

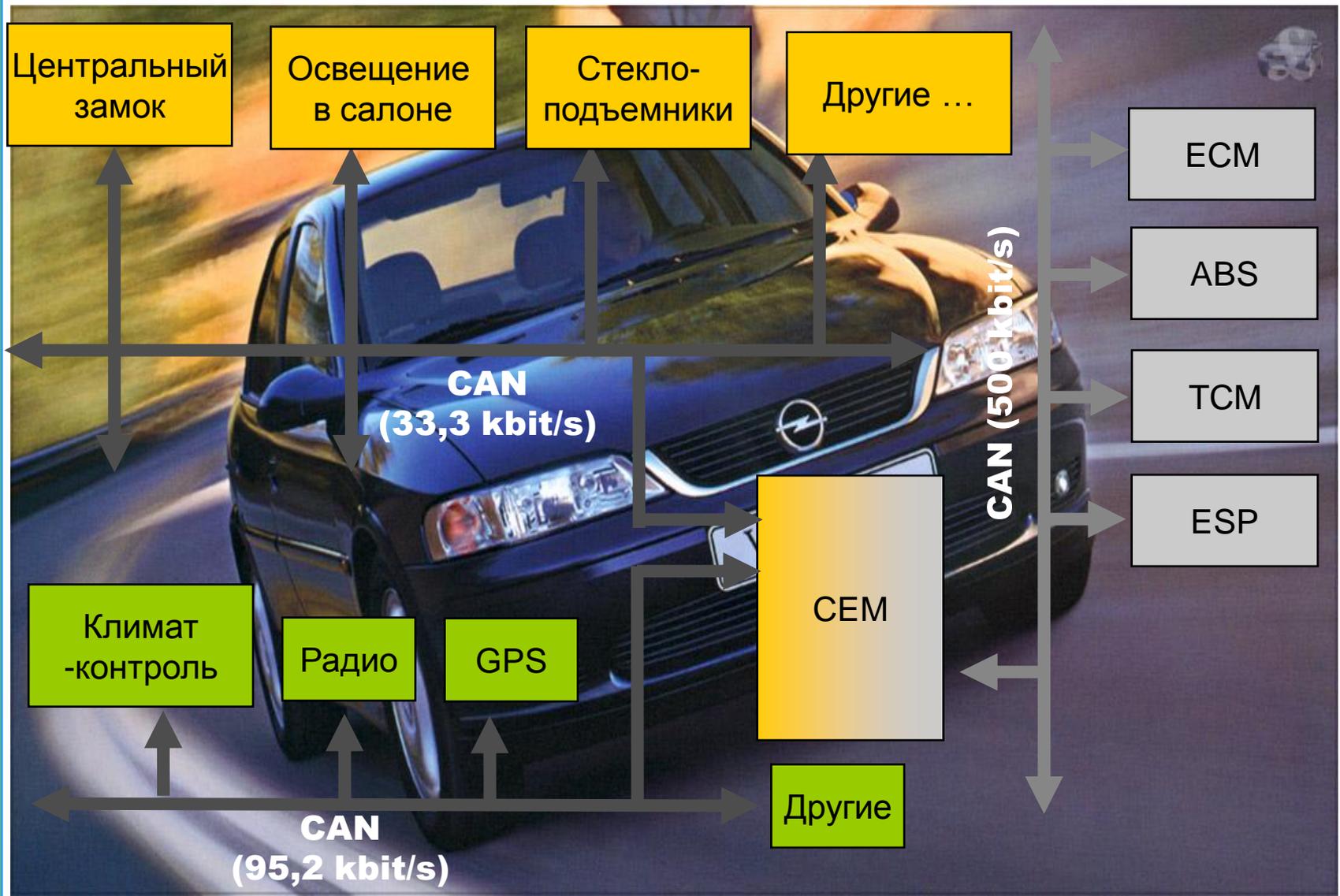
**"Промышленные сети распределенных систем управления"**  
**CAN-технология – как основа для построения современных  
интегрированных систем управления**  
*Чепурнов А.С. (НИИЯФ им. Д.В. Скобельцына МГУ им. М.В. Ломоносова)*



# Особенности шины CAN

- протокол имеет свойства мультимастера
- основана на механизме недеструктивного арбитража, который гарантирует доступ к шине сообщений с максимальным приоритетом без всяких задержек
- отсутствует центральный мастер шины
- специальные, оригинальные механизмы детектирования ошибок были разработаны специально для этого протокола
- обработка ошибок предполагала автоматическое отключение от шины ошибочно работающих узлов для сохранения связи оставшихся работающих узлов
- передаваемые сообщения никак не идентифицировались по номеру узла сети, который это сообщение передает или принимает, а только внутренним содержанием сообщения, в отличие от многих других последовательных сетей того времени

