9
Control Area Network (CAN)

18-540 Distributed Embedded Systems

Philip Koopman

October 4, 2000

Required Reading: **Schill, Overview of the CAN Protocol**

Most of the pictures in the lecture are from:

- **CAN specification (Bosch)**
- **Overview to Can; Infineon**
- **DeviceNet materials -- http://www.odva.org/**

Carnegie Mellon

Assignments

- **By next class read about Protocol building Blocks:**
  - Review protocol survey paper from this week
    (I haven't found papers that directly address this topic)

- **Dates to remember:**
  - Project Part #3: due in one week on Wednesday 10/11
  - HW #5 due at 4 PM on Friday 10/13

Where Are We Now?

- **Where we've been:**
  - Protocol Overview

- **Where we're going today:**
  - CAN -- an important embedded protocol
  - Primarily automotive, but used in many places

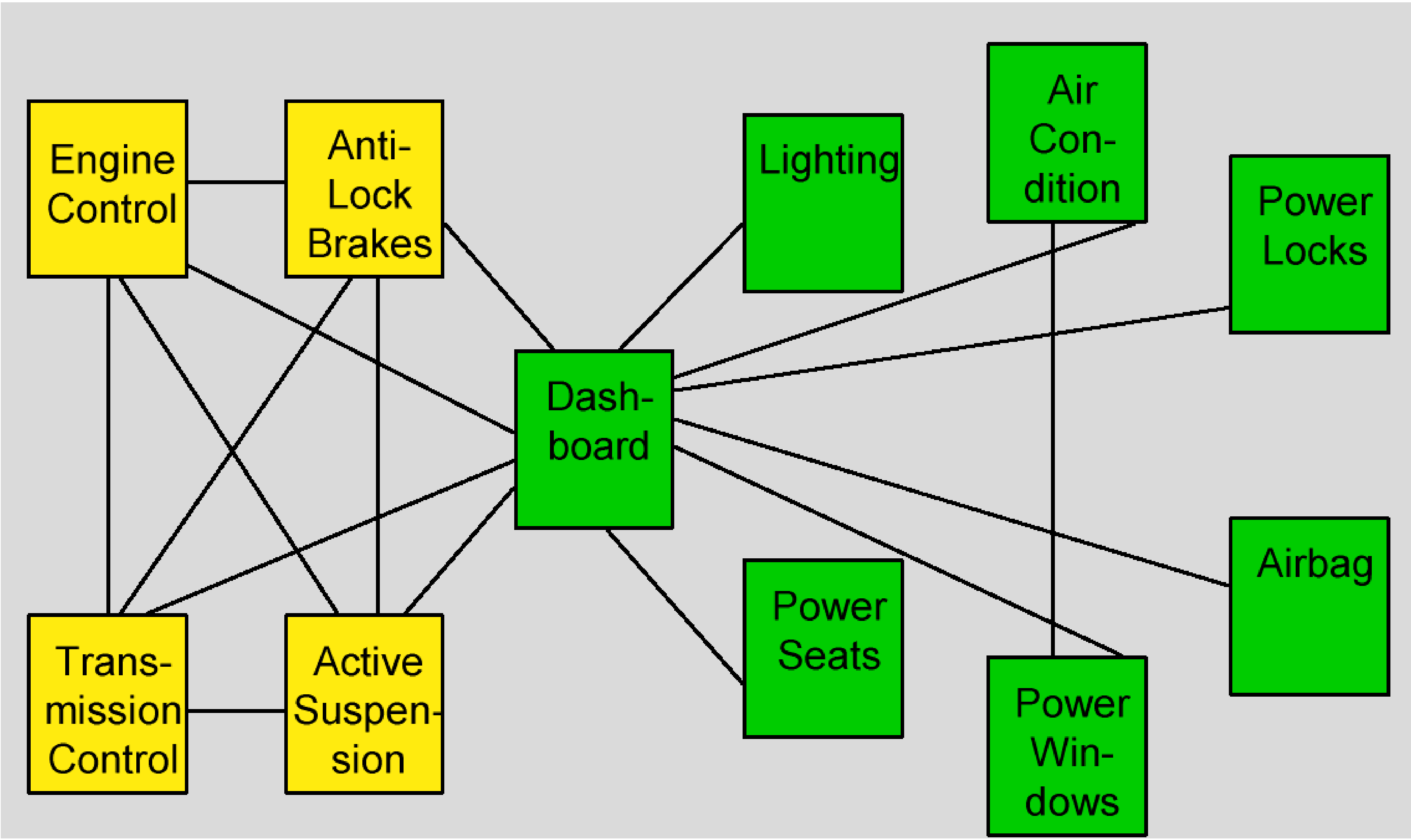
- **Where we're going next:**
  - A building-block approach to protocols:
    - Custom protocols
    - Protocol performance analysis

Preview

- **CAN – important automotive protocol**
  - Physical layer
  - Protocol layer
  - Message filtering layer (with add-on protocols)

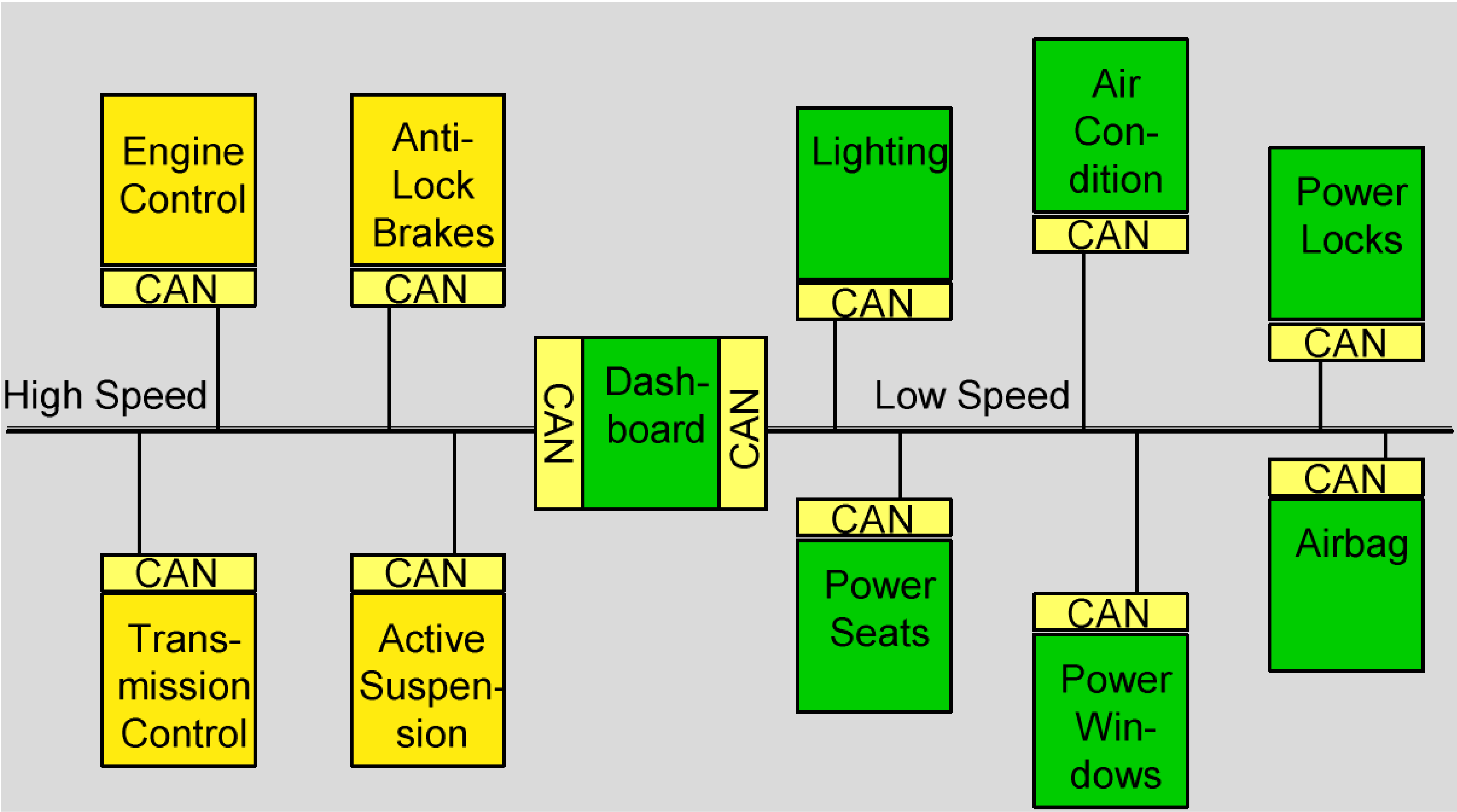
- **Keep an eye out for:**
  - Message prioritization
  - How “small” nodes can be kept from overloading with received messages
  - Tradeoffs

