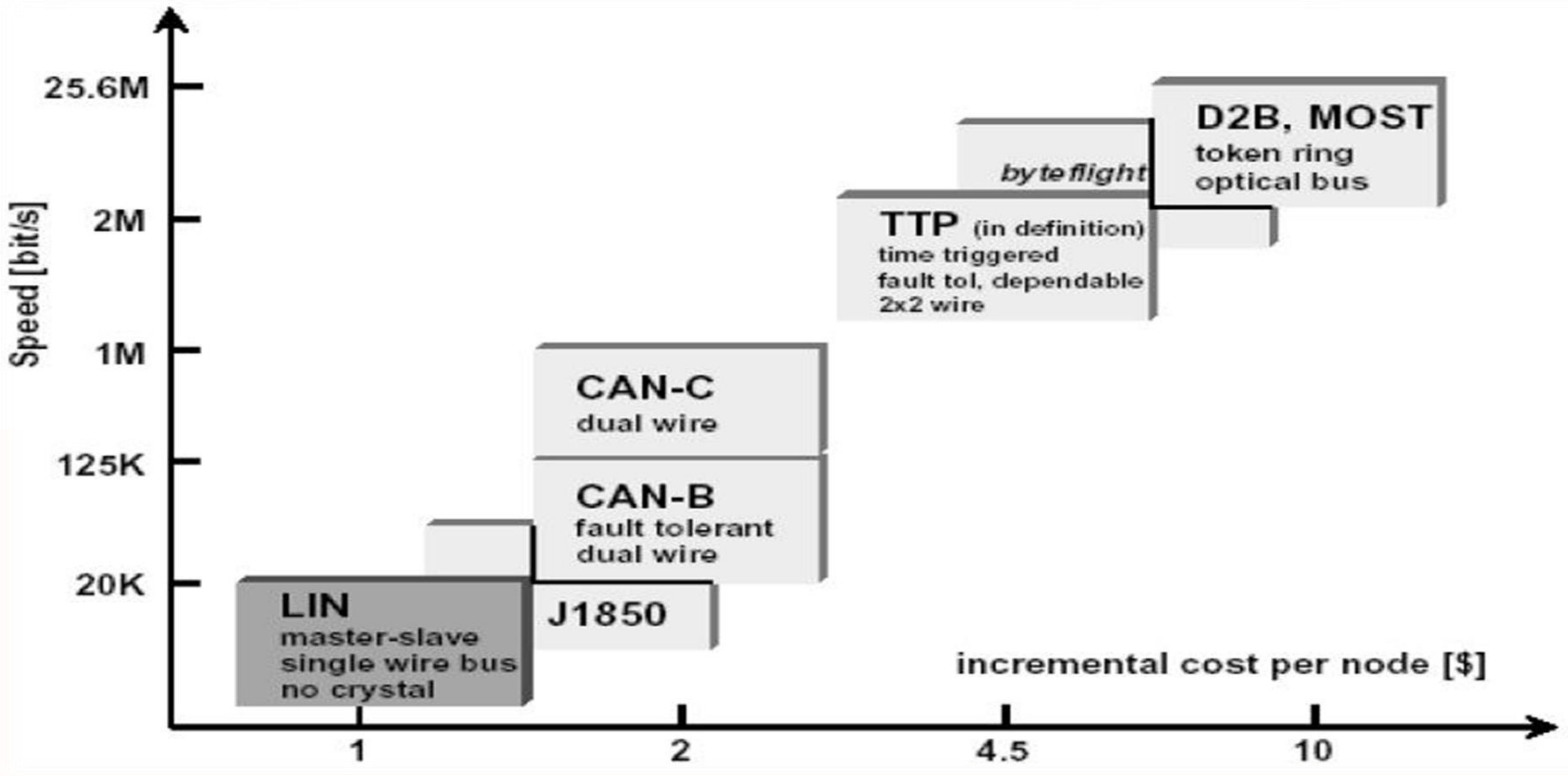
- Multimedia applications, calling for protocols that should provide high bandwidth and speed and even wireless interconnection.
  - **Bluetooth**
  - **MOST**
  - **Firewire**

➢ Introduction

- Safety critical applications, needing protocols that are fault tolerant and reliable. X-by-wire is an emerging market that calls for protocols like
  - **TPIC (Time-Triggered Protocol classified as a SAE type C network)**
  - **FlexRay**
  - **TT-CAN (Time Triggered CAN).**

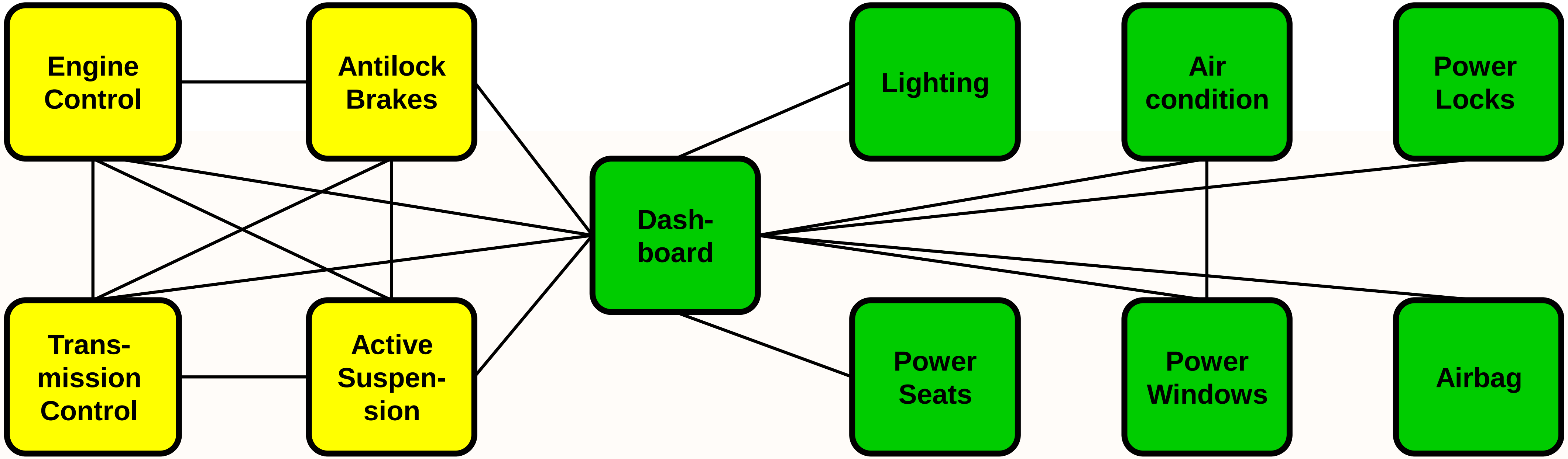
- Mechatronic type applications such as smart sensors and actuators, or even complex ECUs with simple communication needs. These applications are addressed by protocols like
  - **LIN**
  - **TTP/A**
  - **other OEM (Original Equipment Manufacturer) specific protocols**.

➢ Introduction



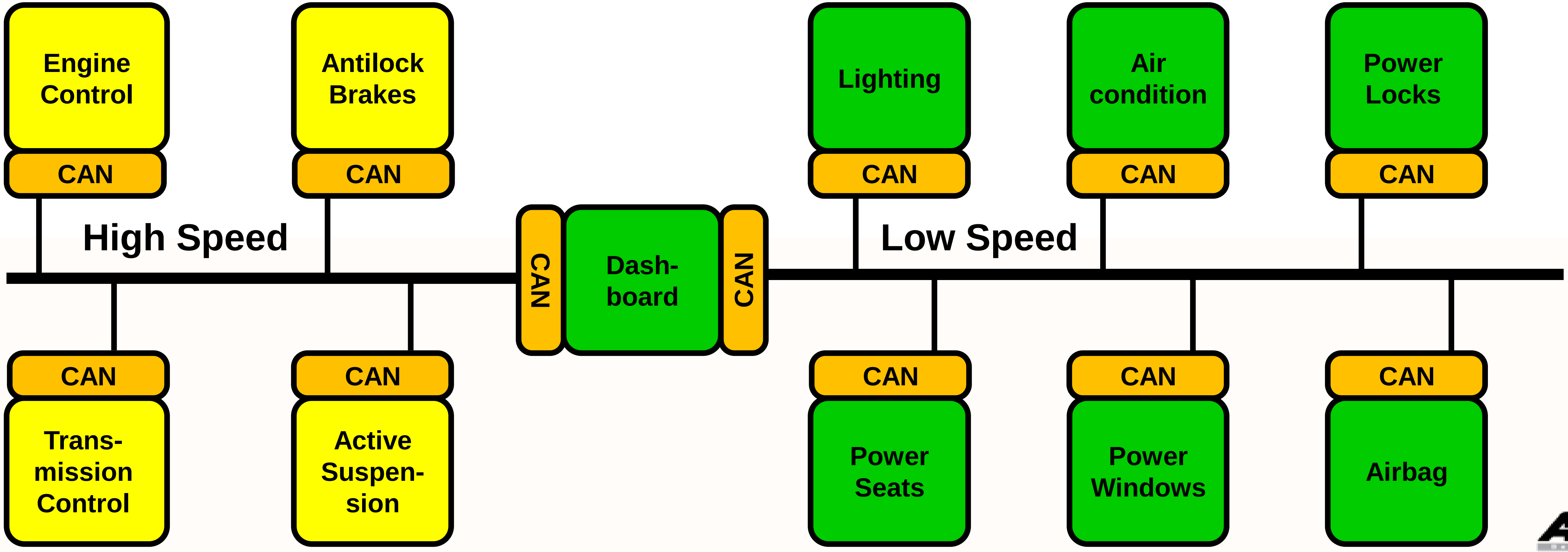
Introduction

- **How it all began ...**
  - Discrete Interconnection of different systems (point to point wiring)
  - Network cable with a length of up to several miles and many connectors was required !!

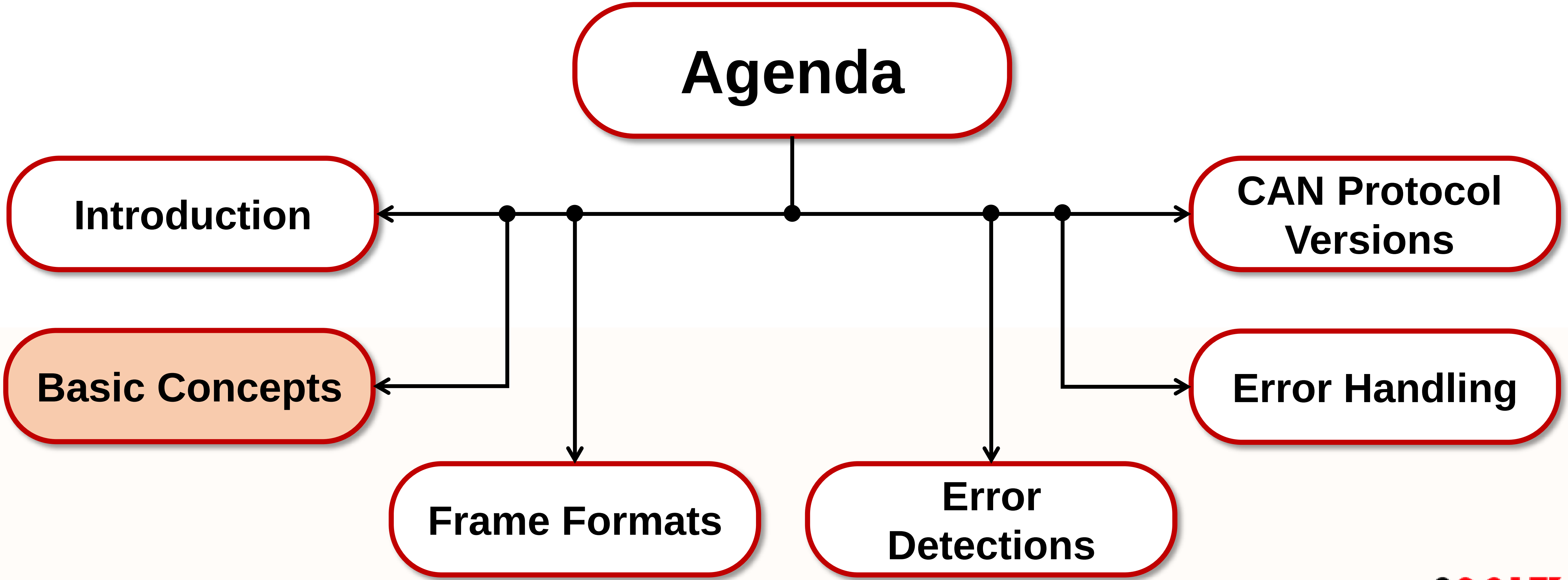


Introduction

How it all began ...