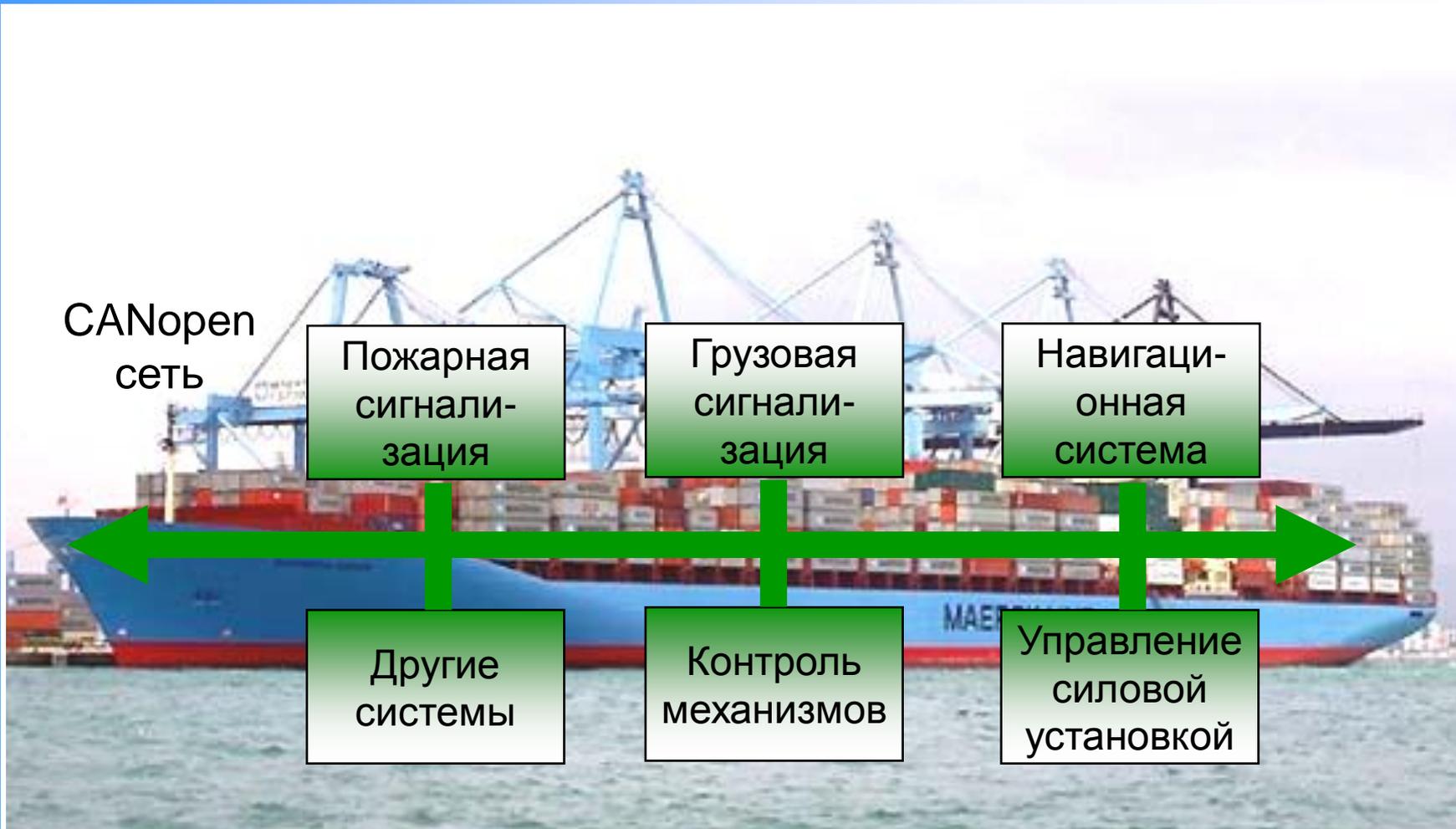
# Opel Vectra (2005)