Before CAN



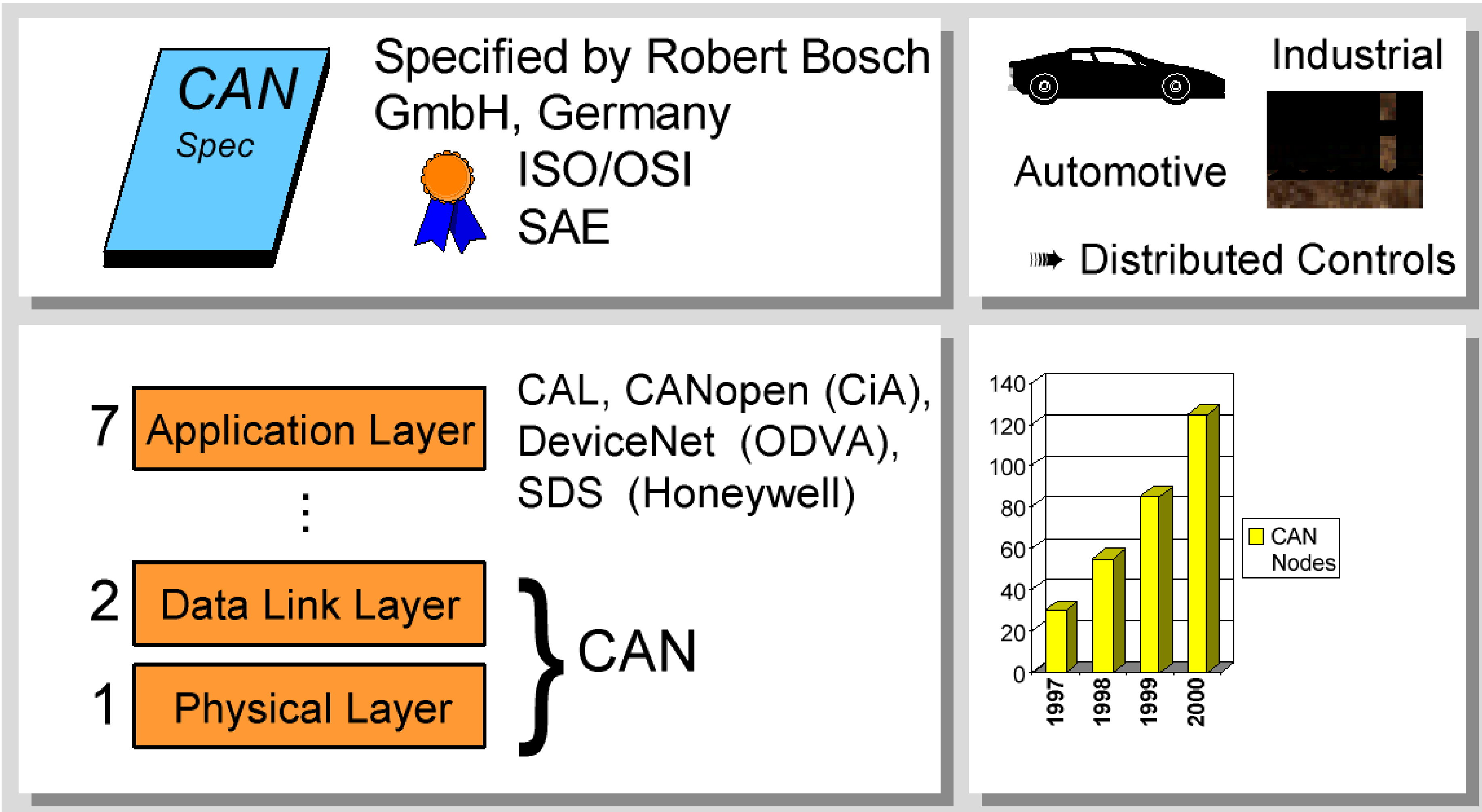
Engine Control
Anti-Lock Brakes
Lighting
Air Con-dition
Power Locks
Dash-board
Airbag
Trans-mission Control
Active Suspen-sion
Power Seats
Power Win-dows

With CAN



Engine Control
CAN
Anti-Lock Brakes
CAN
Lighting
CAN
Air Con-dition
CAN
Power Locks
CAN
High Speed
Low Speed
CAN
Dash-board
CAN
CAN
Trans-mission Control
CAN
Active Suspen-sion
CAN
Power Seats
CAN
Power Win-dows
CAN
Airbag

Generic CAN Propaganda Slide



CAN & the Protocol Layers

- **CAN only standardizes the lower layers**
- **Other high-level protocols are used for application layer**
  - User defined
  - Other standards

Application Layer

Object Layer

- Message Filtering
- Message and Status Handling

Transfer Layer

- Fault Confinement
- Error Detection and Signalling
- Message Validation
- Acknowledgment
- Arbitration
- Message Framing
- Transfer Rate and Timing

Physical Layer

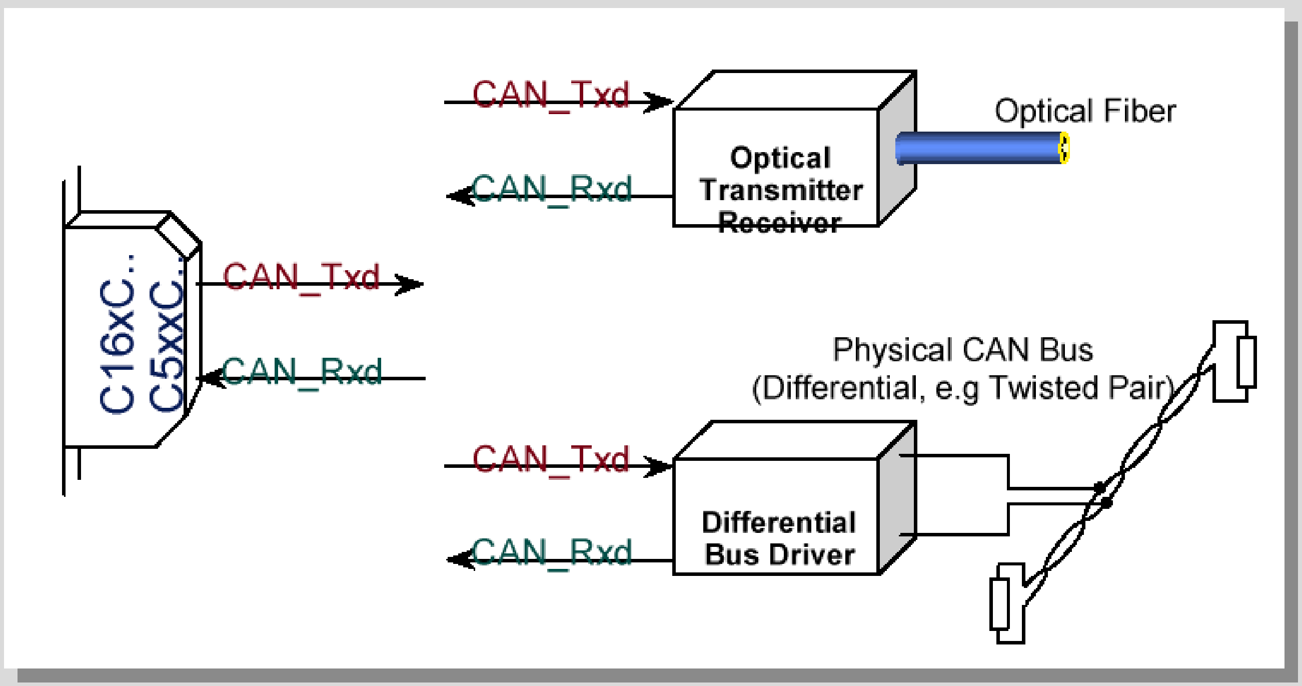
- Signal Level and Bit Representation
- Transmission Medium

Physical Layer Possibilities

- **MUST support bit dominance (discussed later)**
- **Specifically rules out transformer coupling for high-noise applications**
  - But, cars are high-noise, right????
  - Differential drive and optical fibers help in most cases, but not all