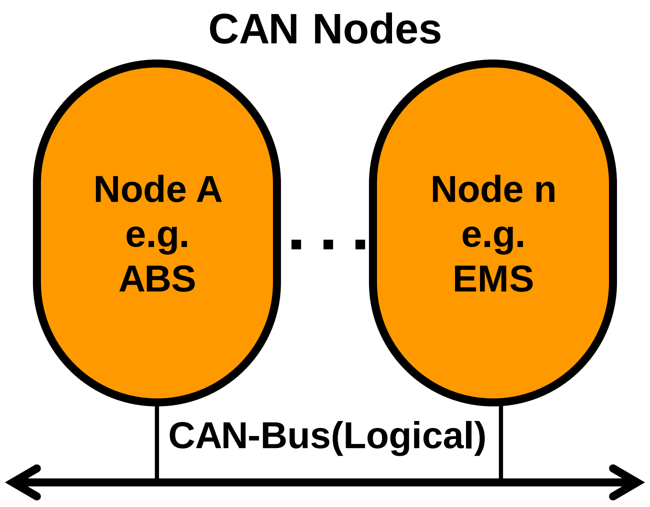
Agenda


Agenda
Introduction
CAN Protocol Versions
Basic Concepts
Error Handling
Frame Formats
Error Detections

➢ Basic Concepts

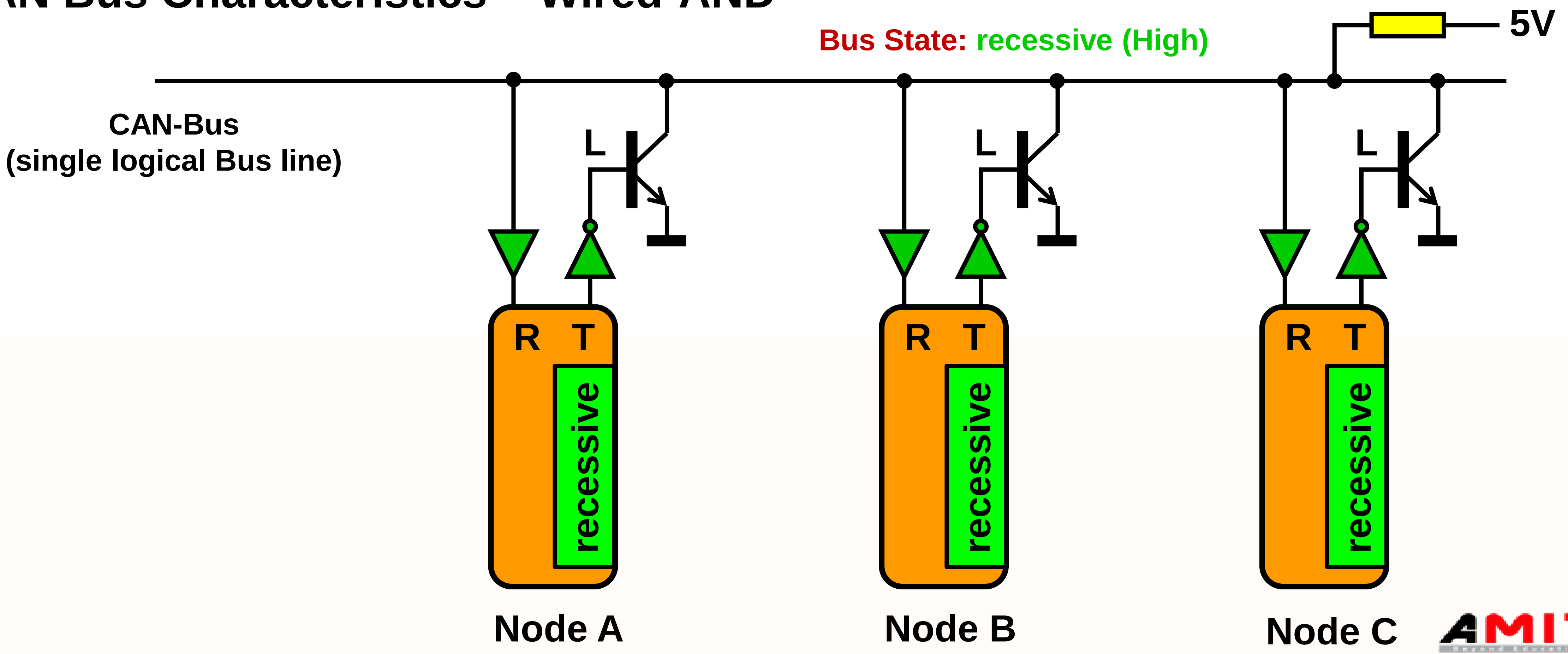
- Multi-Master Concept.
- Number of Nodes is not limited by protocol
- No Node addressing
  - **Message ID specifies contents and priority.**

- Easy connection/disconnection of nodes.
- Broadcast/Multicast capability.
- CAN Network Speed:
  - **LOW Speed CAN: baud rates from 40Kbtis/s to 125Kbits/sec**
  - **High Speed CAN: baud rates from 40Kbtis/s to 1Mbits/sec, depending on Cable length**



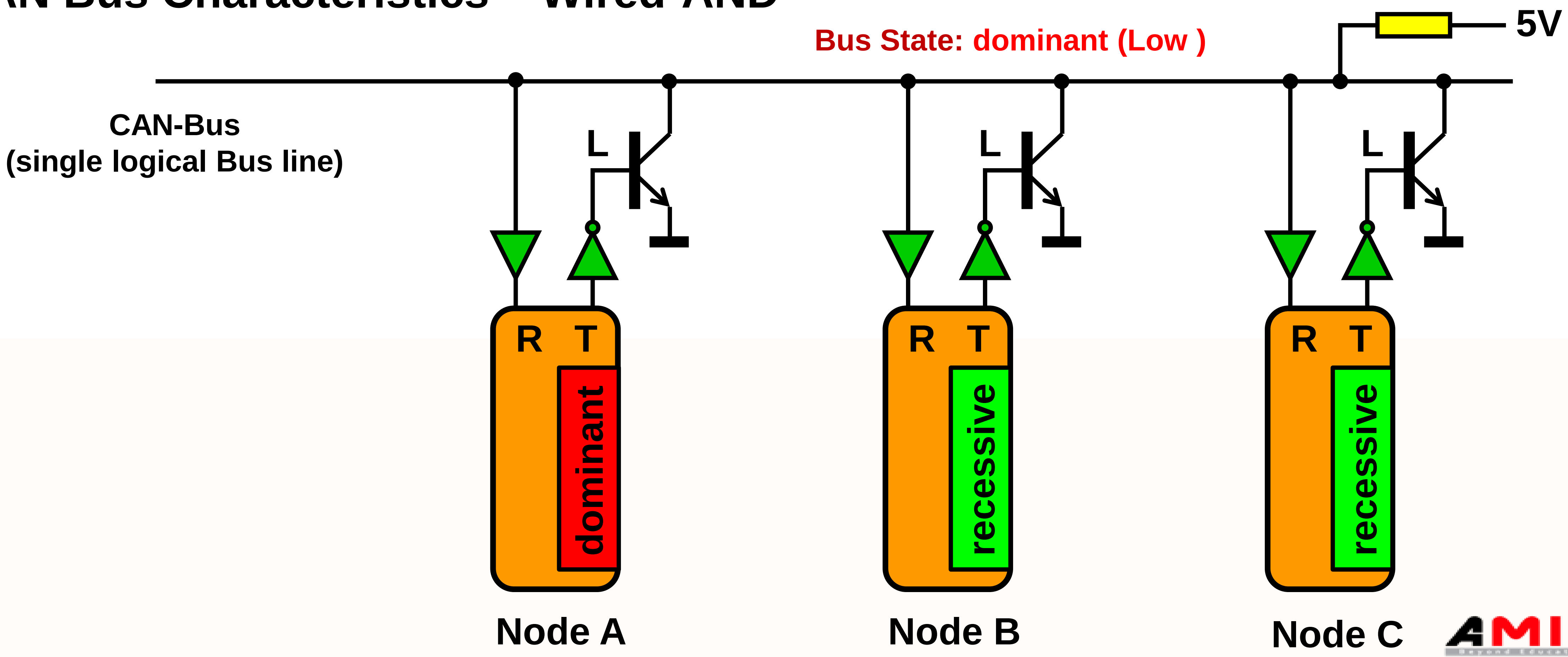
➢ Basic Concepts

➢ CAN Bus Characteristics – Wired-AND



➢ Basic Concepts

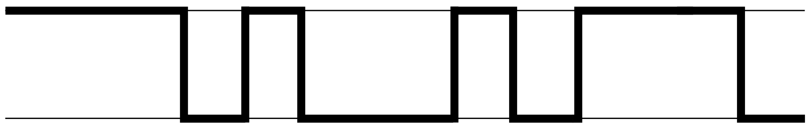
➢ CAN Bus Characteristics – Wired-AND



Basic Concepts

CAN Bus Characteristics – Wired-AND

Recessive Vs Dominant



“1” = recessive

“0” = dominant

Two logic states possible on The bus.

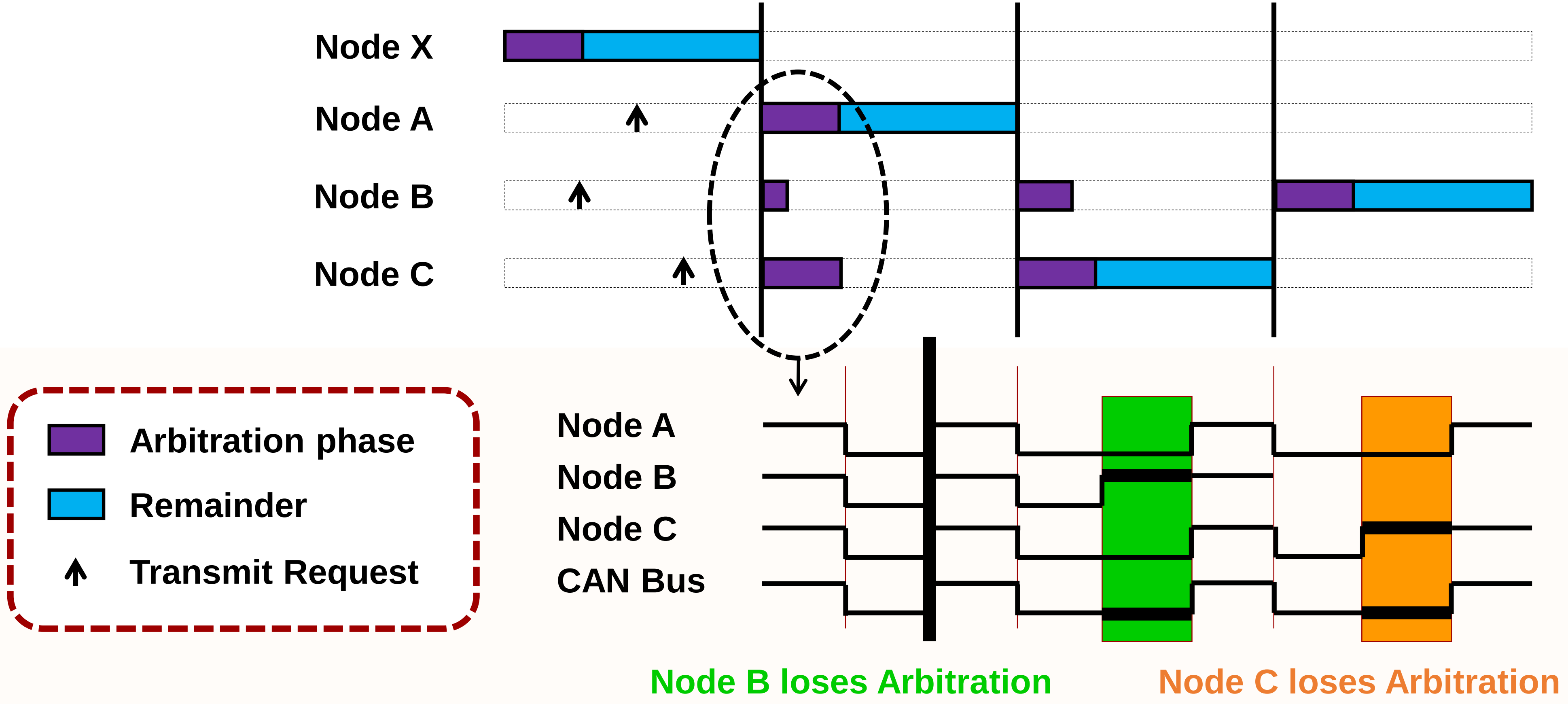
A	B	C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
R	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