# Сельскохозяйственный трактор

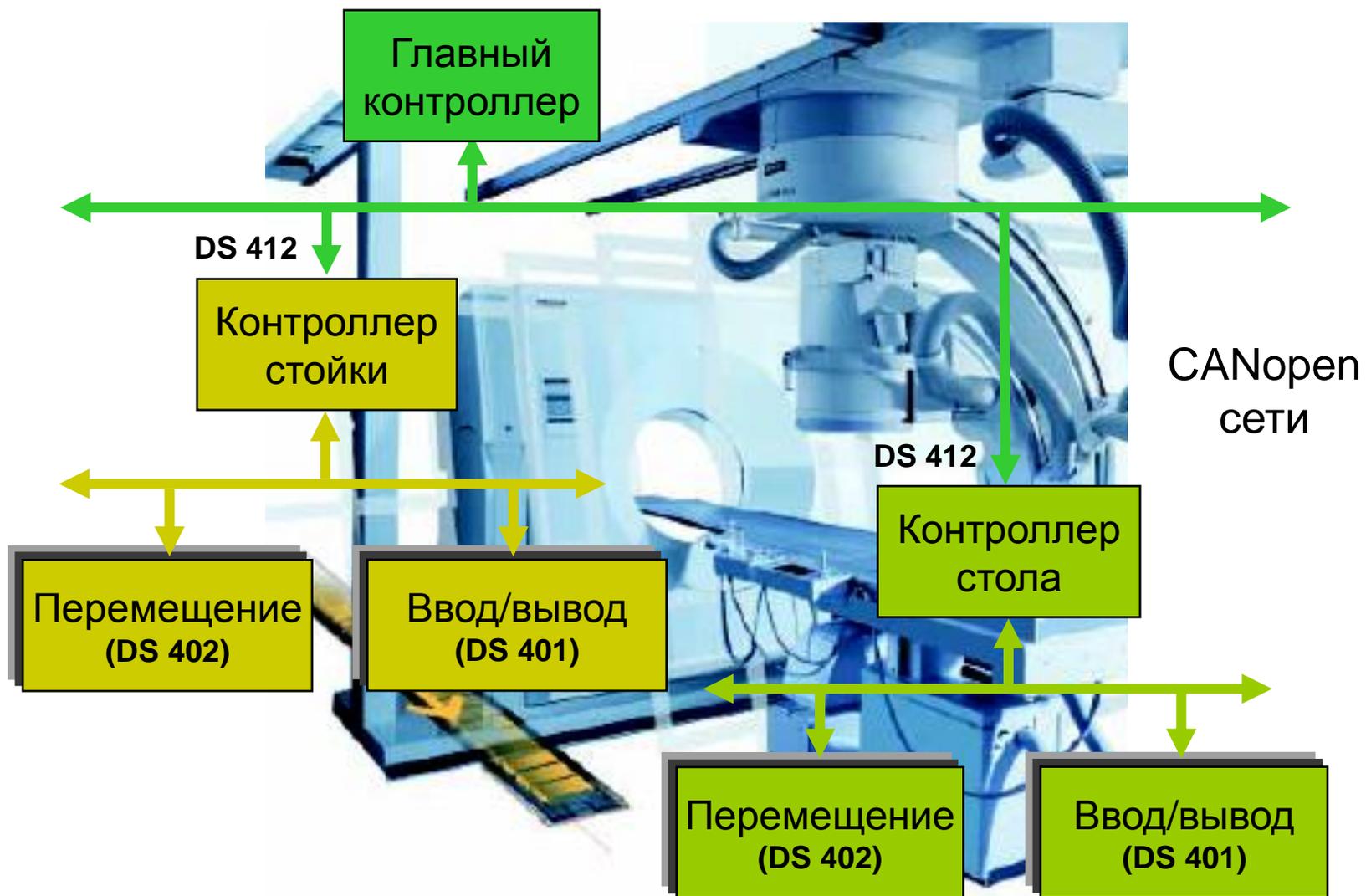


# Морская электроника



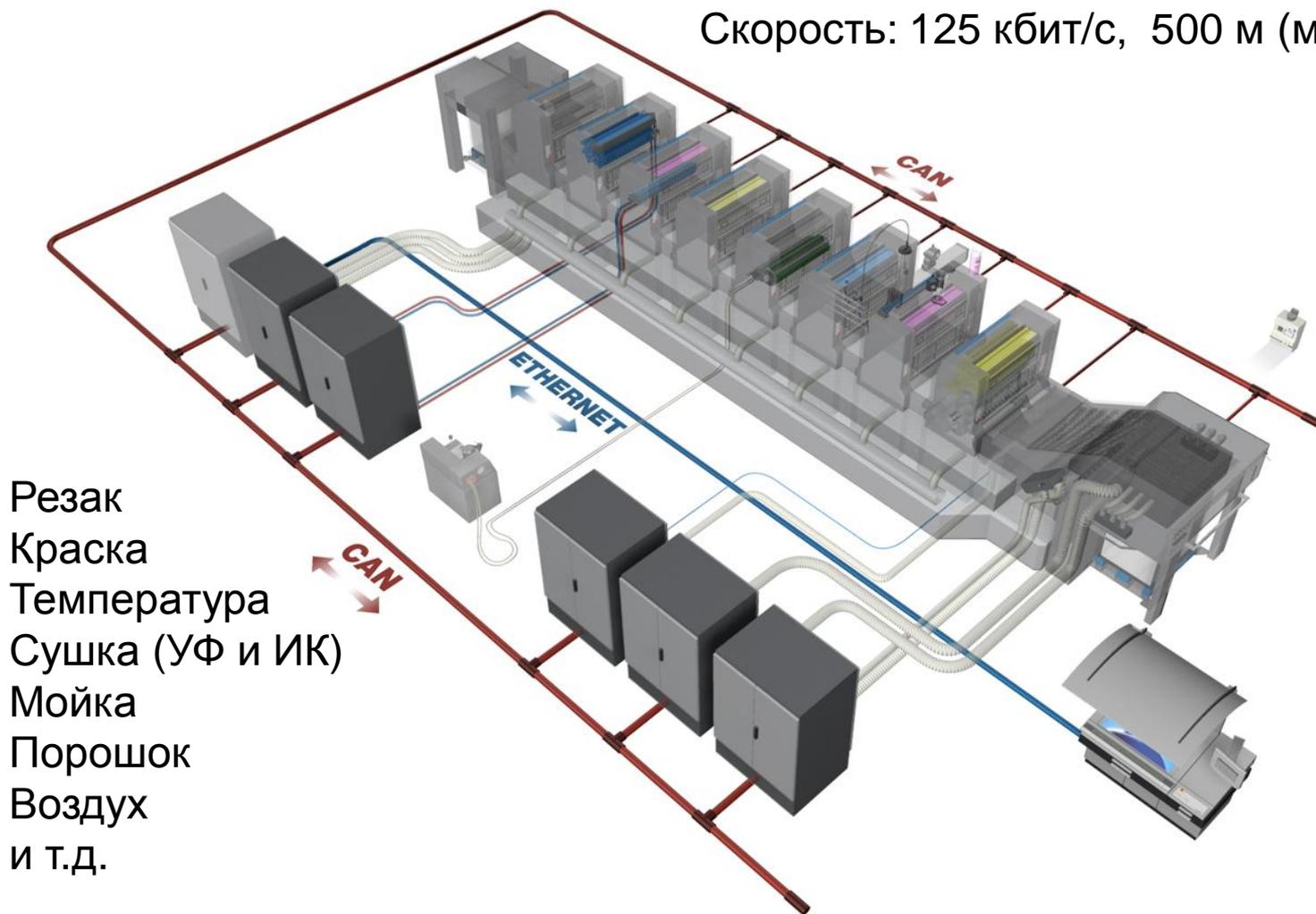
**СiA DSP 304: спецификация протокола безопасного CANopen для морской электроники**