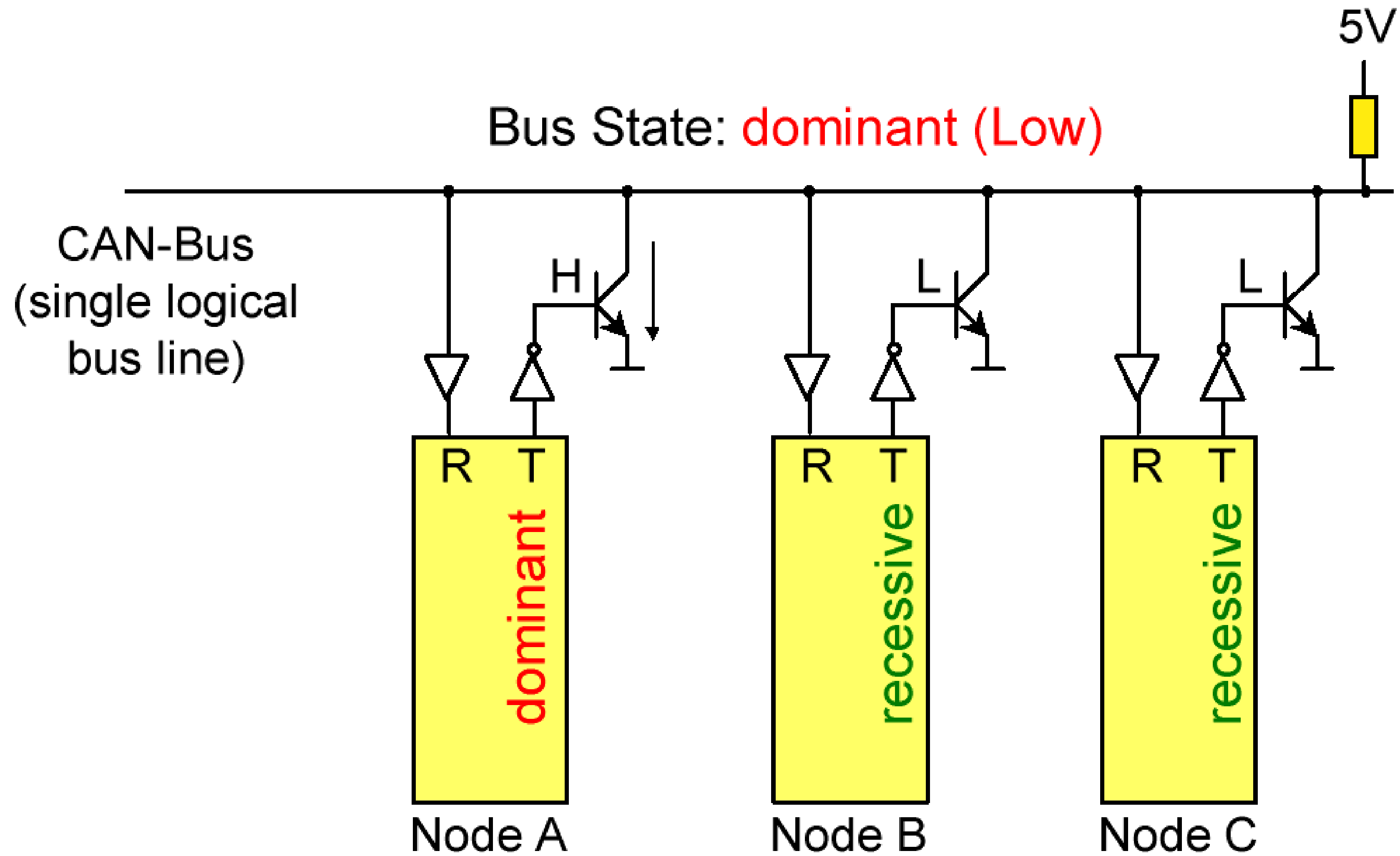
❑ Usual ISO Physical Layer :-

- **Bus wires twisted pair, 120R Termination at each end**
- **2 wires driven with differential signal (CAN_H, CAN_L)**

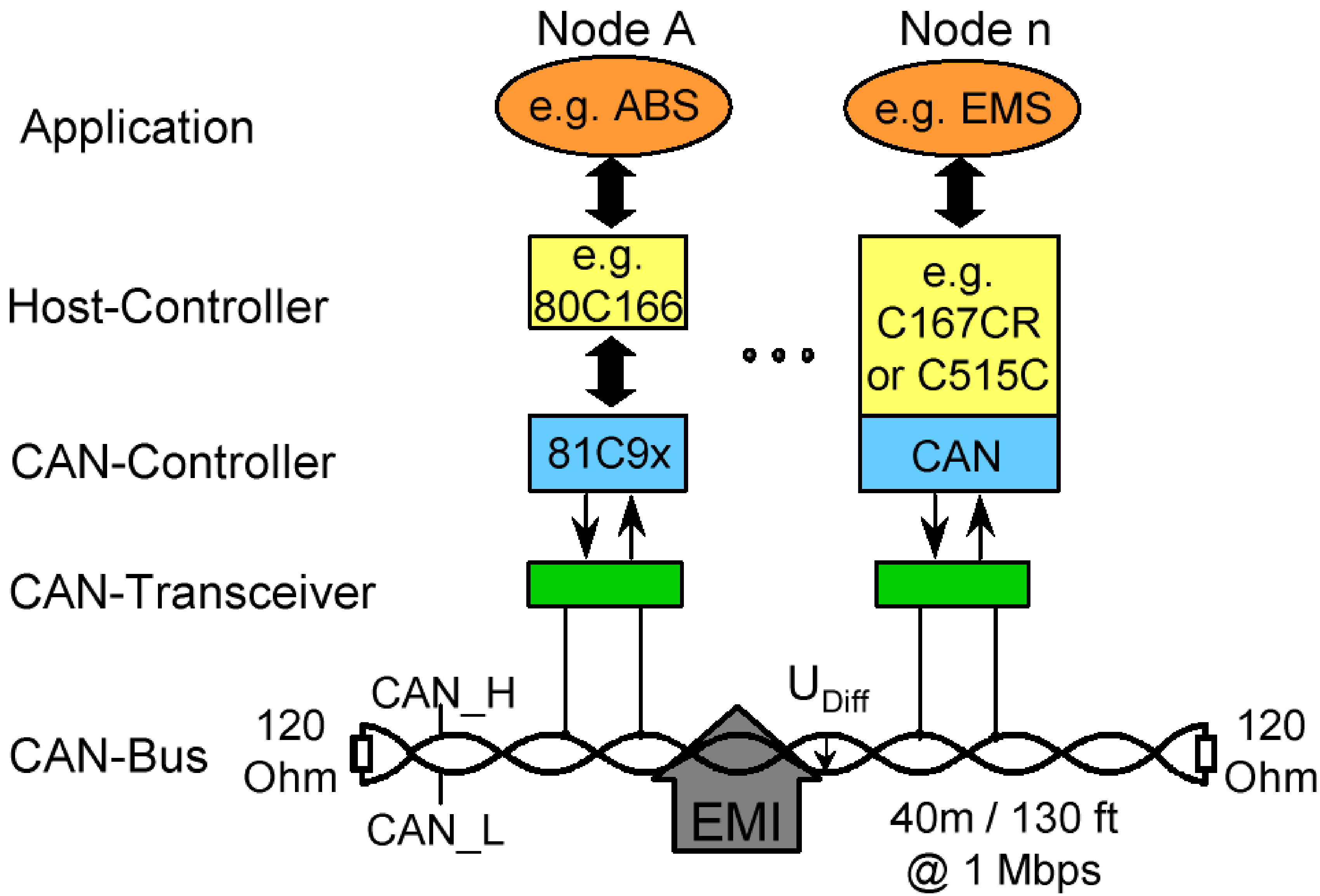


Bit Dominance

- **Wired "Or" design**
  - (Called "open collector logic" before TTL/tristate was invented…)



Generic CAN Network Implementation



Basic Bit Encoding - NRZ

- **NRZ = Non-Return-To_Zero**
  - Fewer transitions (on average) = less EMI, but requires less oscillator drift

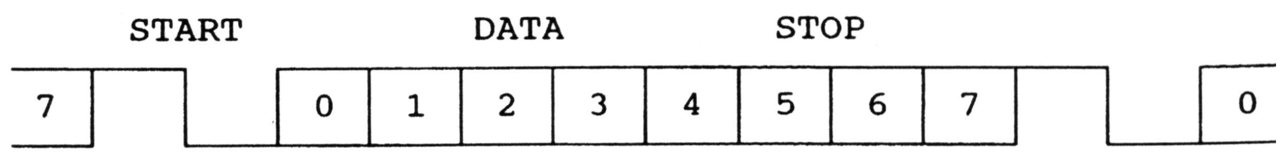


FIGURE 26.21 A 10-bit NRZ waveform (LSB first).