As soon as one node nodes transmits
a dominant bit (zero):
Bus is in the dominant state

Only if all nodes transmit
recessive bit (ones):
Bus is in the recessive state

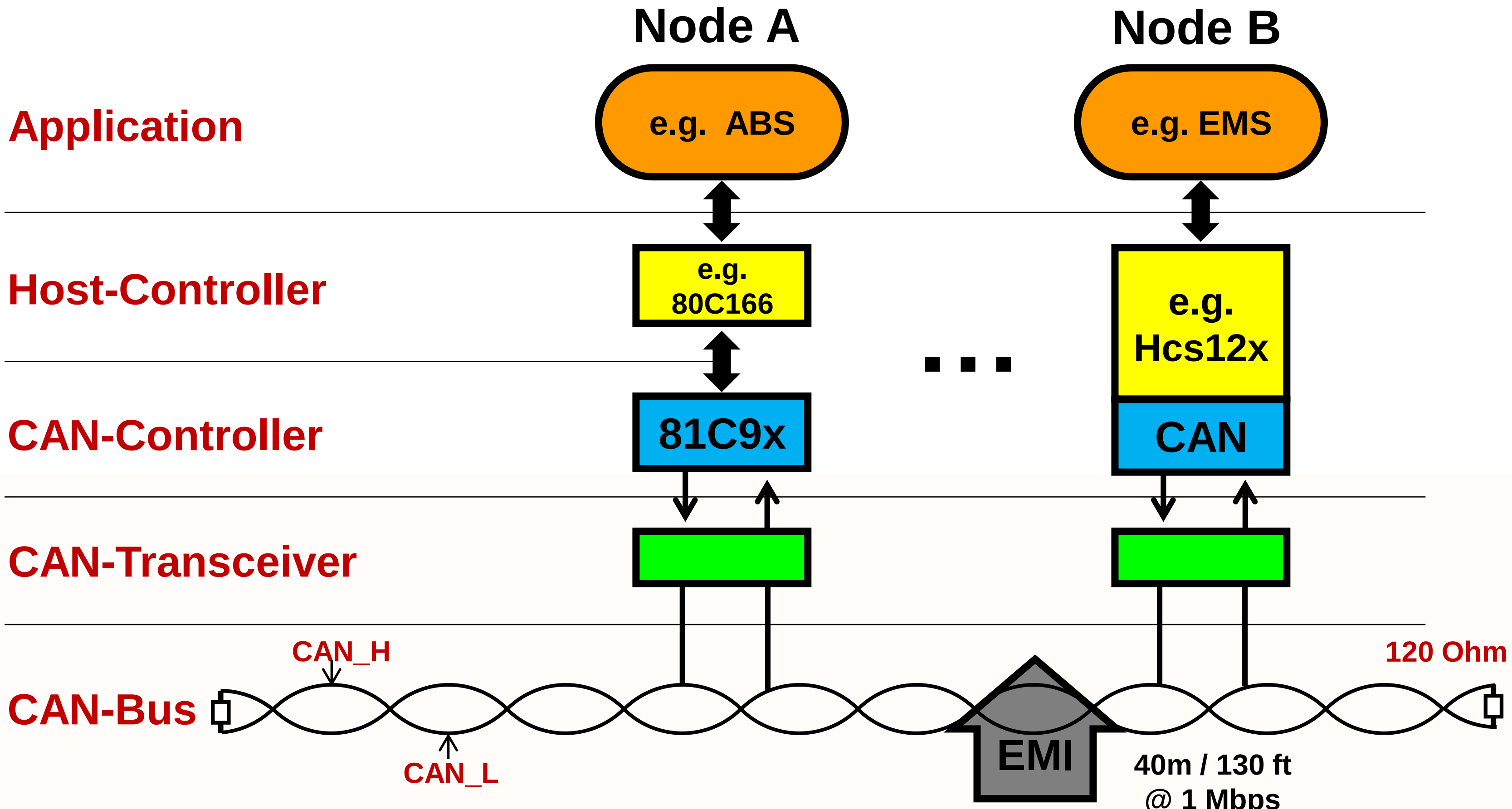
Basic Concepts

Bus Access and Arbitration



➢ Basic Concepts

- **Sophisticated Error Detection/ Handling**
- **NRZ and Bit Stuffing for Synchronization**
- **Bus Access via CSMA/CD**

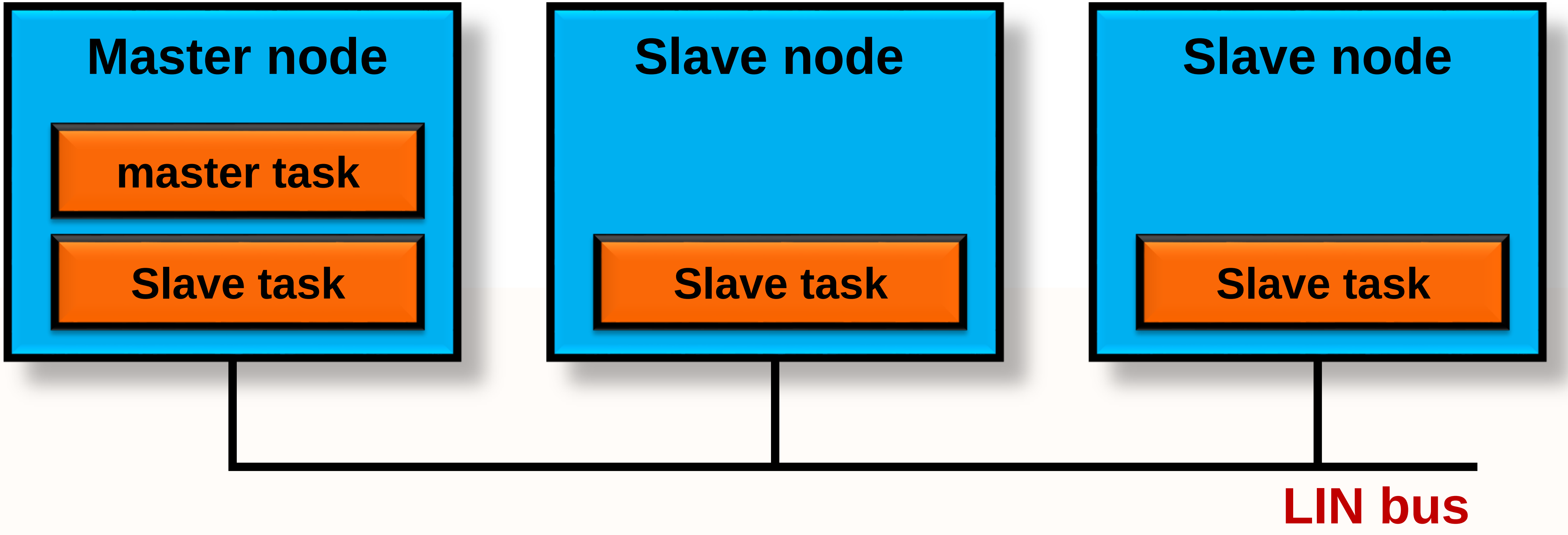


CSMA/CD → Carrier Sense Multiple Access / Collision Detection

Basic Concepts

- The LIN is a SCI/UART-based serial
- Single-Master / Multi Slave Concept.
- Number of Nodes is limited up to 16 slaves
- No Node addressing
  - **Message ID specifies Contents and priority**
- Broadcast / Multicast capability.
- LIN Network Speed**.**
  - **LIN baud rates up to 20 Kbit/s**

➢ Basic Concepts



Basic Concepts

- LIN is a Time Triggered communication protocol designed to support automotive networks in conjunction with Controller Area Network (CAN)

- No collision detection exists in LIN, therefore all messages are initiated by the master with at most one slave replying for a given message identifier

➢ Basic Concepts

- The LIN bus is connected between smart sensor or actuators and an Electronic Control Unit (ECU) which is often a gateway With CAN bus.

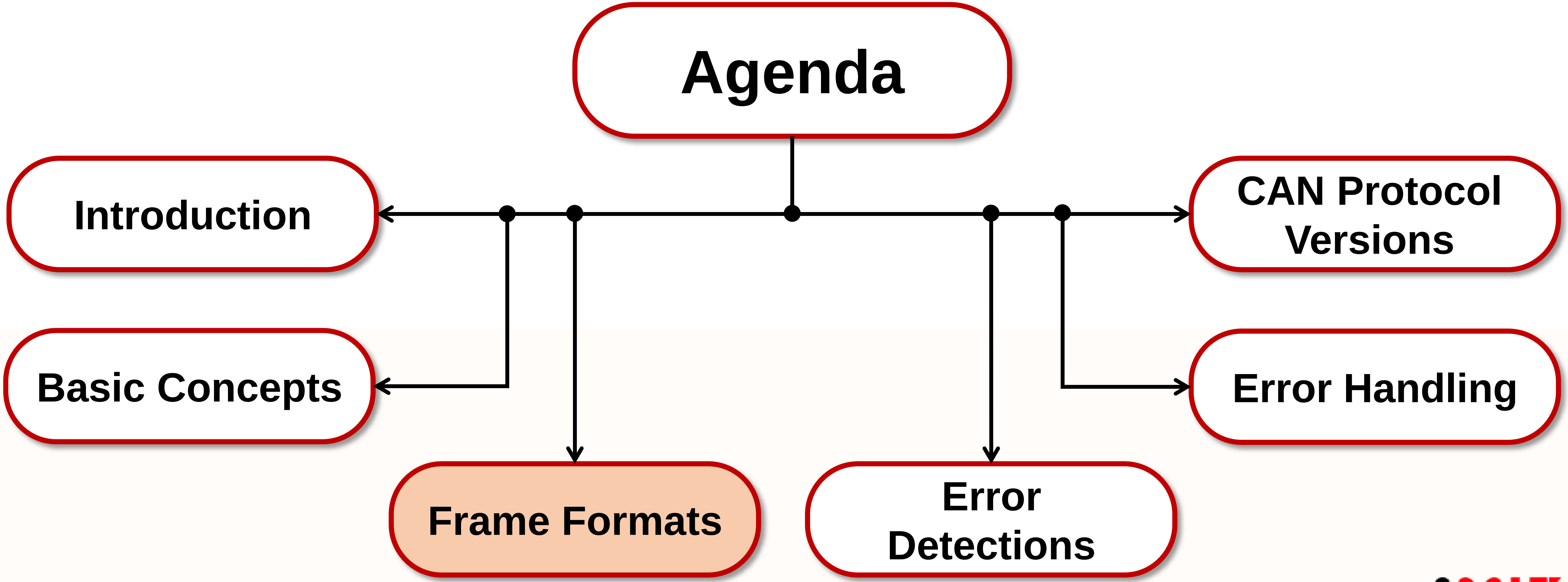
- Enables cost-effective communication with sensors and actuators when all the features of CAN are not required.

- The main features of this protocol (compared to CAN) are low cost and low speed and used for short distance networks.

➢ Basic Concepts

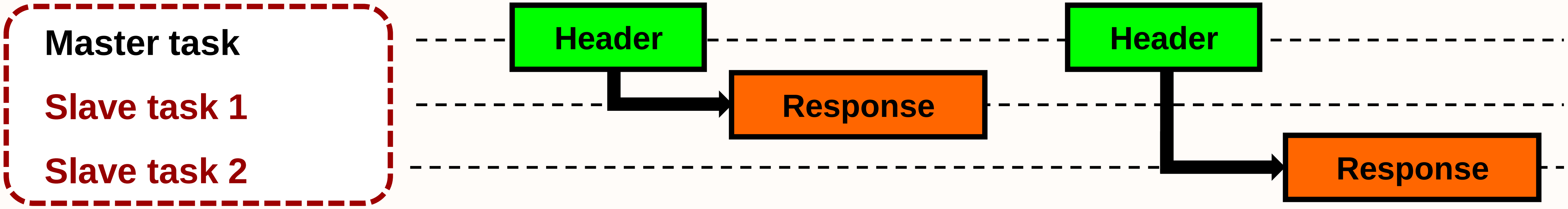
- The master is typically a moderately powerful microcontroller, whereas the slaves can be less powerful, cheaper microcontrollers or dedicated ASICs.
- The LIN is a single wire 12V bus connection, in which the communication protocol is based upon ISO9141 NRZ- standard.
- An important feature of LIN is the synchronization mechanism that allows the clock recover by slave nodes without quartz or ceramics resonator.
- Only the master node will be using the oscillating device. Nodes can be added to the LIN network Without requiring hardware or software changes in other slave nodes.