# Медицинское оборудование



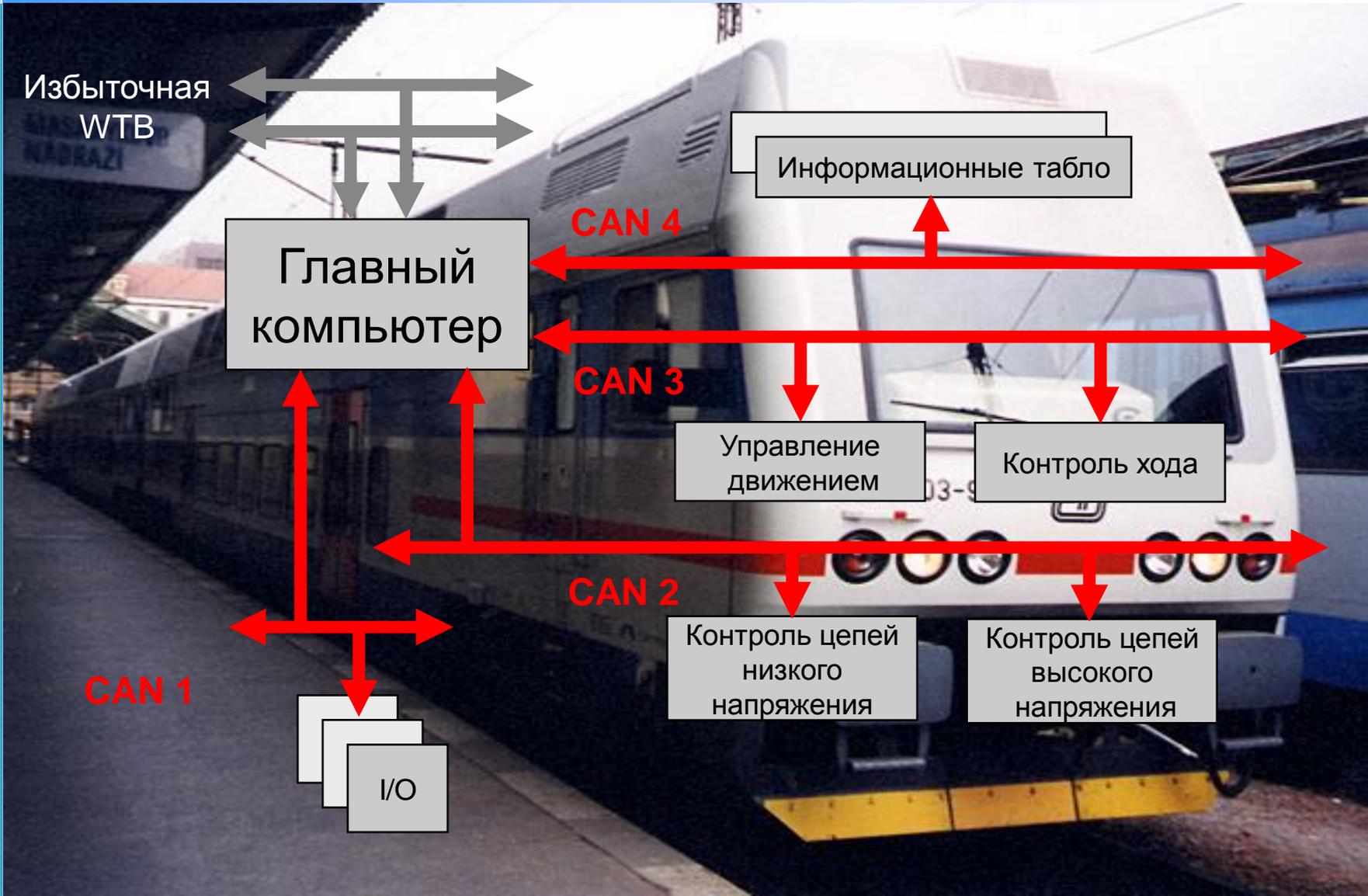
# Машина офсетной печати

Скорость: 125 кбит/с, 500 м (макс.)

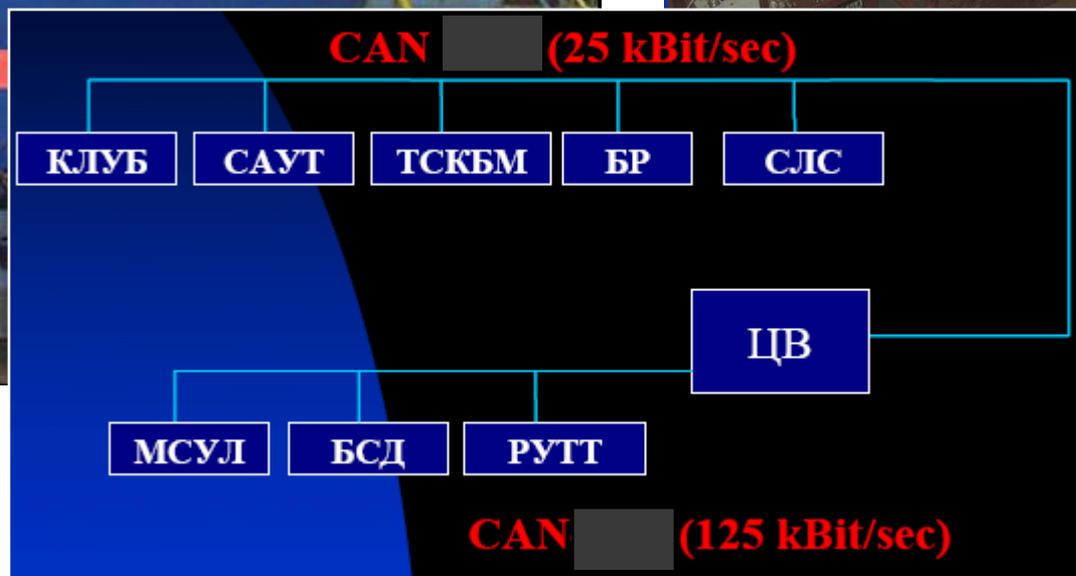


Резак  
Краска  
Температура  
Сушка (УФ и ИК)  
Мойка  
Порошок  
Воздух  
и т.д.

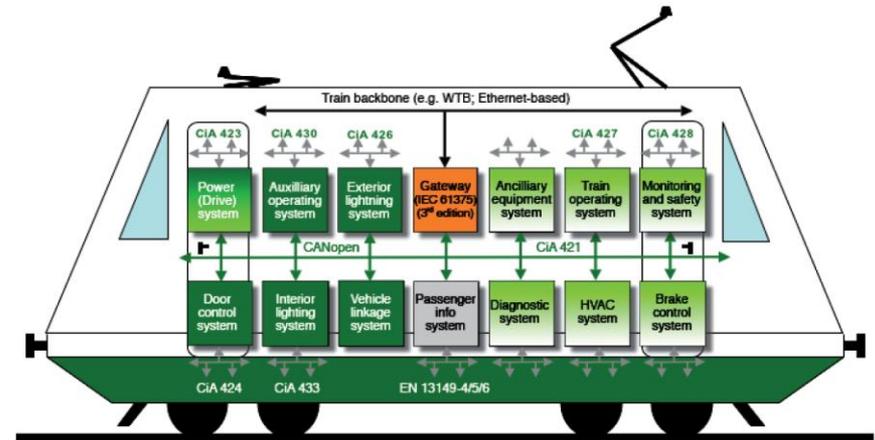
# Пригородный поезд EM471



# Унифицированная кабина машиниста



# ЭП-20 ЭЛЕКТРОВОЗ МАГИСТРАЛЬНЫЙ ПАССАЖИРСКИЙ ДВОЙНОГО ПИТАНИЯ 25 кВ/3 кВ



Архитектура системы управления ж/д поездом на основе CANopen в соответствии с DIN25002-2

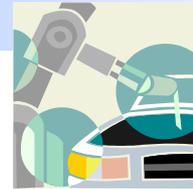
Май 2009 г. - принято решение о применении CANopen в системе управления ЭП-20



# CAN: везде и всюду



- Легковые автомобили
- Грузовики и автобусы
- Оборудование грузовиков
- Внедорожные машины
- Автодиагностика