  - Bit stuffing relaxes oscillator drift requirements

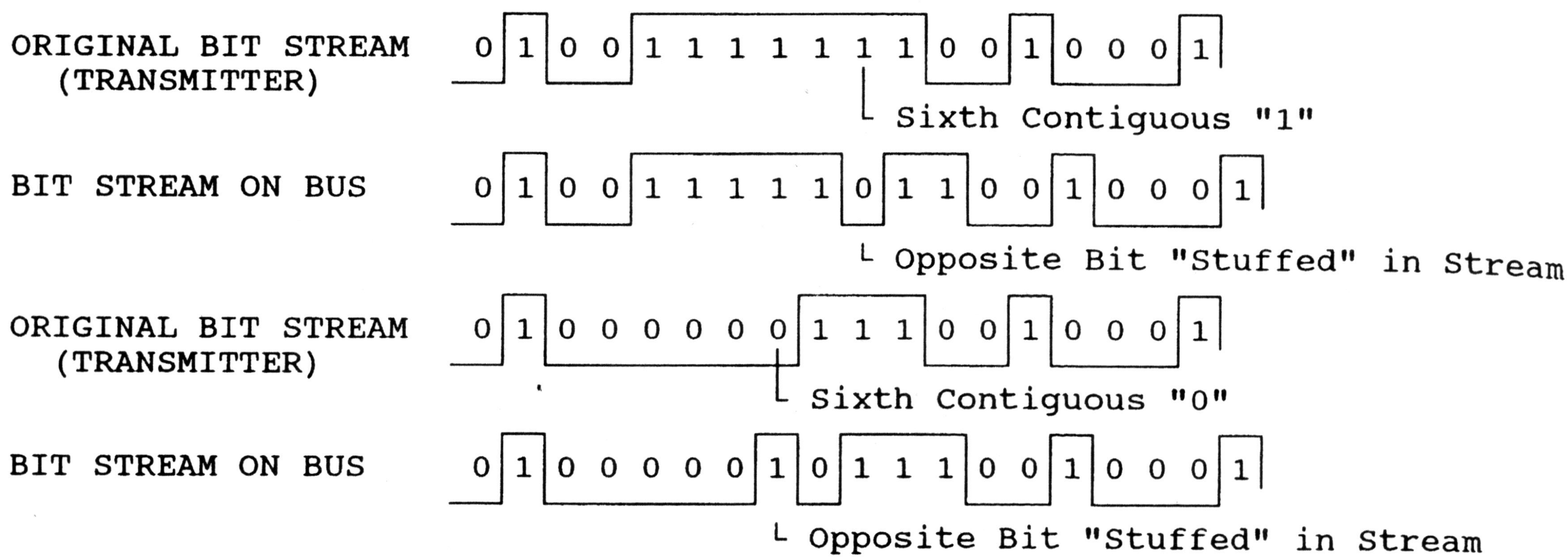
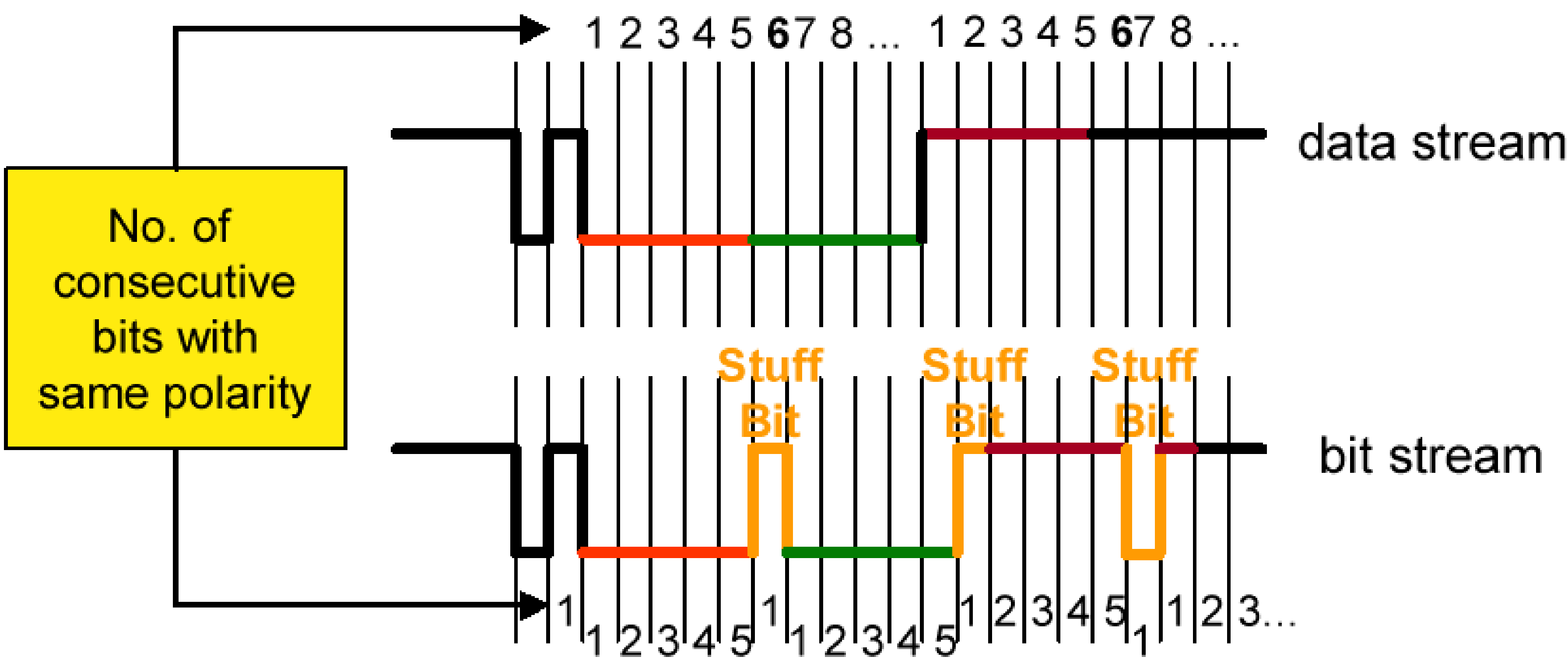


FIGURE 26.23 NRZ bit-stuffing.

Another Look at Bit Stuffing

- **Five identical bits in a row triggers an inverted Stuff Bit**
  - Bit de-stuffer must take it back out on the receiving end…
  - ***[This picture is slightly wrong -- it is 5 bits in source stream, not counting stuff bits…]***



Generic Message Format

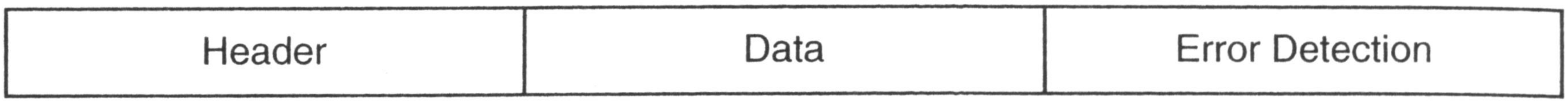


FIGURE 26.1 Three parts of a vehicle network frame or message.