Agenda


Agenda
Introduction
CAN Protocol Versions
Basic Concepts
Error Handling
Frame Formats
Error Detections

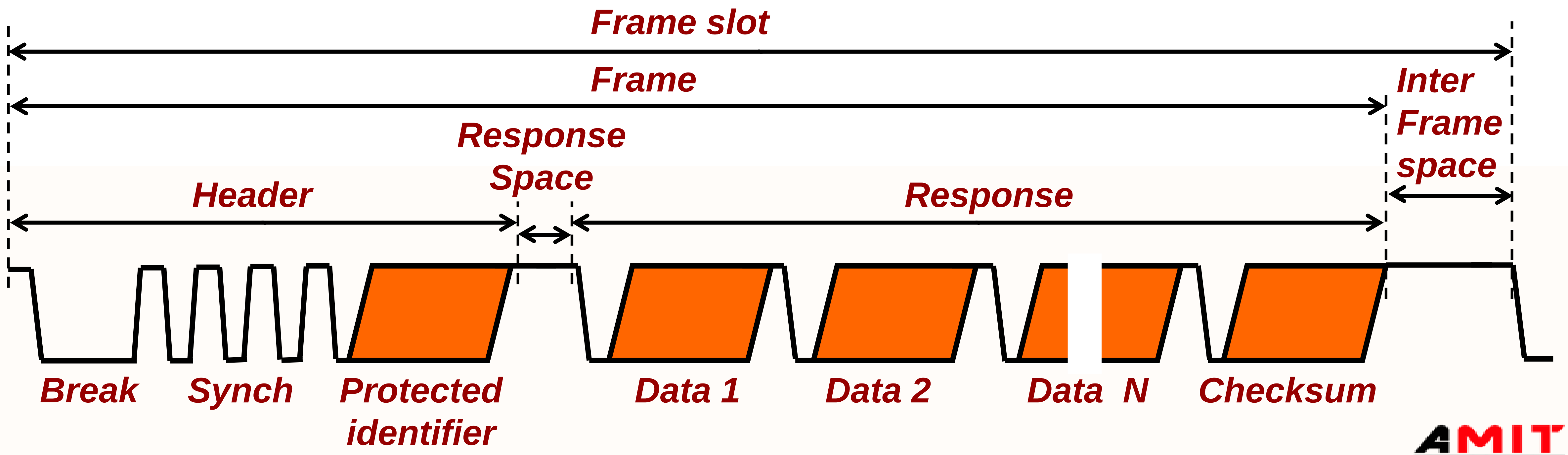
➢ Frame Formats

- A frame consists of a header (provided by the master task) and a response (provided by a slave task)
- The slave tasks interested in the data associated with the identifier receives the response, verifies the checksum and uses the data transported



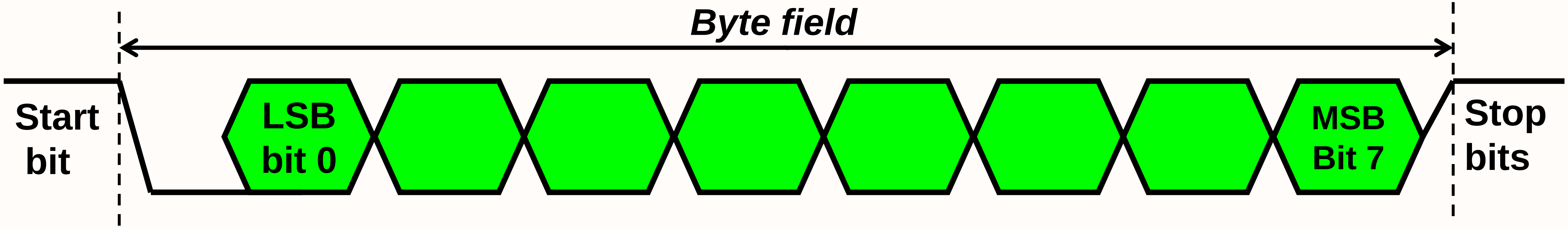
➢ Frame Formats

➢ Frame Structure



➢ Frame Formats

- Frame Structure
- Structure od a Byte field
  - **The LSB of data is sent first and the MSB last.**
  - **The start bit is encoded as a bit with value zero (dominant) and the stop bit is encoded as a bit with value one (recessive)**

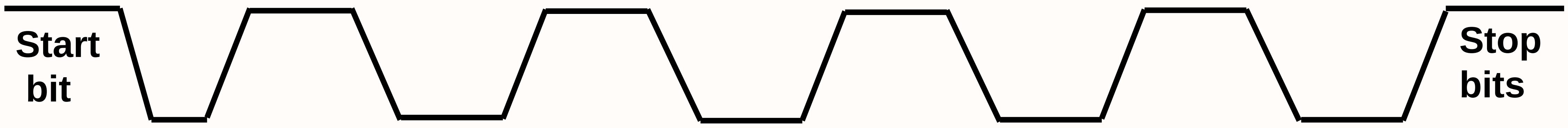


Frame Formats

- Break:
  - **The break symbol is used to signal the beginning of a new frame**
  - **A break is always generated by the master task and it shall be at least 3 bits of dominant value, including the start bit, followed by a break delimiter**

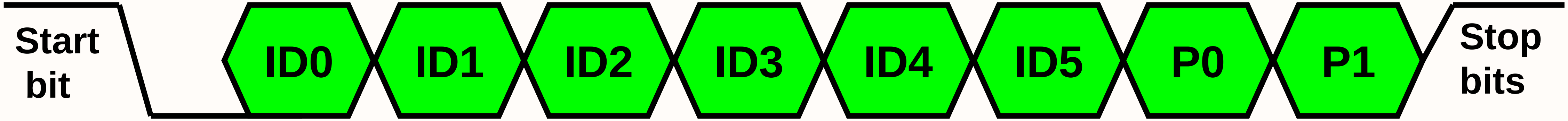
Frame Formats

- Synch Byte:
  - **Synch is a byte field with data value 0x55**
  - **A slave task shell always be able to detect the break/synch symbol sequence**



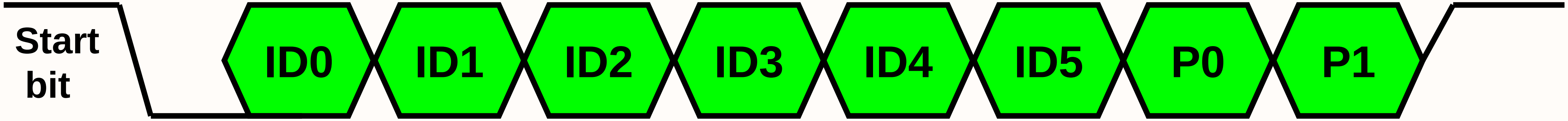
Frame Formats

- Protected Identifier
- Identifier
  - **Six bits are reserved for the identifier (ID), values in the range 0 to 63 can be used**
  - **The identifiers are split in four categories:**
  - **Values 0 to 59 (0x3b) are used for signal-carrying frames**
  - **60 (0x3c) and (0x3d) are used to carry diagnostic data**
  - **62 (0x3e) is reserved for user-defined extensions**
  - **63 (0x30) is reserved for future protocol enhancements**



Frame Formats

- **Parity:**
  - **The parity is calculated on the identifier bits**
  - **PO = ID0 $\oplus$ ID1 $\oplus$ ID2 $\oplus$ ID4**
  - **P1 = (ID1 $\oplus$ ID3 $\oplus$ ID4 $\oplus$ ID5)**



Frame Formats

- Data:
  - **A frame carries between one and eight bytes of data**
  - **A data byte is transmitted in a byte field**
  - **For data entities longer than one byte, the entity LSB is contained in the byte sent first and the entity MSB in the byte sent last (little-endian)**

Frame Formats

- Checksum:
  - The checksum contains the inverted eight bit sum with carry 6 over all data bytes or all data bytes and the protected identifier
  - Checksum calculation over the data bytes only is called classic checksum and it is used for communication with LIN 1.3 slaves
  - Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and it is used for communication with LIN 2.0 slaves
  - Identifiers 60 (0x3c) to 63 (0x3f) shall always use classic checksum