- Автоматизация предприятий
- Производственные линии



- Ж-д транспорт
- Ж-д сигнализация



- Управление промышленным оборудованием



- Морская электроника
- Береговое оборудование



- Медицинские приложения
- Операционные



- Самолеты и вертолеты
- Аэрокосмические применения



- Кондиционирование
- Лифты
- Двери



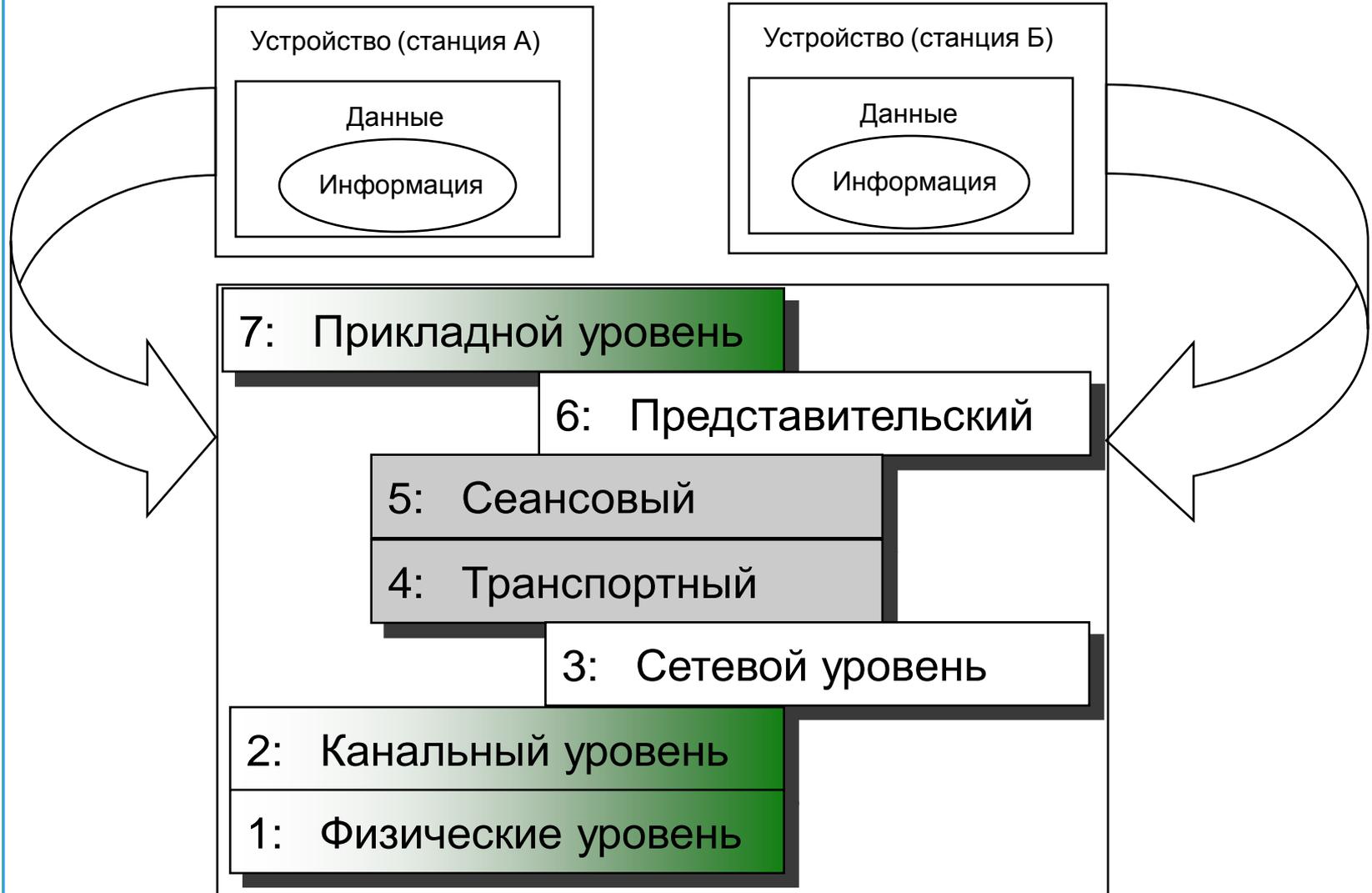
- Подъемные краны
- Асфальтоукладчики



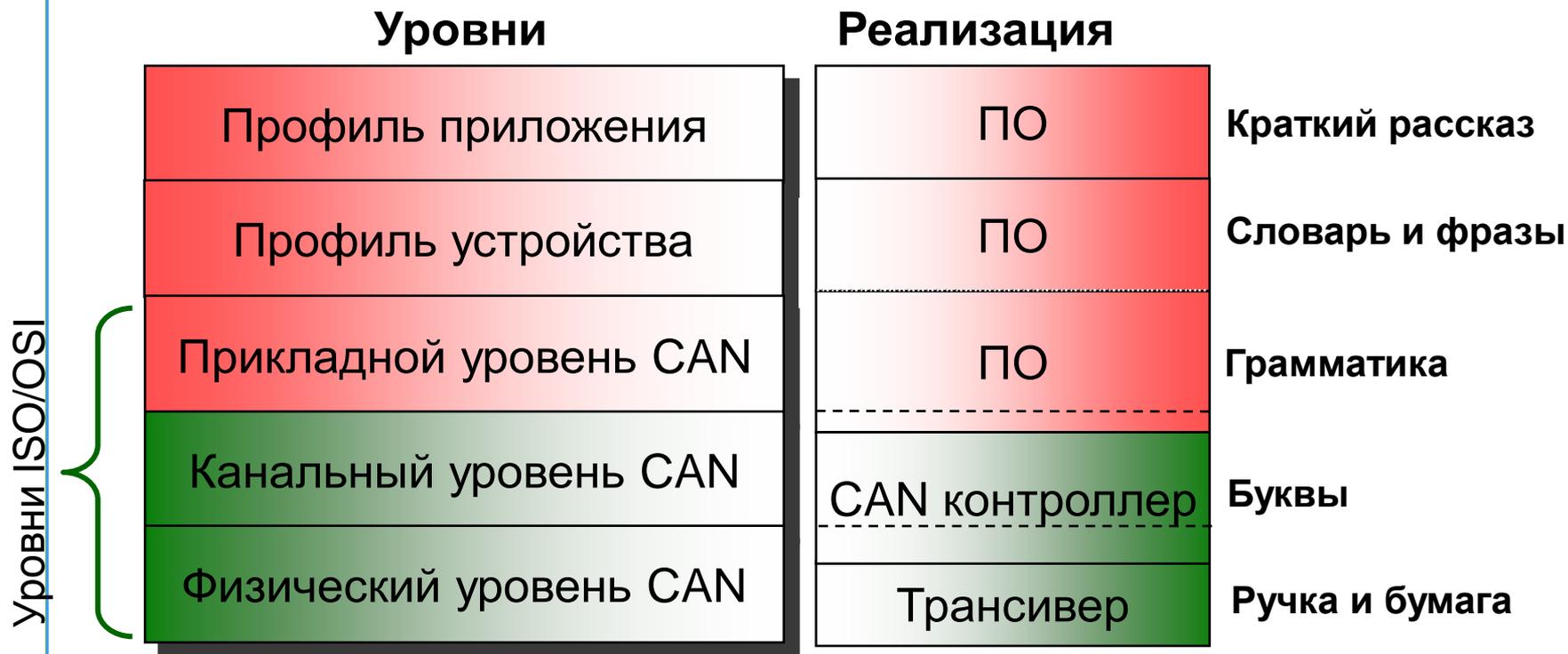
- Встроенные системы управления

# ***Физический и каналный уровни CAN***

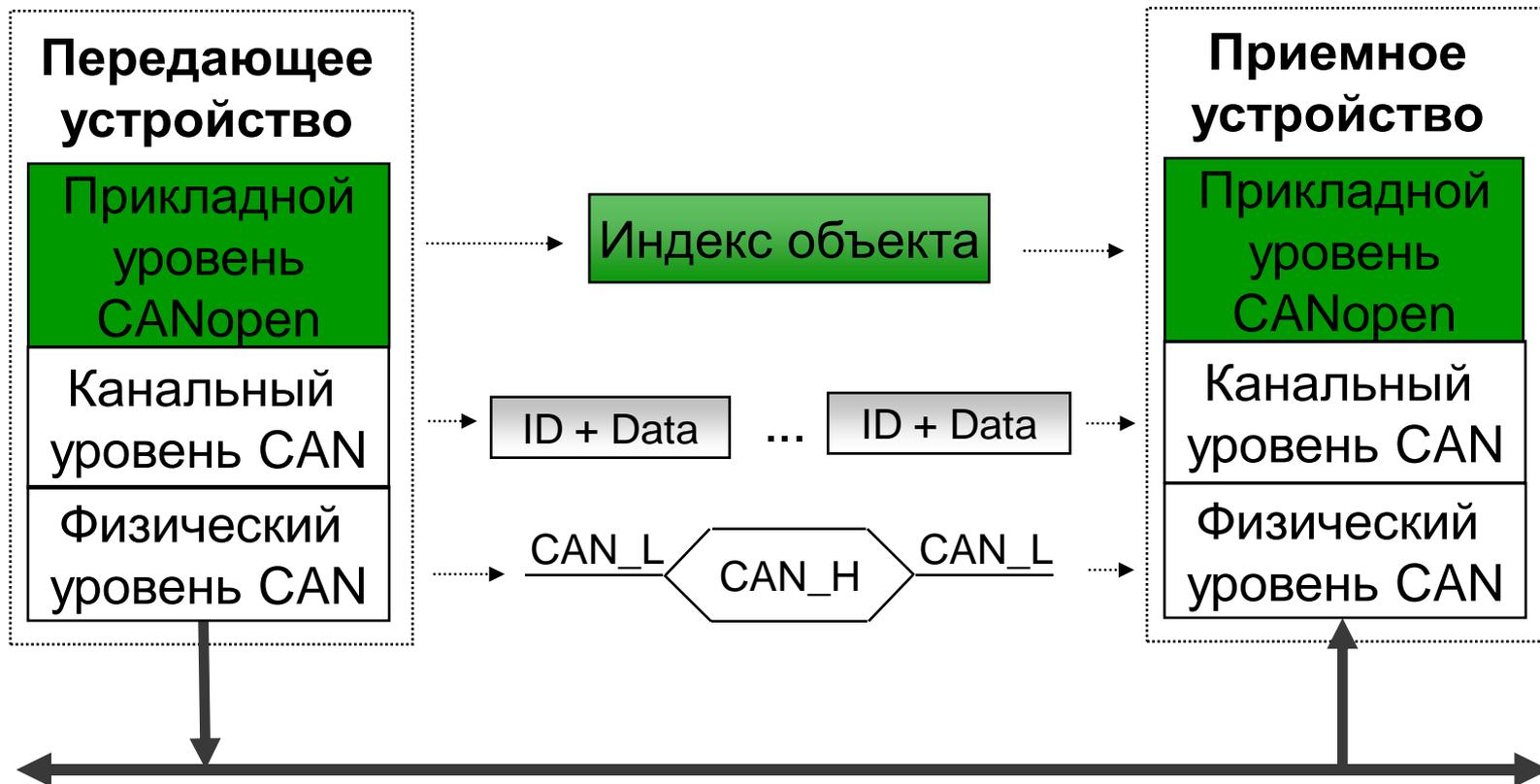
# Эталонная модель ISO/OSI (МС-7498 (1984) ГОСТ 28906-91)