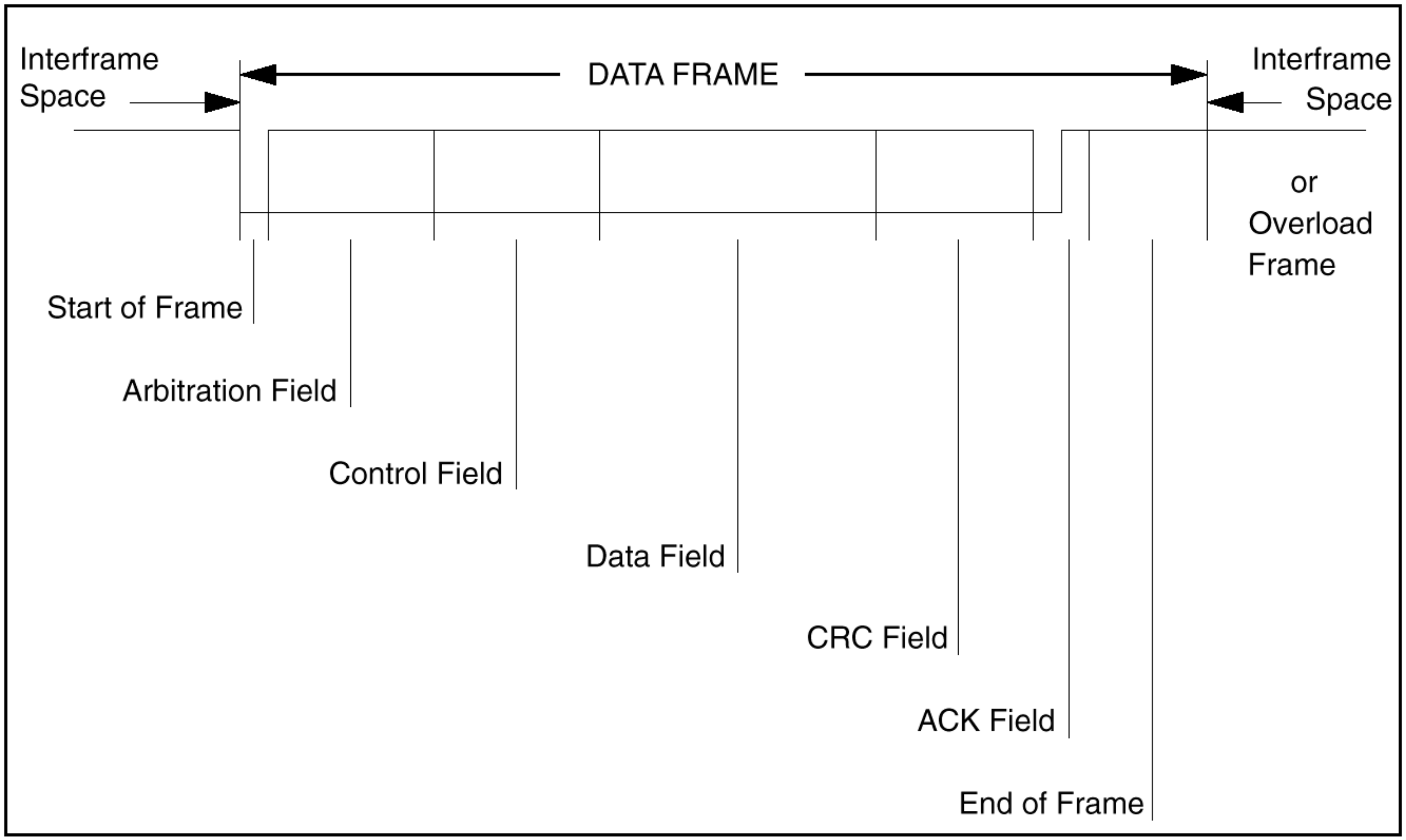
- **Header**
  - Routing information (source, destination)
  - Global priority information (which message gets on bus first?)

- **Data**
  - Application- or high-level-standard defined data fields
  - Often only 1-8 bytes

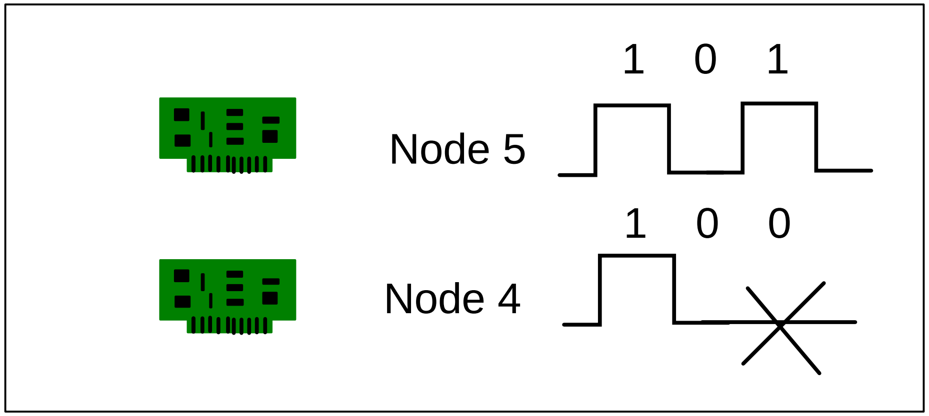
- **Error detection**
  - Detects corrupted data (e.g., using a CRC)
  - Embedded networks can have *very* high bit error rates

CAN Message Format

- **What's inside the message?**
  - "Arbitration Field" = "Message ID"



Binary Countdown (Bit Dominance)

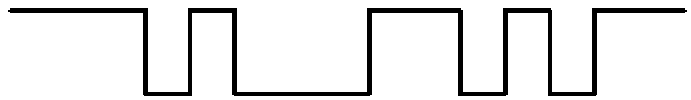


- **Operation**
  - Each node is assigned a unique identification number
  - All nodes wishing to transmit compete for the channel by transmitting a binary signal based on their identification value
  - A node drops out the competition if it detects a dominant state while transmitting a passive state
  - Thus, the node with the highest identification value wins

- **Examples**
  - CAN, SAE J1850

More Detailed Arbitration Example

Two logic states
possible on the bus:
“1” = recessive
“0” = dominant



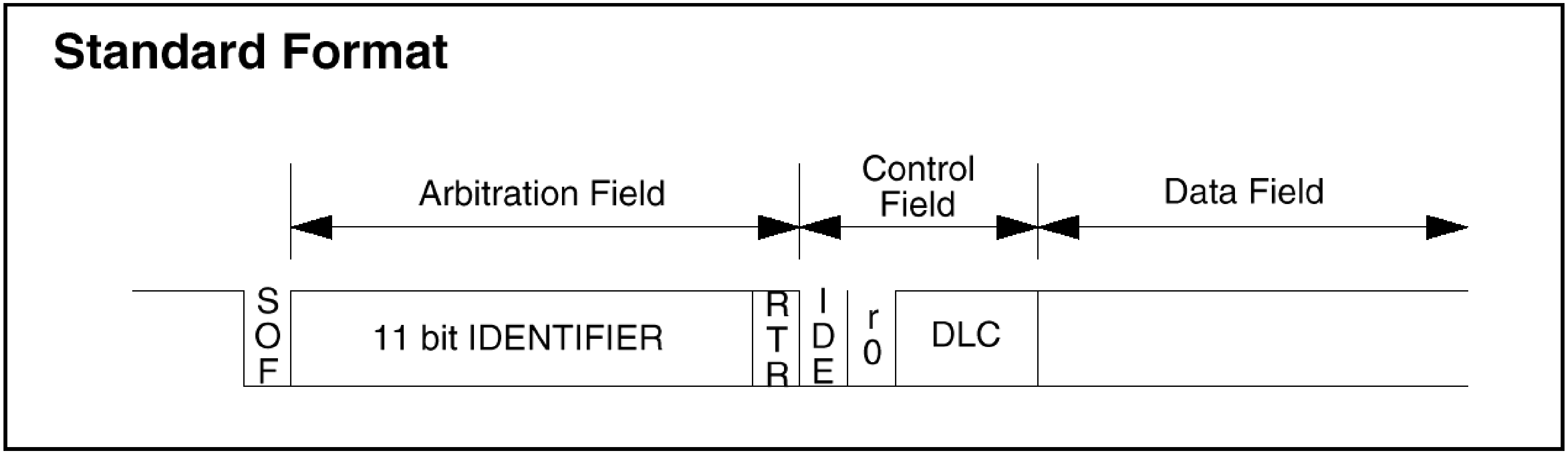
“1” = recessive
“0” = dominant

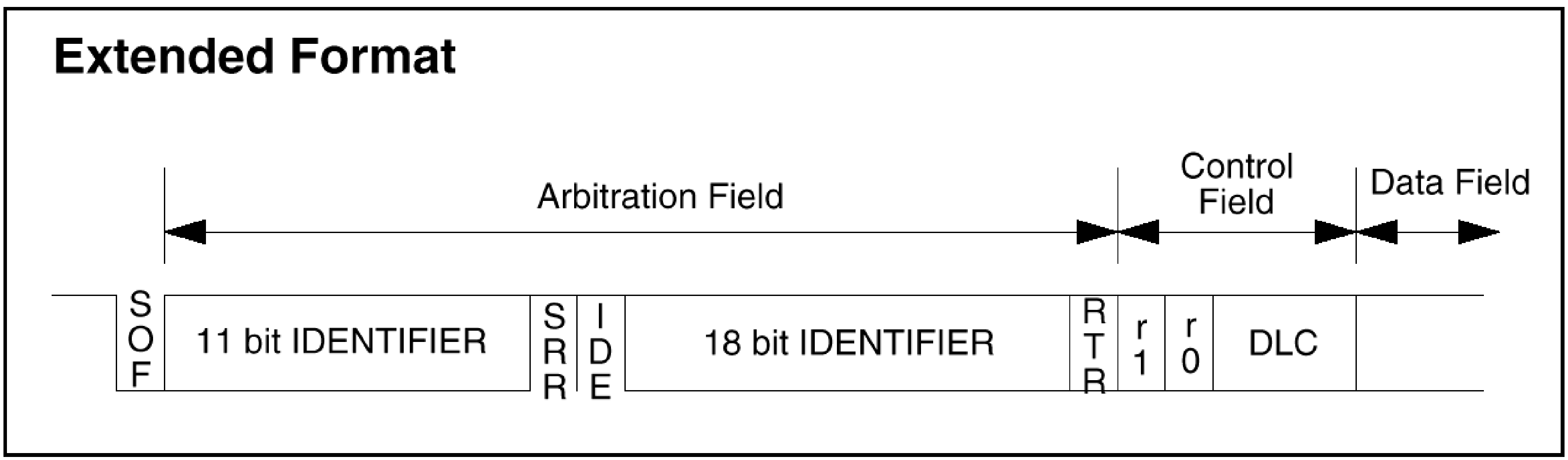
A	B	C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