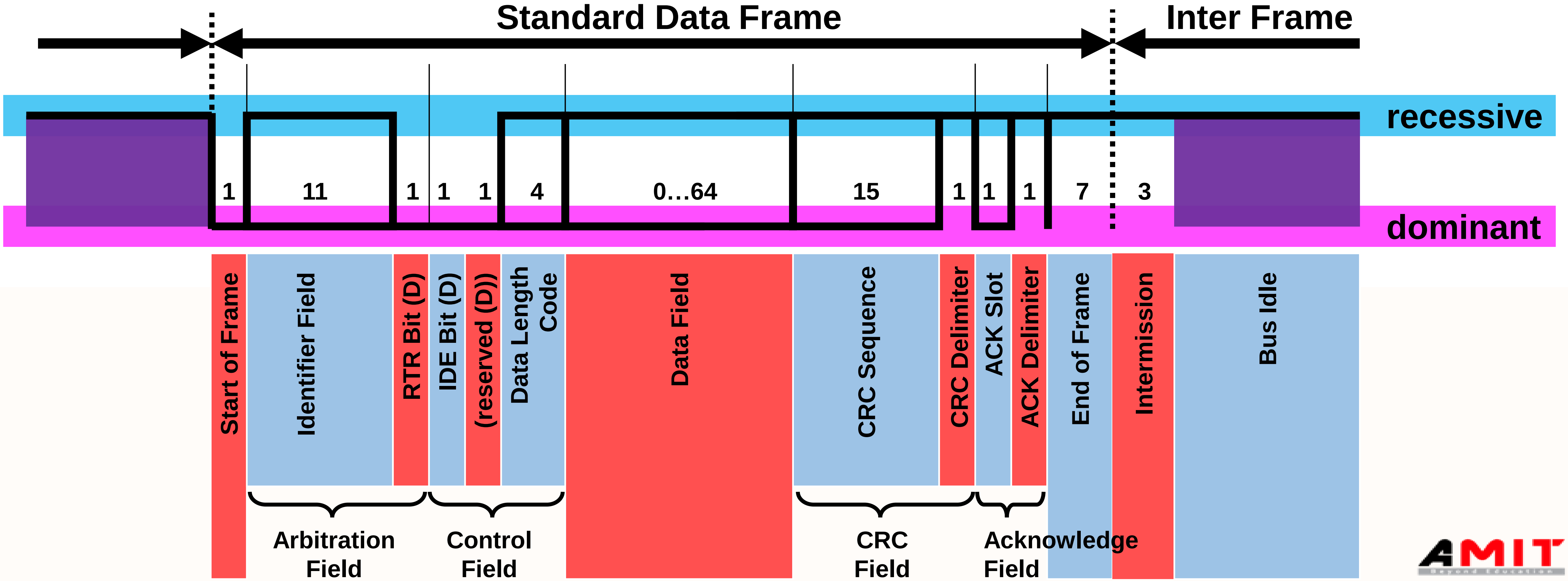
➤ Frame Formats

Frame Formats

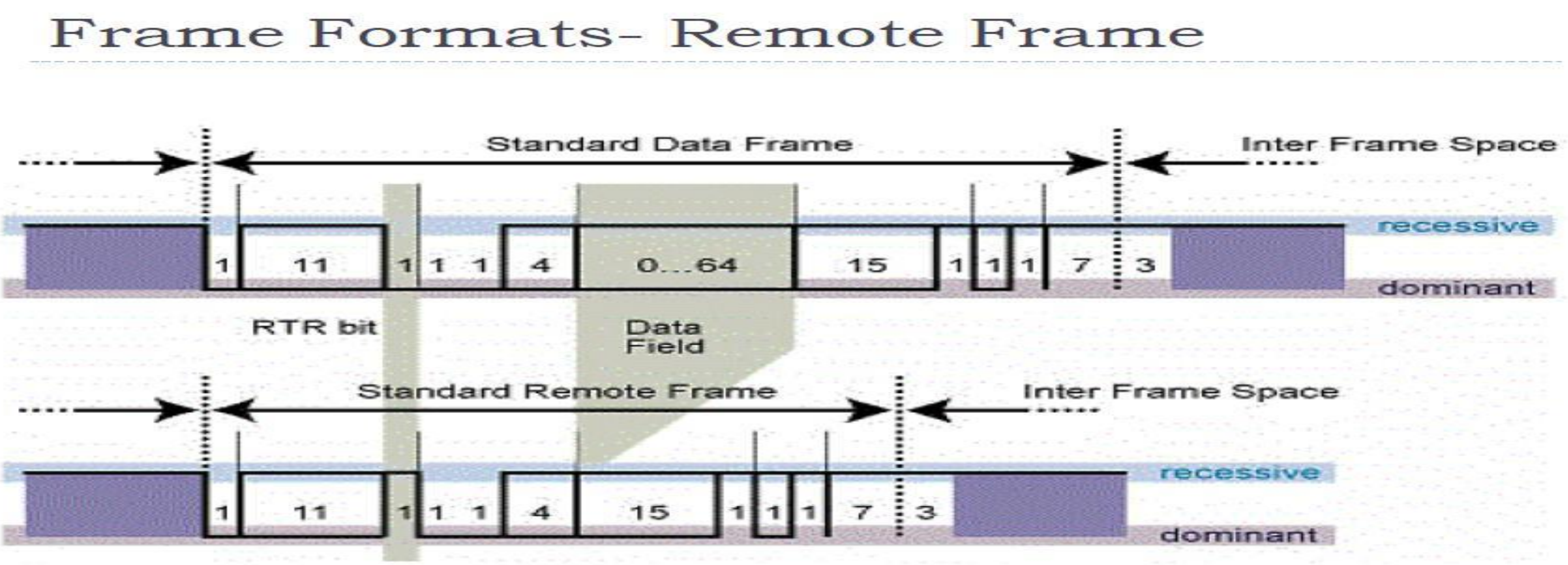
- Data Frame
- Remote Frame
- Error Frame
- Message vs. Signal
- Inter- Frame Space
- Overload Frame

➢ Frame Formats

➢ Data Frame



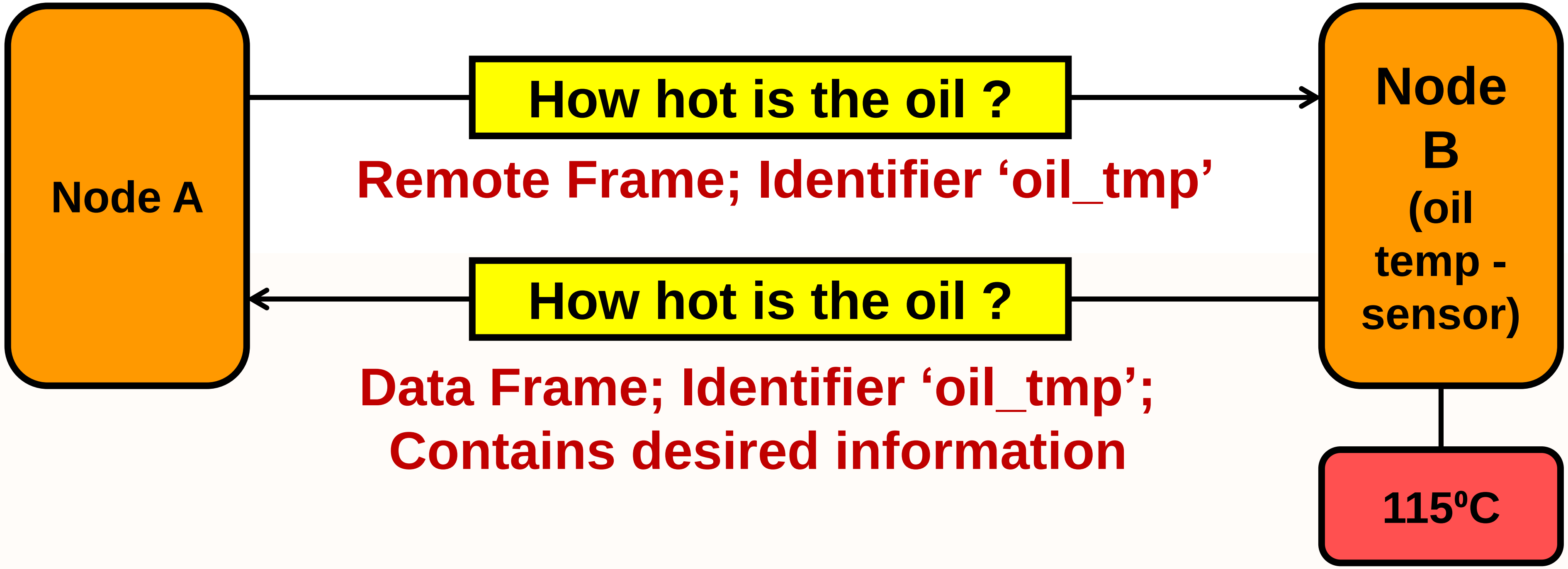
➢ Frame Formats



Frame Formats

Remote Frame

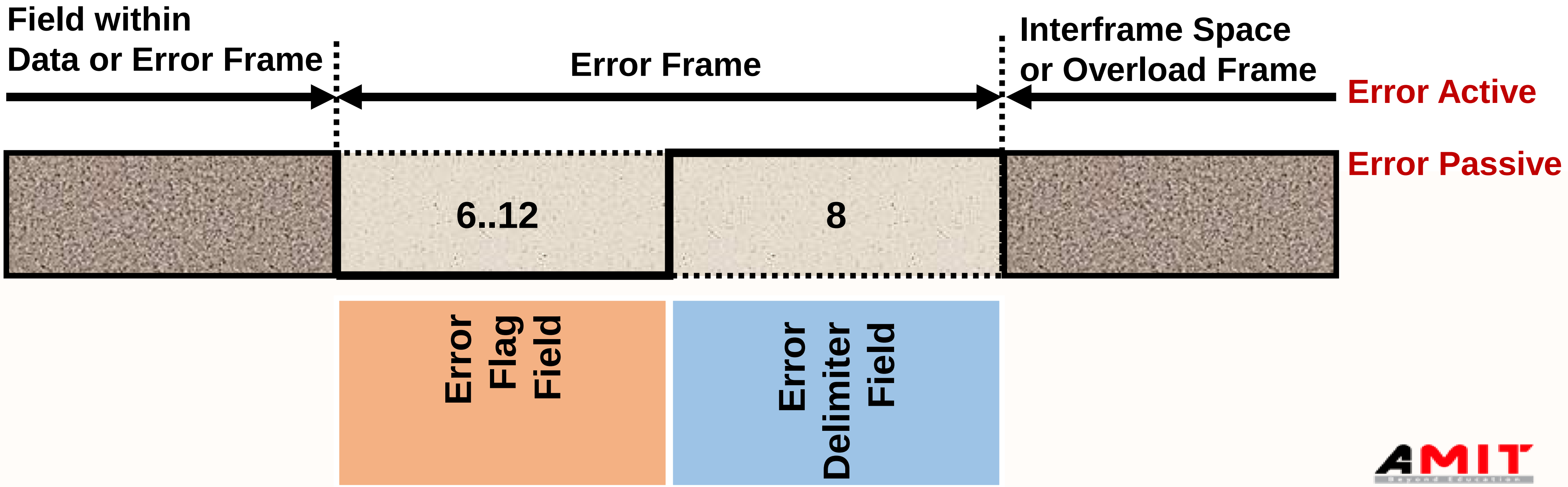
Remote Frame Scenario



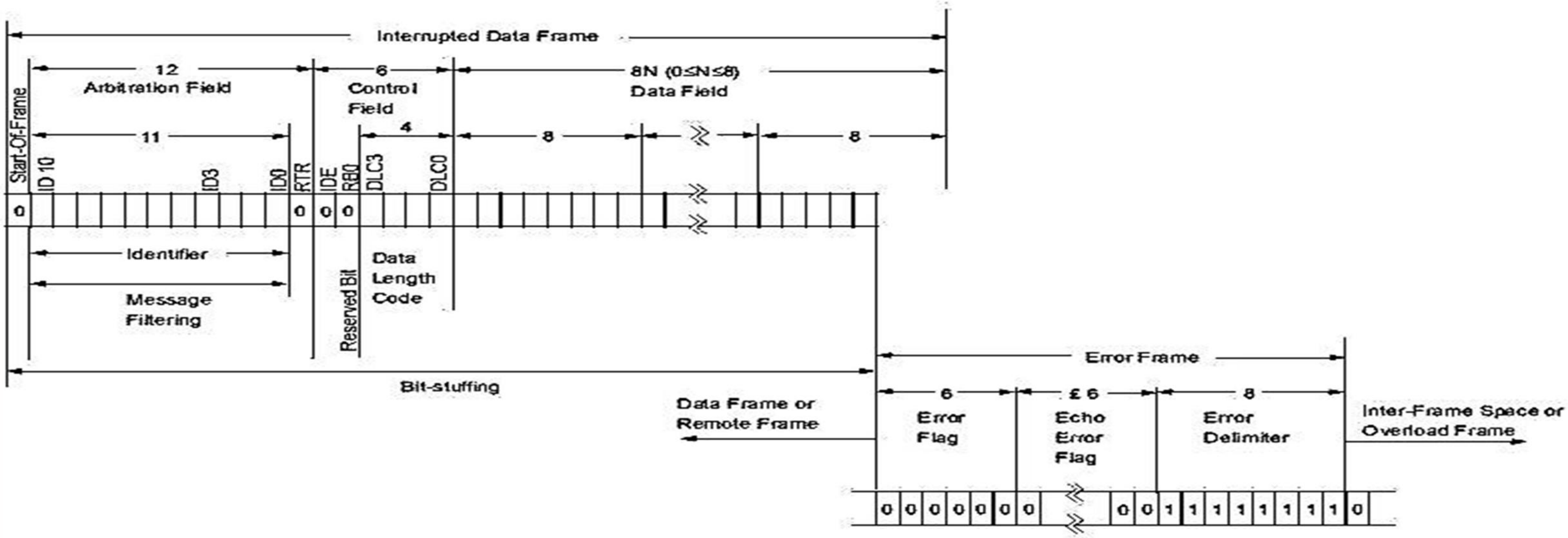
Frame Formats

Error Frame

Active Error Frame



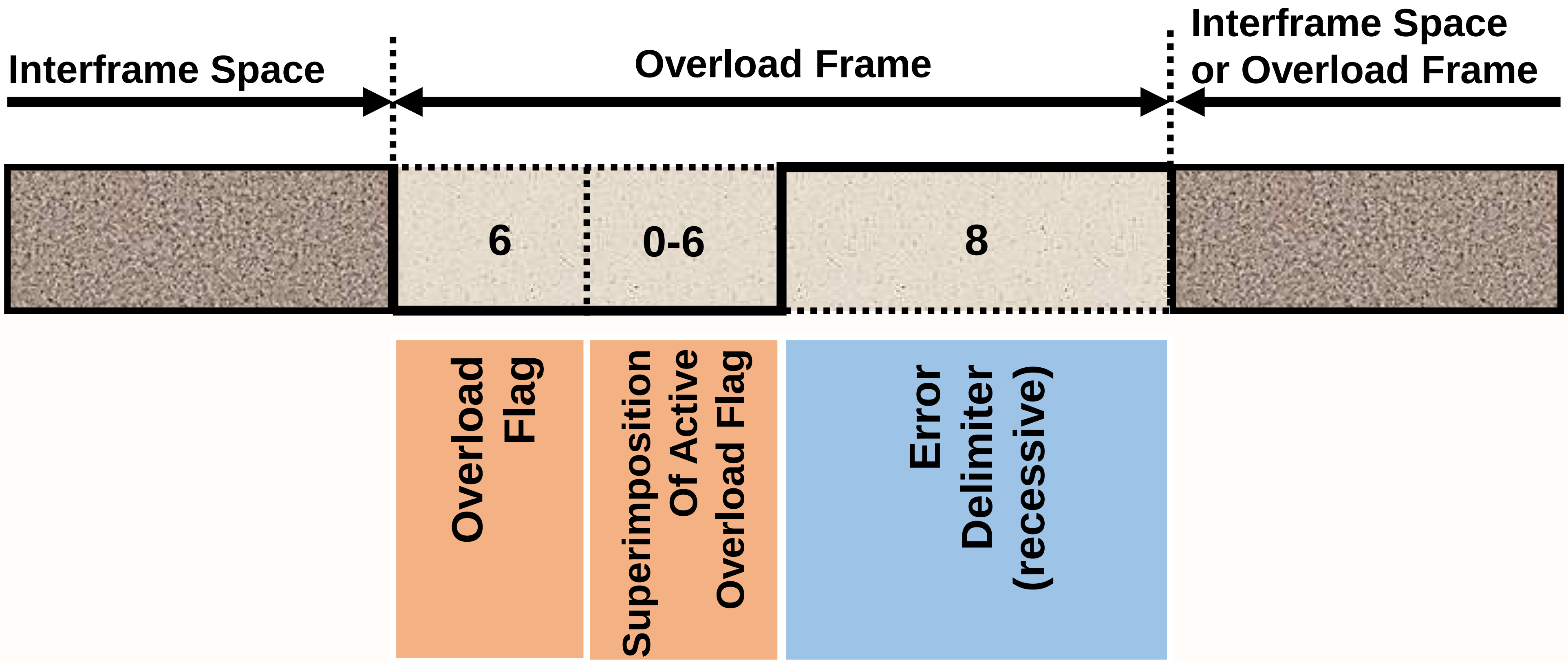
➢ Frame Formats



Frame Formats

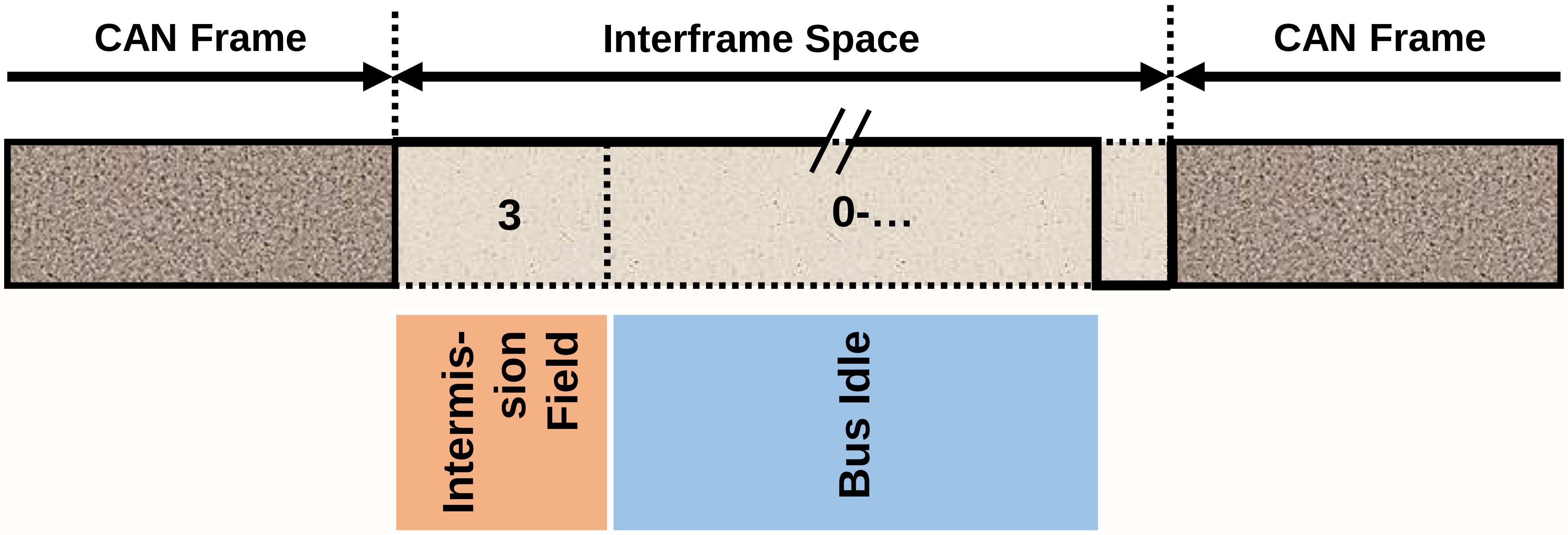
Overload Frame

- **Overload Frame used to delay next CAN message**



Frame Formats

- **Inter- Frame Space**
  - **Separates a frame (of whatever type) from a following Data or Remote frame**



➢ Frame Formats

Message vs. Signal

- ➢ Messages are transmitted between Network Nodes over the bus

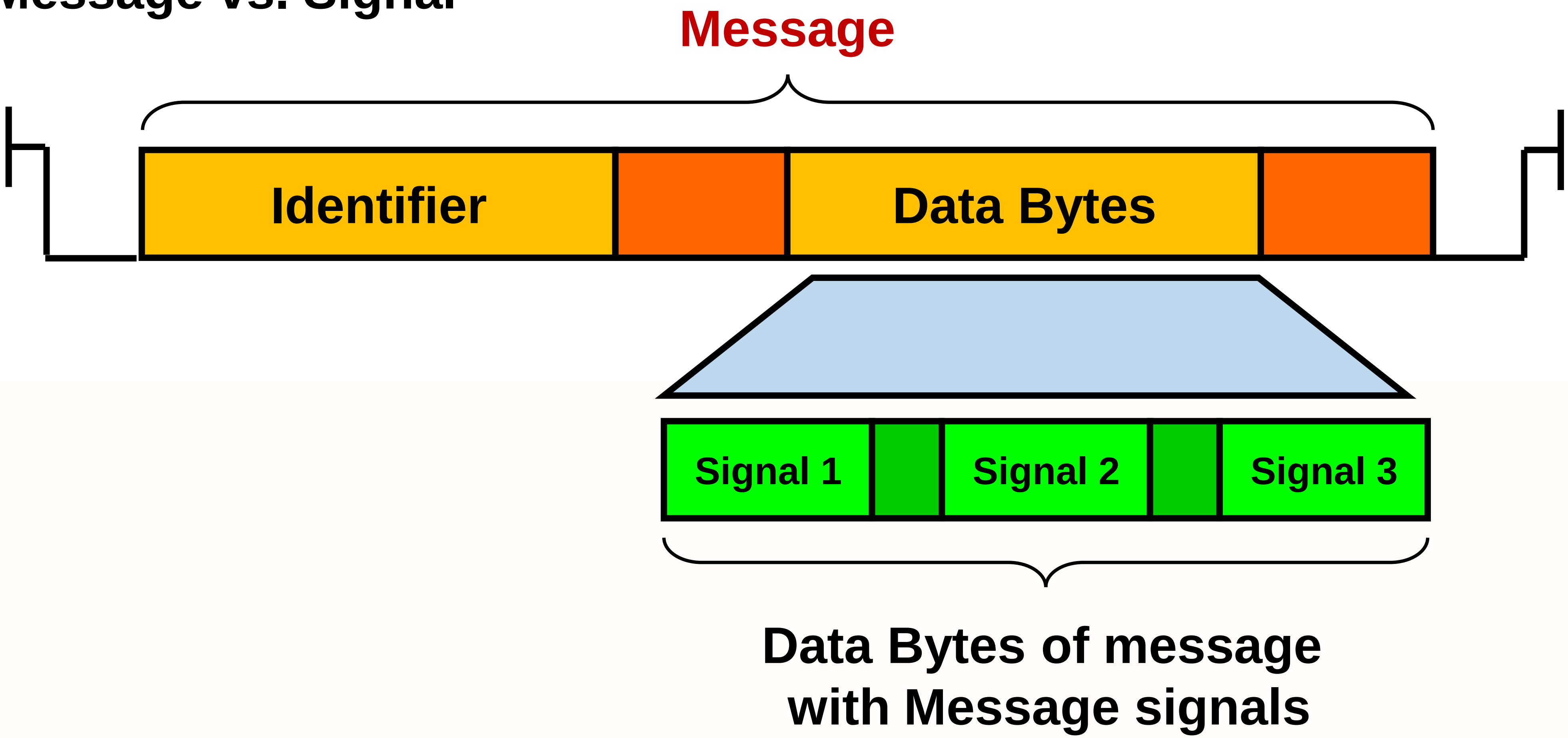
- ➢ The data bytes of any message is divided into Signals

- ➢ Signals represent a physical value

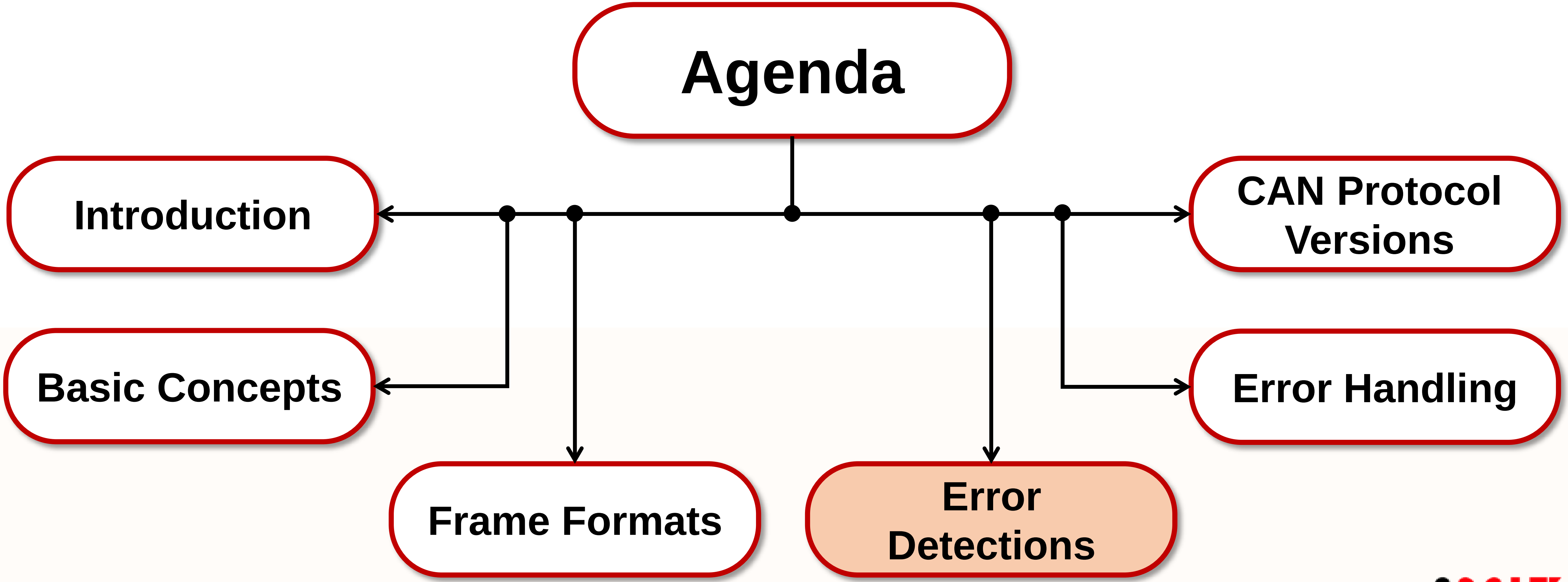
- ➢ For example, one data byte could be divided into 3 Signals as:
  - ➢ **4 bits that represent the vehicle speed => e.g: Named SpeedSig**
  - ➢ **2 bits that represent the gear speed => e.g: Named GearPosSig**
  - ➢ **2 bits that represent the light speed => e.g: Named LightStaSig**