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

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

Two Sizes of CAN Arbitration Fields

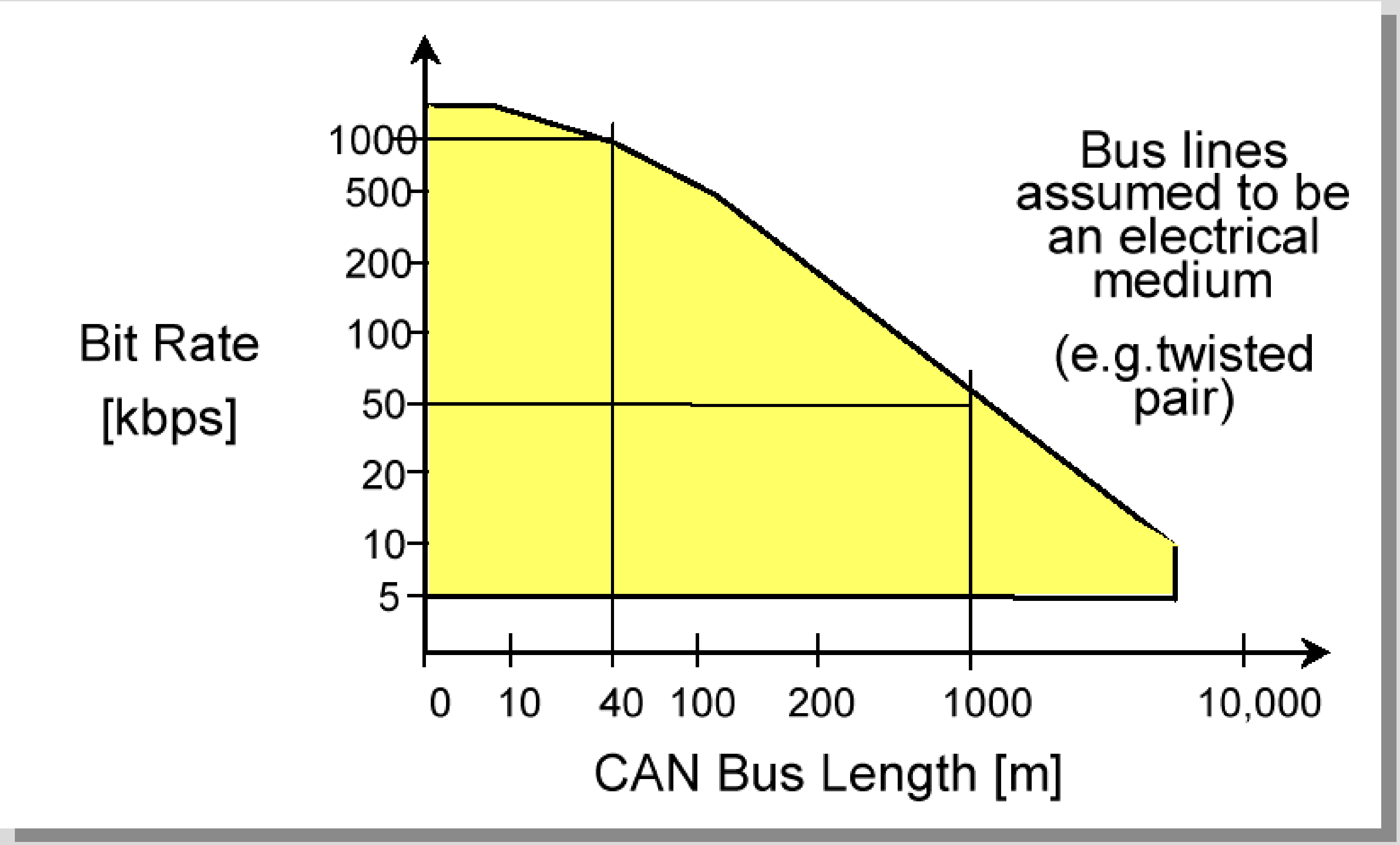




Arbitration Limits Network Size

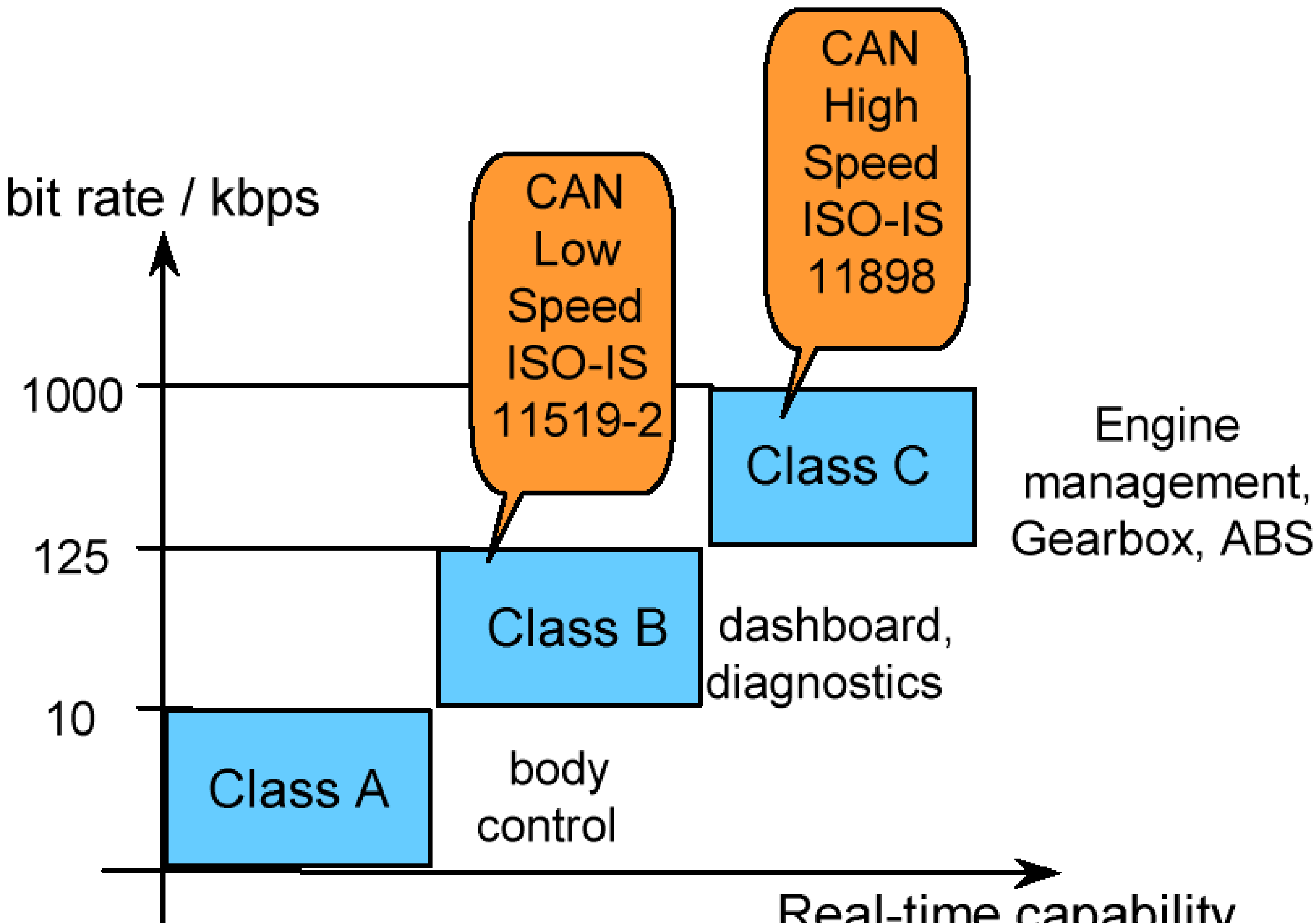
- Need $2*t_{pd}$ per bit maximum speed

- Up to 1Mbit / sec @40m bus length (130 feet)



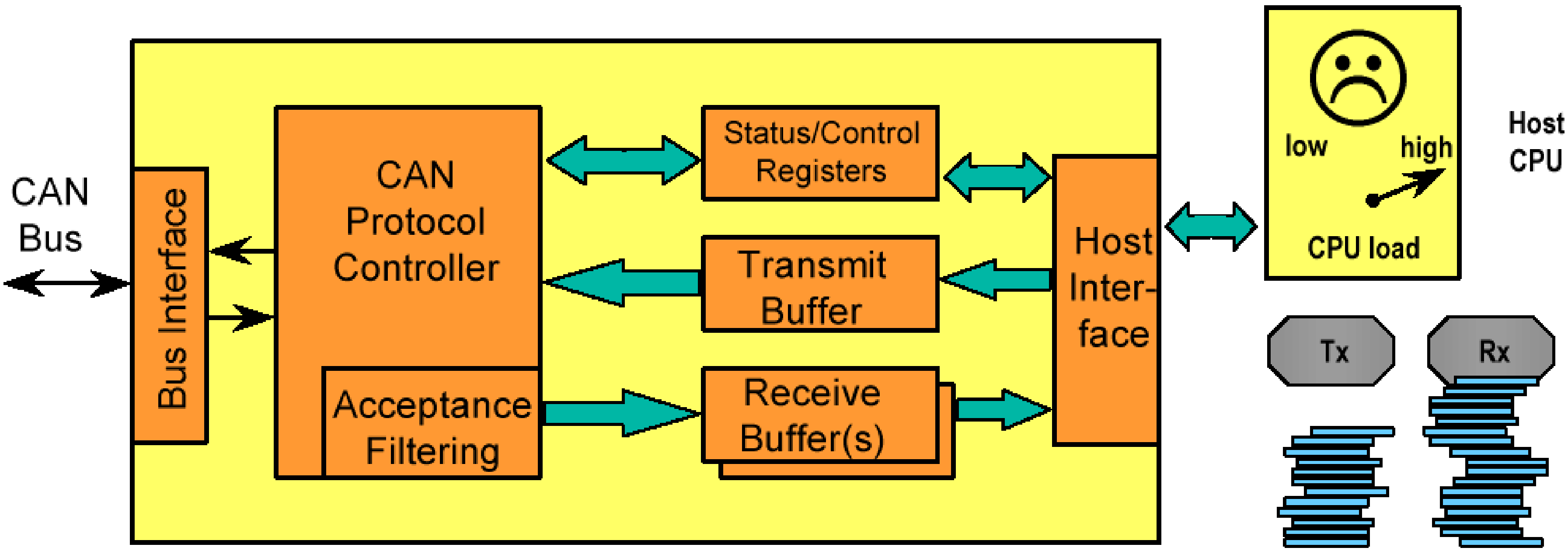
SAE Message Classes

- **Fast tends to correlate with critical control**
  - But, this is not always true; just often true



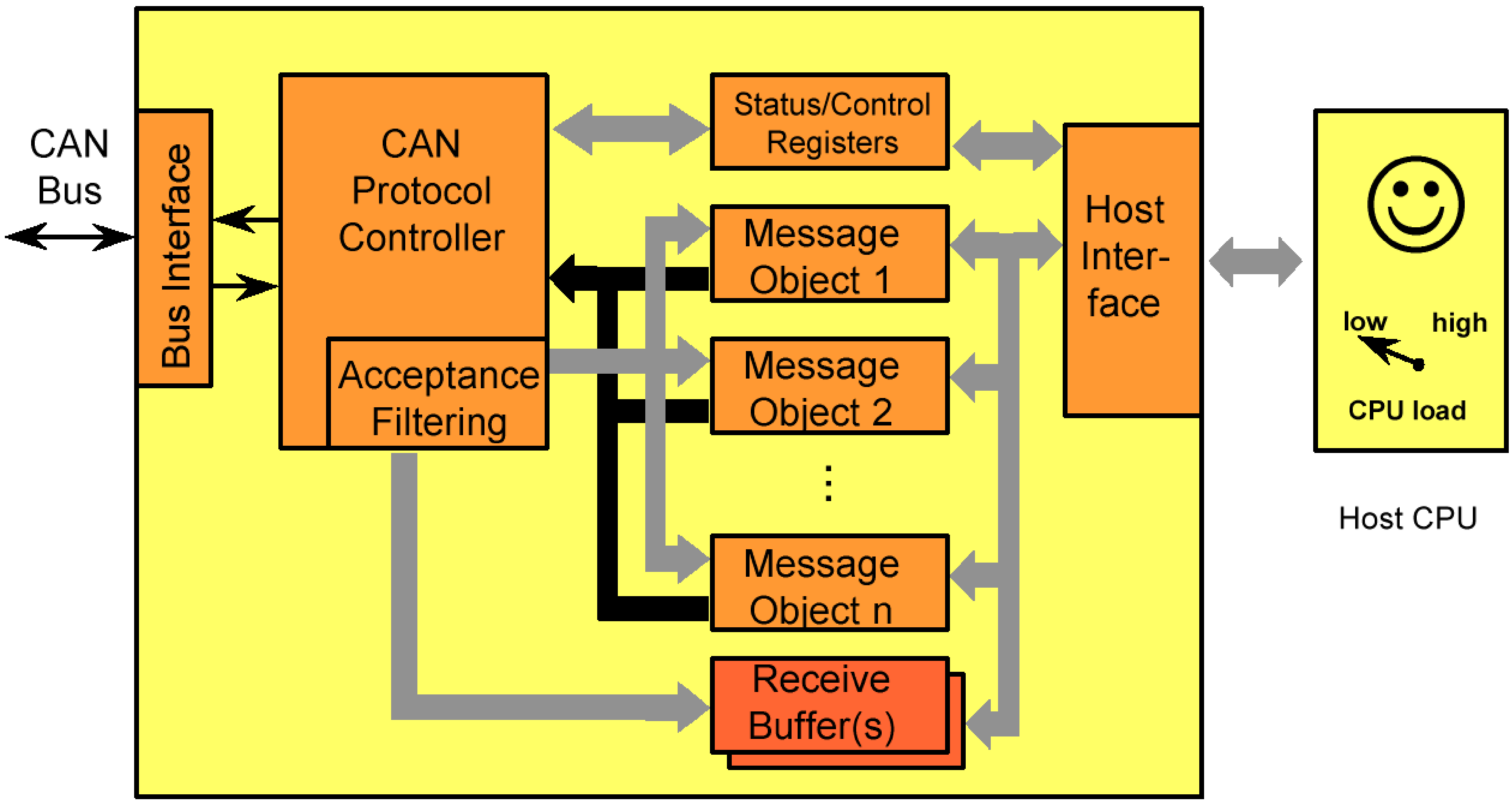
Basic CAN Controller (Don’t Use This One)

- **“Cheap” node**
  - Could get over-run with messages even if it didn’t need them