➢ Frame Formats

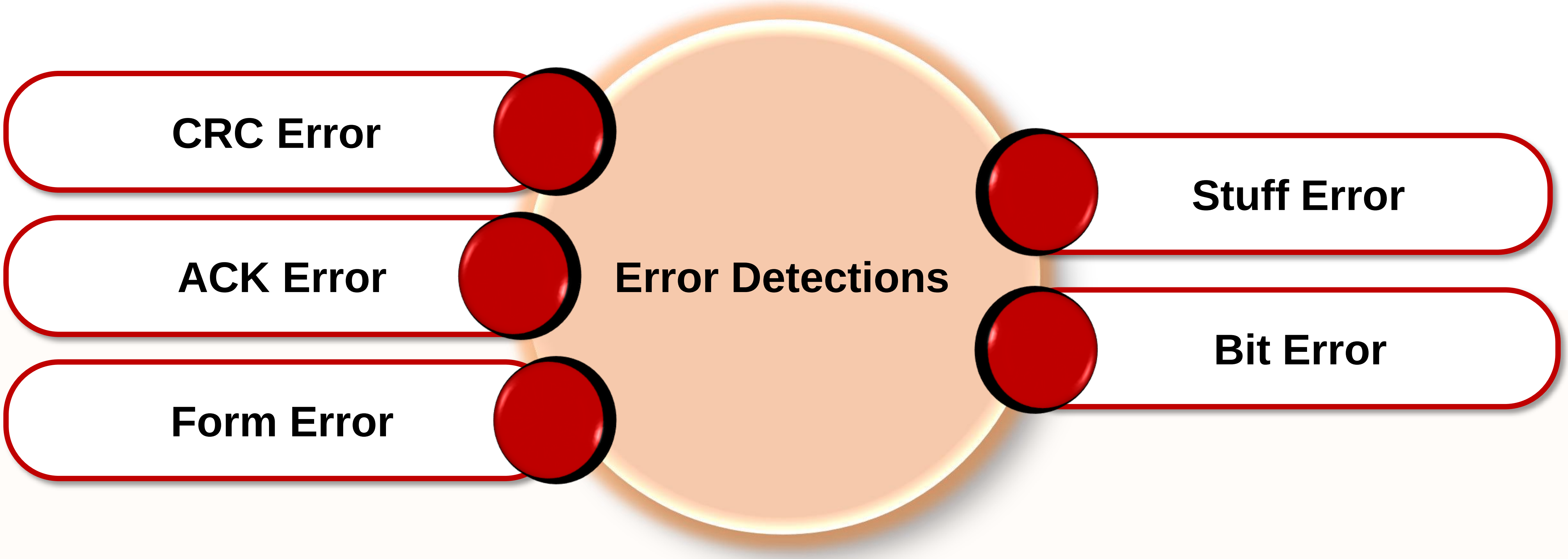
➢ **Message vs. Signal**



Agenda


Agenda
Introduction
CAN Protocol Versions
Basic Concepts
Error Handling
Frame Formats
Error Detections

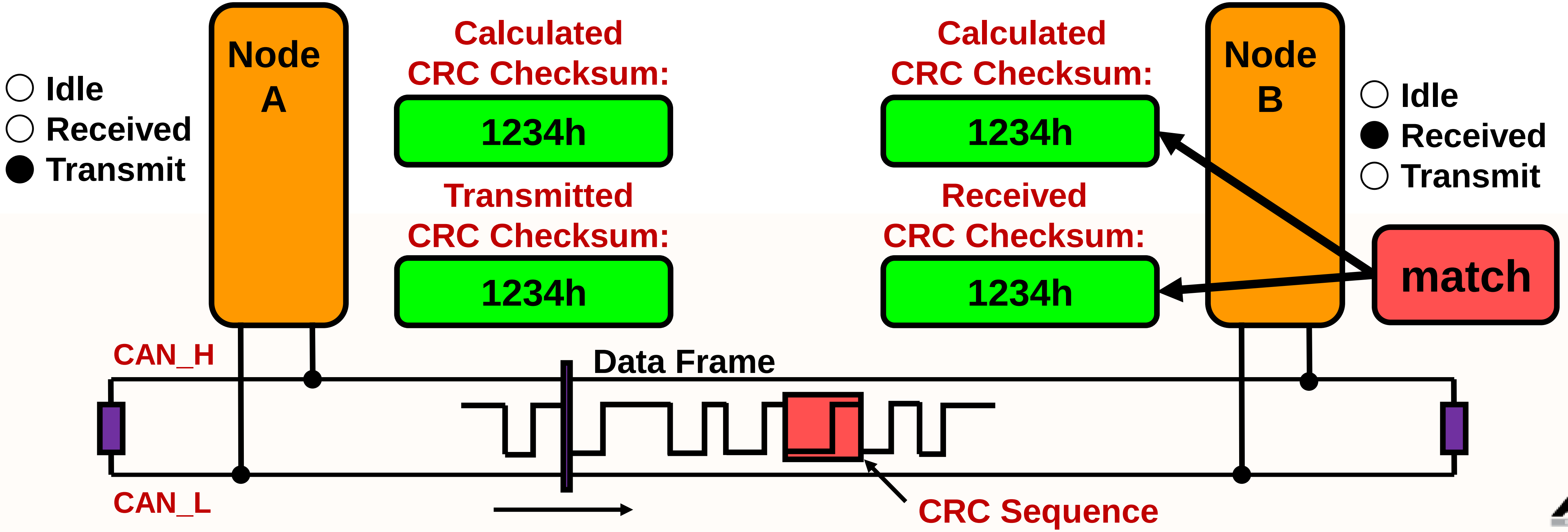
Error Detections



Error Detections

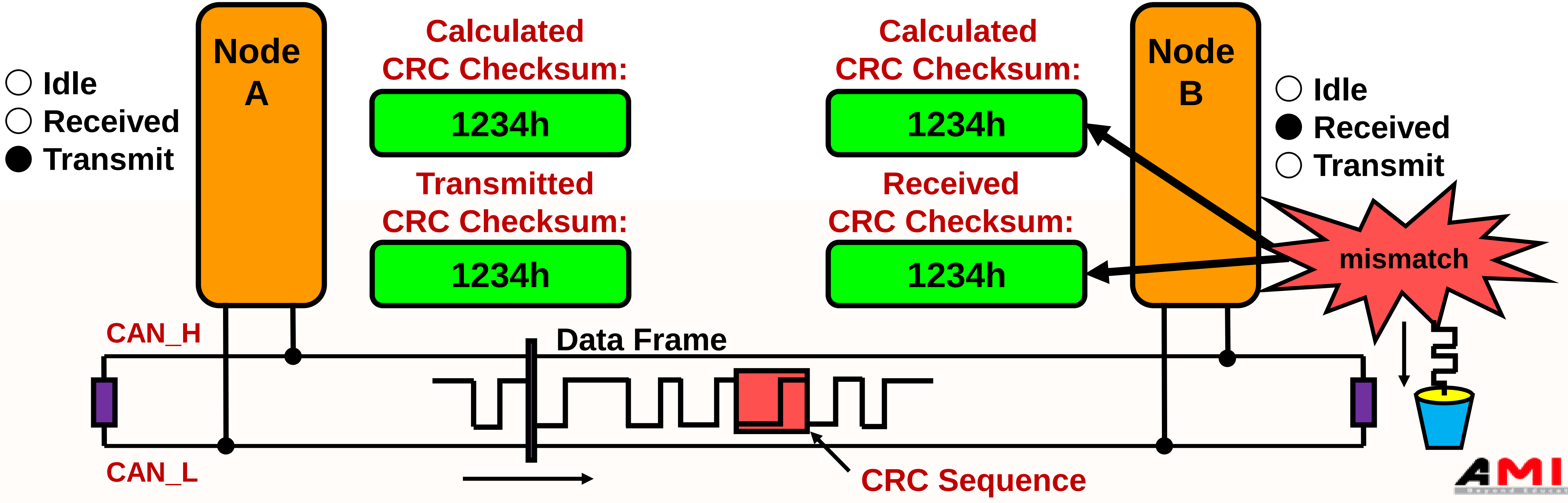
CRC Error

- **Calculated and Received checksum must match**



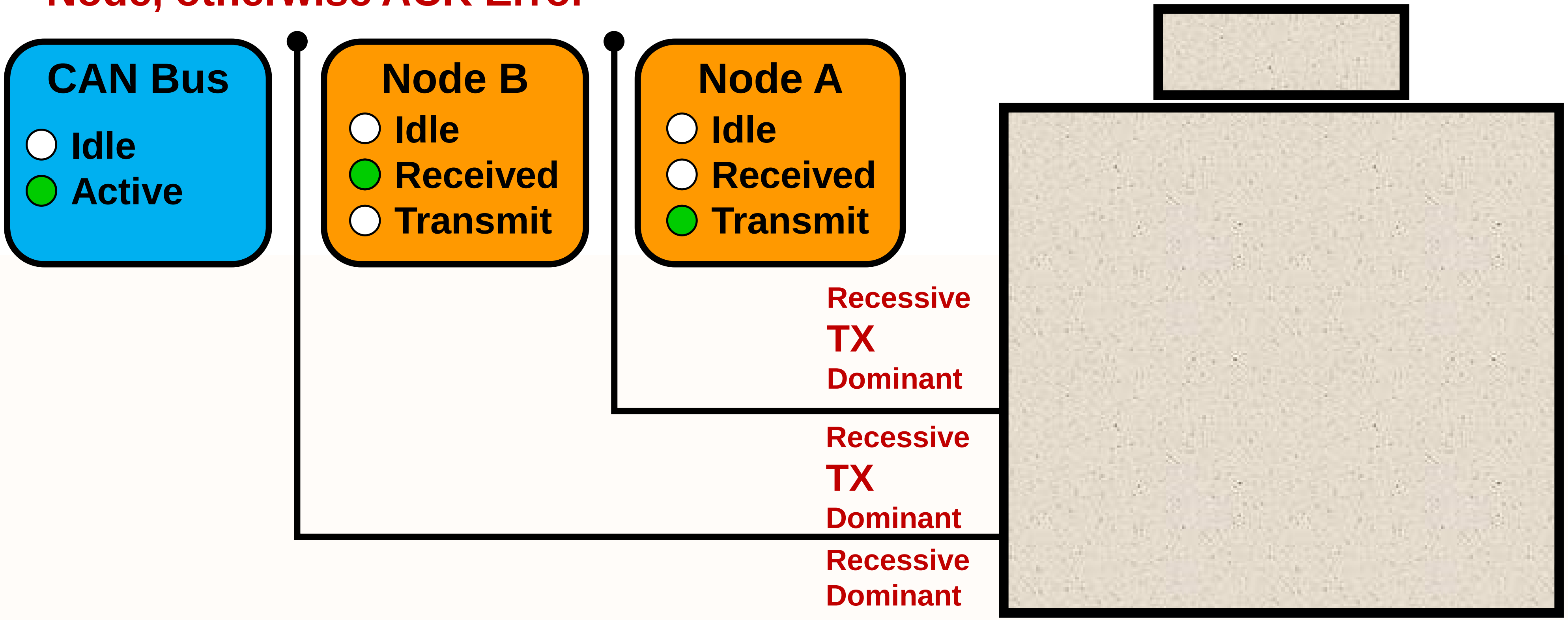
- **Agenda**
  - Error Detections

- **CRC Error**
  - Otherwise frame wasn't received correctly (CRC Error)



Error Detections

- **Acknowledge**
  - **A frame must be Acknowledged by at least one other Node, otherwise ACK Error**



➢ Error Detections

Frame Check

- No Dominant Bits allowed in
  - **CRC Delimiter**
  - **ACK Delimiter**
  - **End of Frame**
  - **Inter-frame space**