Full CAN Controller

- Hardware message filters sort & filter messages without interrupting CPU



Mask Registers

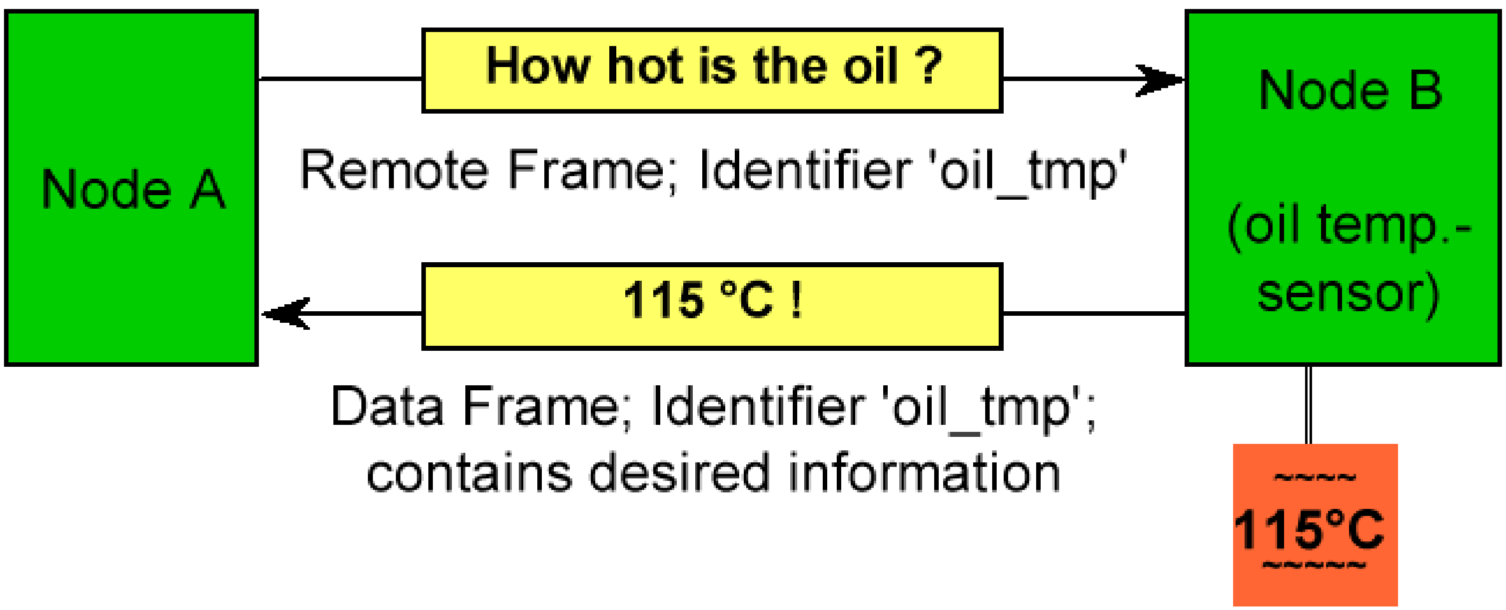
- **Used to set up message filters**
  - Mask register selects bits to examine
  - Object Arbitration register selects bits that must match to be accepted

	10										0	
Mask Register (std ID)	1	1	1	1	1	1	1	0	1	1	0	(= 0x7f6)
Message Object Arbitration Register	1	0	0	1	0	0	1	0	0	0	0	(= 0x490)
Resulting ID matches (X = don't care)	1	0	0	1	0	0	1	X	0	0	X	
ID's received:	1	0	0	1	0	0	1	0	0	0	0	(= 0x490)
	1	0	0	1	0	0	1	0	0	0	1	(= 0x491)
	1	0	0	1	0	0	1	1	0	0	0	(= 0x498)
	1	0	0	1	0	0	1	1	0	0	1	(= 0x499)

Various Special Messages

- **Various error messages**

- **Remote Frames –atomic request for data / provide data**
  - (Not used in most in cars)



DeviceNet

- **One of several higher-level protocols**
  - Based on top of CAN
  - Used for industrial control (valves, motor starters, display panels, …)
    - Caterpillar is a member of ODVA as well (Open DeviceNet Vendors Assn.), but for factory automation.

- **Basic ideas:**
  - CAN is used in high volumes = cheaper network chips than competitors
  - Use structured approach to message formats to standardize operation

- **Does *NOT* standardize specific message contents**
  - But it does specify a hierarchy of message ID formats

DeviceNet Message ID Scheme

Message Identifier Bits

10	9	8	7	6	5	4	3	2	1	0	**Hex Range**	**Identity Usage**
0	Message ID				Source Node #						000 - 3ff	Group 1
1	0	Source Node #						Msg ID			400 - 5ff	Group 2
1	1	Msg ID (0..6)			Source Node #						600 - 7bf	Group 3
1	1	1	1	1	Message ID (0..2f)						7c0 - 7ef	Group 4
10	1	1	1	1	1	1	X	X	X	X	7f0 - 7ff	Invalid

DeviceNet Group Strategy

- **Group 1**
  - Prioritized by Message ID / Node number
  - High priority messages with fairness to nodes

- **Group 2**
  - Prioritized by Node number / Message ID
  - Gives nodes priority

- **Group 3**
  - Essentially same as Group 1, but allows Group 2 to have higher priority

- **Group 4**
  - Global housekeeping messages / must be unique in system (no node number)

Other Approaches Are Possible

- **And, you can invent your own too…**

- **Variations include:**
  - Automatic assignment of node numbers (include hot-swap)
  - Automatic assignment of message numbers (include hot-swap)
  - Mixes of node-based vs. message-ID based headers

CAN Workloads – Spreadsheets

- **"SAE Standard Workload" (53 messages)** V/C = Vehicle Controller

Signal Number	*Signal Description*	*Size /bits*	*J /ms*	*T /ms*	*Periodic /Sporadic*	*D /ms*	*From*	*To*
1	Traction Battery Voltage	8	0.6	100.0	P	100.0	Battery	V/C
2	Traction Battery Current	8	0.7	100.0	P	100.0	Battery	V/C
3	Traction Battery Temp, Average	8	1.0	1000.0	P	1000.0	Battery	V/C
4	Auxiliary Battery Voltage	8	0.8	100.0	P	100.0	Battery	V/C
5	Traction Battery Temp, Max.	8	1.1	1000.0	P	1000.0	Battery	V/C
6	Auxiliary Battery Current	8	0.9	100.0	P	100.0	Battery	V/C
7	Accelerator Position	8	0.1	5.0	P	5.0	Driver	V/C
8	Brake Pressure, Master Cylinder	8	0.1	5.0	P	5.0	Brakes	V/C
9	Brake Pressure, Line	8	0.2	5.0	P	5.0	Brakes	V/C
10	Transaxle Lubrication Pressure	8	0.2	100.0	P	100.0	Trans	V/C
11	Transaction Clutch Line Pressure	8	0.1	5.0	P	5.0	Trans	V/C
12	Vehicle Speed	8	0.4	100.0	P	100.0	Brakes	V/C
13	Traction Battery Ground Fault	1	1.2	1000.0	P	1000.0	Battery	V/C
14	Hi&Lo Contactor Open/Close	4	0.1	50.0	S	5.0	Battery	V/C
15	Key Switch Run	1	0.2	50.0	S	20.0	Driver	V/C
16	Key Switch Start	1	0.3	50.0	S	20.0	Driver	V/C
17	Accelerator Switch	2	0.4	50.0	S	20.0	Driver	V/C
18	Brake Switch	1	0.3	20.0	S	20.0	Brakes	V/C
19	Emergency Brake	1	0.5	50.0	S	20.0	Driver	V/C
20	Shift Lever (PRNDL)	3	0.6	50.0	S	20.0	Driver	V/C
21	Motor/Trans Over Temperature	2	0.3	1000.0	P	1000.0	Trans	V/C
22	Speed Control	3	0.7	50.0	S	20.0	Driver	V/C
23	12V Power Ack Vehicle Control	1	0.2	50.0	S	20.0	Battery	V/C
24	12V Power Ack Inverter	1	0.3	50.0	S	20.0	Battery	V/C
25	12V Power Ack I/M Contr.	1	0.4	50.0	S	20.0	Battery	V/C
26	Brake Mode (Parallel/Split)	1	0.8	50.0	S	20.0	Driver	V/C

Why Use An Embedded Network

- **Potential Advantages (for CAN?)**
  - Reduces wires and increases reliability
  - Lowers weight, size, and installation costs
  - Logical choice for physically distributed systems
  - Allows sharing of system resources
  - Increases system capability and flexibility
  - Self-configuration, self-installation, and advanced diagnostics
  - Foundation for system integration and automation
  - Integrated, modular product line leads to interoperability

- **Potential Network Drawbacks (for CAN?)**
  - May initially increase product cost
  - Requires knowledge and new skills in networking
  - Requires special tools for fault detection

CAN Tradeoffs

- **Advantages**
  - High throughput under light loads
  - Local and global prioritization possible
  - Arbitration is part of the message - low overhead

- **Disadvantages**
  - Propagation delay limits bus length (2 t_{pd} bit length)
  - Unfair access - node with a high priority can "hog" the network
    - Can be reduced in severity with Message + Node # prioritization
  - Poor latency for low priority nodes
    - Starvation is possible

- **Optimized for:**
  - Moderately large number of message types
  - Arbitration overhead is constant
  - Global prioritization (*but* limited mechanisms for fairness)

Review

- **Controller Area Network**
  - Binary-countdown arbitration
  - Standard used in automotive & industrial control

- **CAN Tradeoffs**
  - Good at global priority (but difficult to be “fair”)
  - Efficient use of bandwidth
  - Requires bit-dominance in physical layer
  - Message filters are required to keep small nodes from being overloaded
    - (But, these are easy to implement)

- **Next lecture: Protocol building blocks (custom protocols)**