Otherwise Form Error is generated

Error Detections

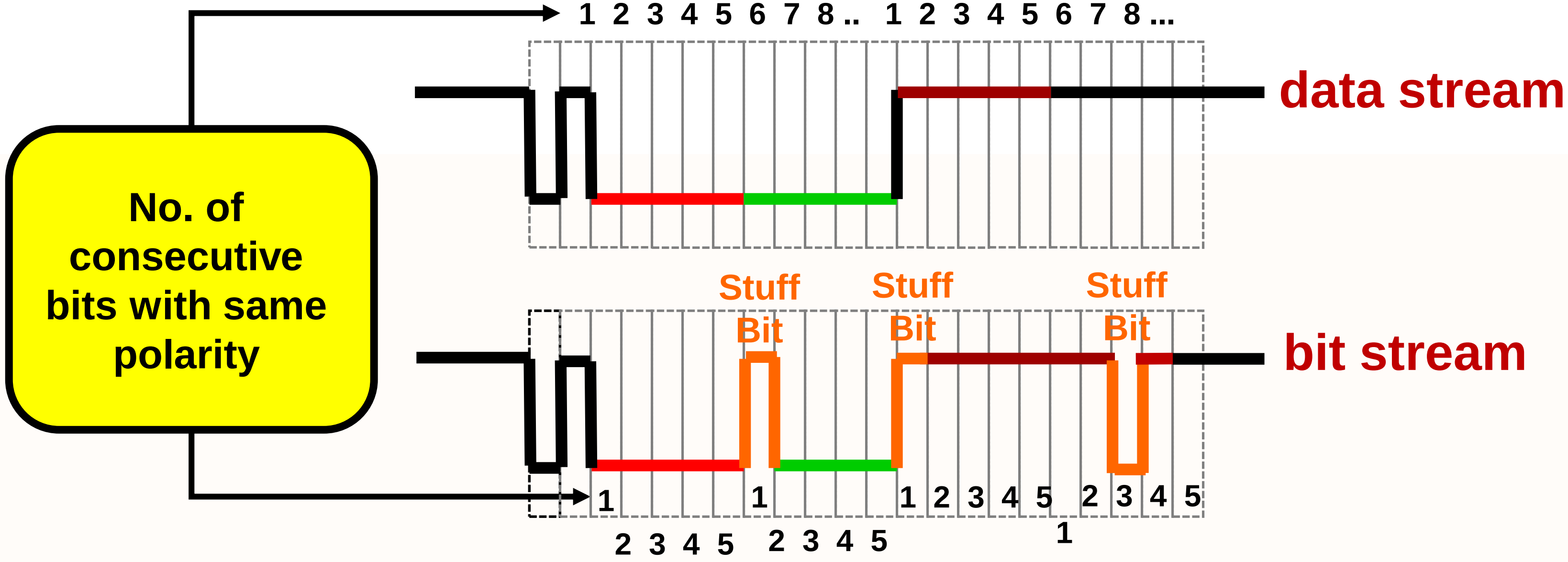
- **Bit Monitoring**
  - **A transmitted bit must be correctly read back from CAN Bus, otherwise Bit Error**
  - **Dominant bits may overwrite recessive bits only in the Arbitration field and in the Acknowledge slot.**

Error Detections

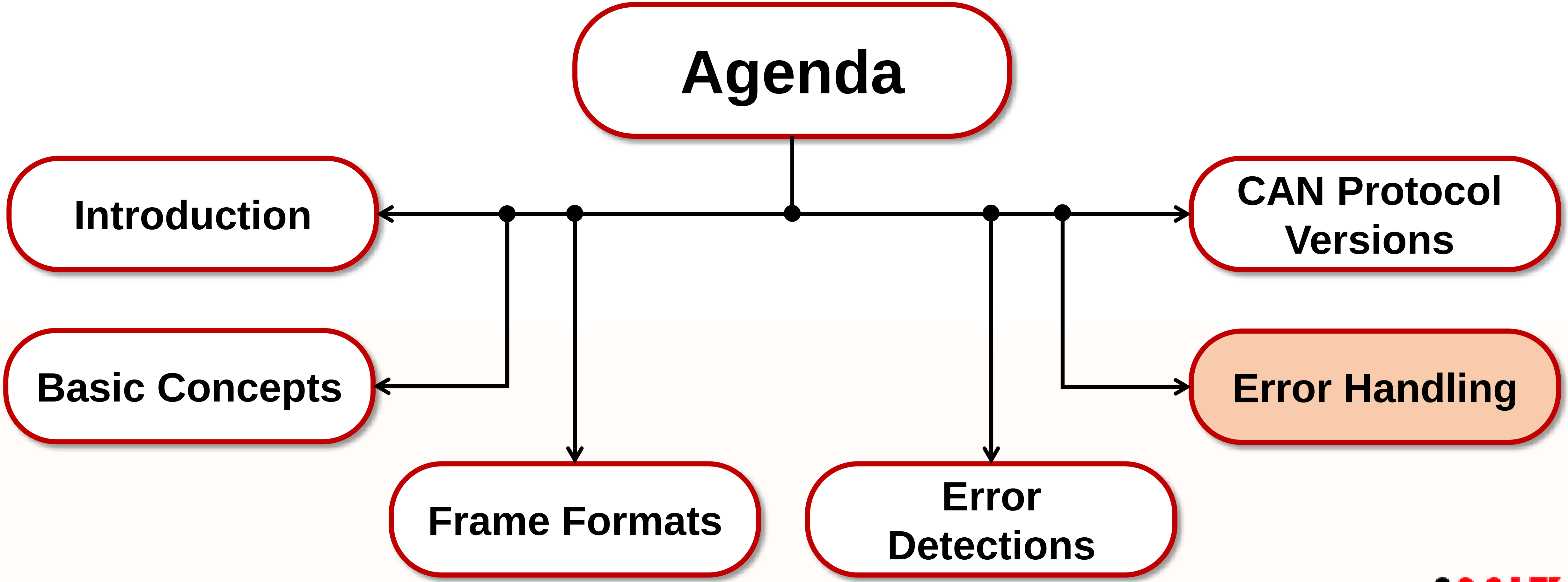
Bit Stuffing

- **Six consecutive bits with same polarity are not allowed between start of frame and CRC Delimiter**

Otherwise Bit Stuffing Error

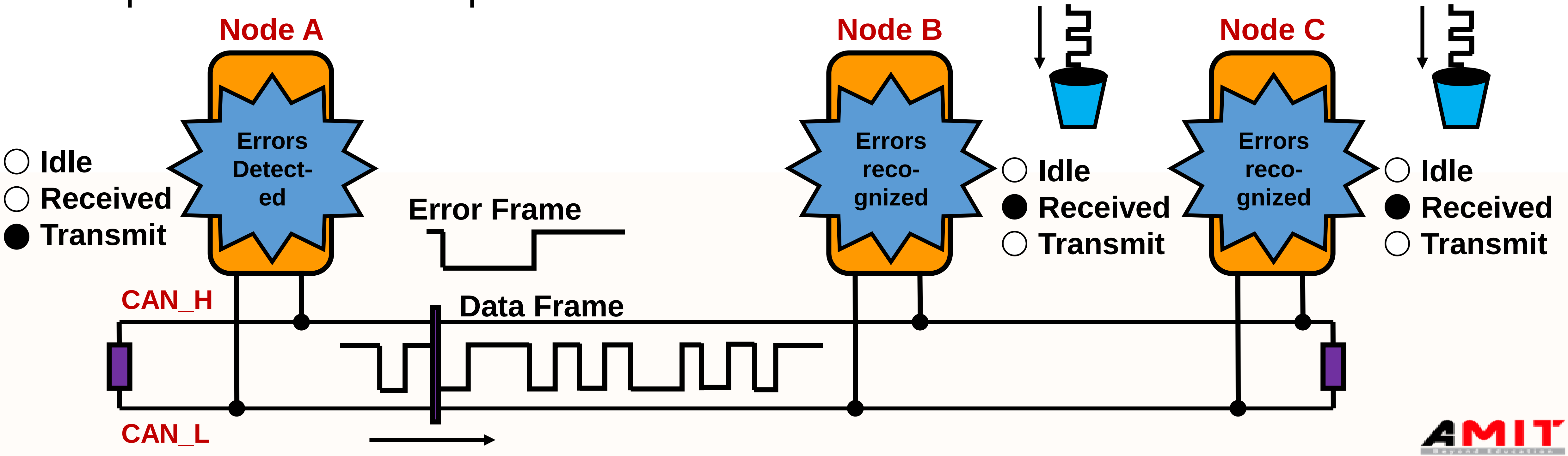


Agenda


Agenda
Introduction
Basic Concepts
Frame Formats
Error Detections
Error Handling
CAN Protocol Versions

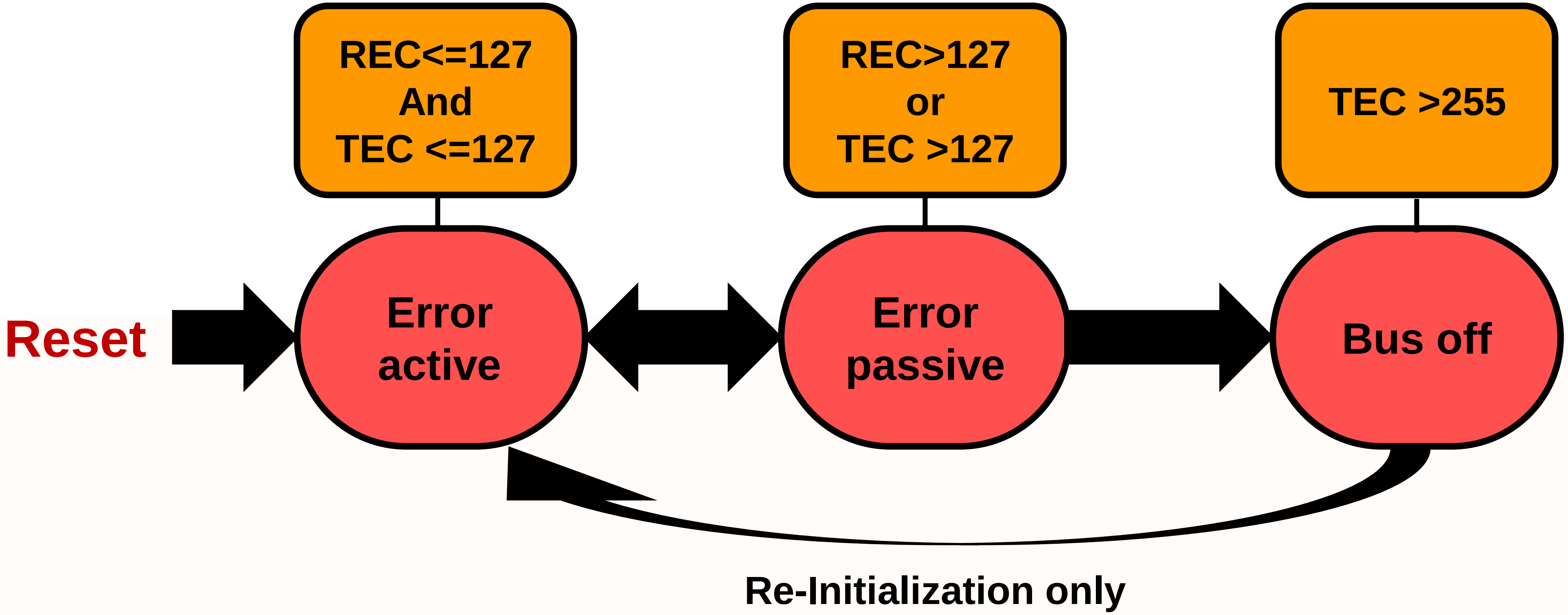
➢ Error Handling

- ➢ Detected Errors are made public to all other nodes via Error Frames.
- ➢ The transmission of the erroneous message is aborted and the frame is repeated as soon as possible.



Error Handling

- **Each Node is either in Error Active, Error Passive or off state.**



faulty nodes withdraw from the bus automatically (Bus off State)

Error Handling

Undetected Errors

- **Imagine a vehicle equipped with CAN**
  - **Running 2000hr/year**
  - **At CAN bus speed of 500 Kbps**
  - **With 25% bus load**

=> will result in I undetected errors every 2000 years!!

Agenda

- **Agenda**
  - Introduction
  - Basic Concepts
  - Frame Formats
  - Error Detections
  - Error Handling
  - CAN Protocol Versions

CAN Protocol Versions

- Two CAN protocol versions available:
  - V2.0A (Standard) – I I bit Massage ID's – 2048 ID's available

Start of Frame	Identifier 11 bits	Control Field	Data Field (0..8 Bytes)	CRC Field	ACK Field	End of Frame

  - V2.0B (Extended) – 29 bit Massage ID's – more than 536 Million ID's available

Start of Frame	Identifier 29 bits	Control Field	Data Field (0..8 Bytes)	CRC Field	ACK Field	End of Frame

CAN Protocol Versions

- **Three types of CAN modules available (all handles I I bit ID’s)**
  - **2.0A – Considers 29 bit ID as an error.**
  - **2.0B Passive – Ignores 29 bi ID message**
  - **2.0B Active – Handles both I I and 29 bit ID Message**

	Frame with 11 bit ID	Frame with 29 bit ID
V2.0B Active CAN Module	Tx/Rx OK	Tx/Rx OK
V2.0B Passive CAN Module	Tx/Rx OK	Tolerated
V2.0A CAN Module	Tx/Rx OK	Bus ERROP

THANK YOU!

AMIT
Beyond Education