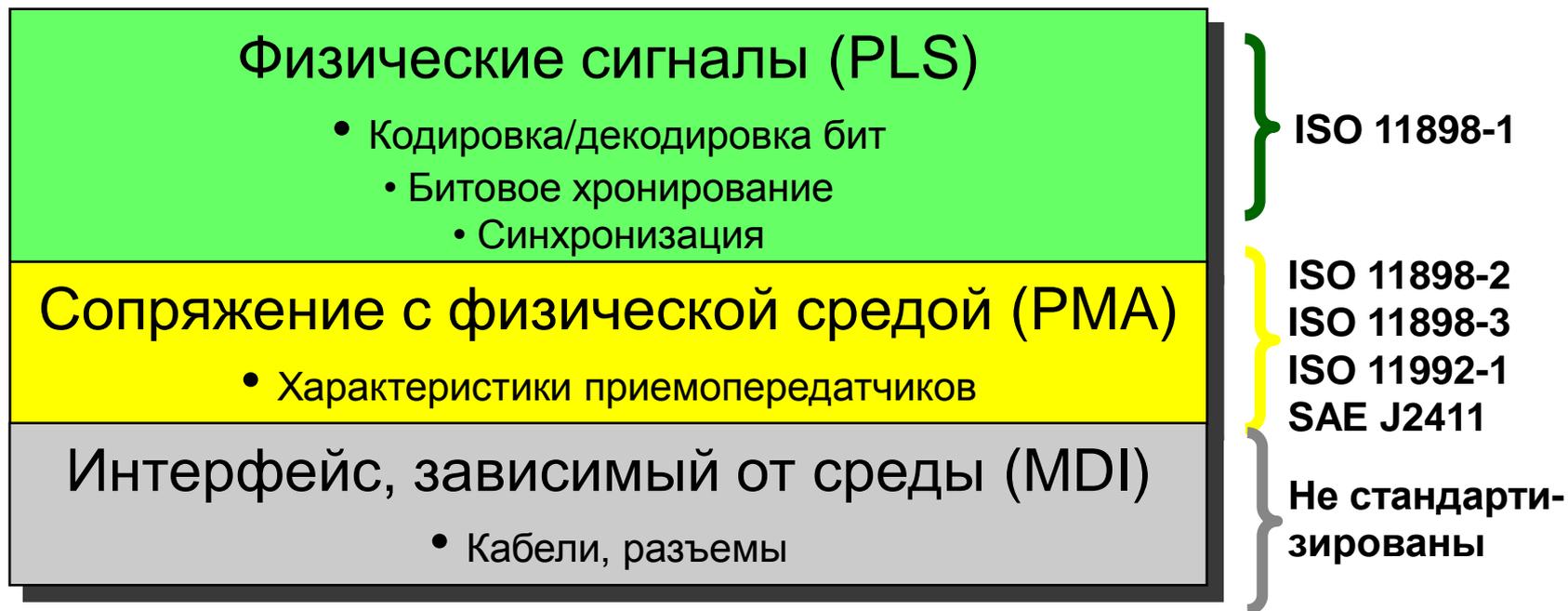
# Коммуникационная модель CAN



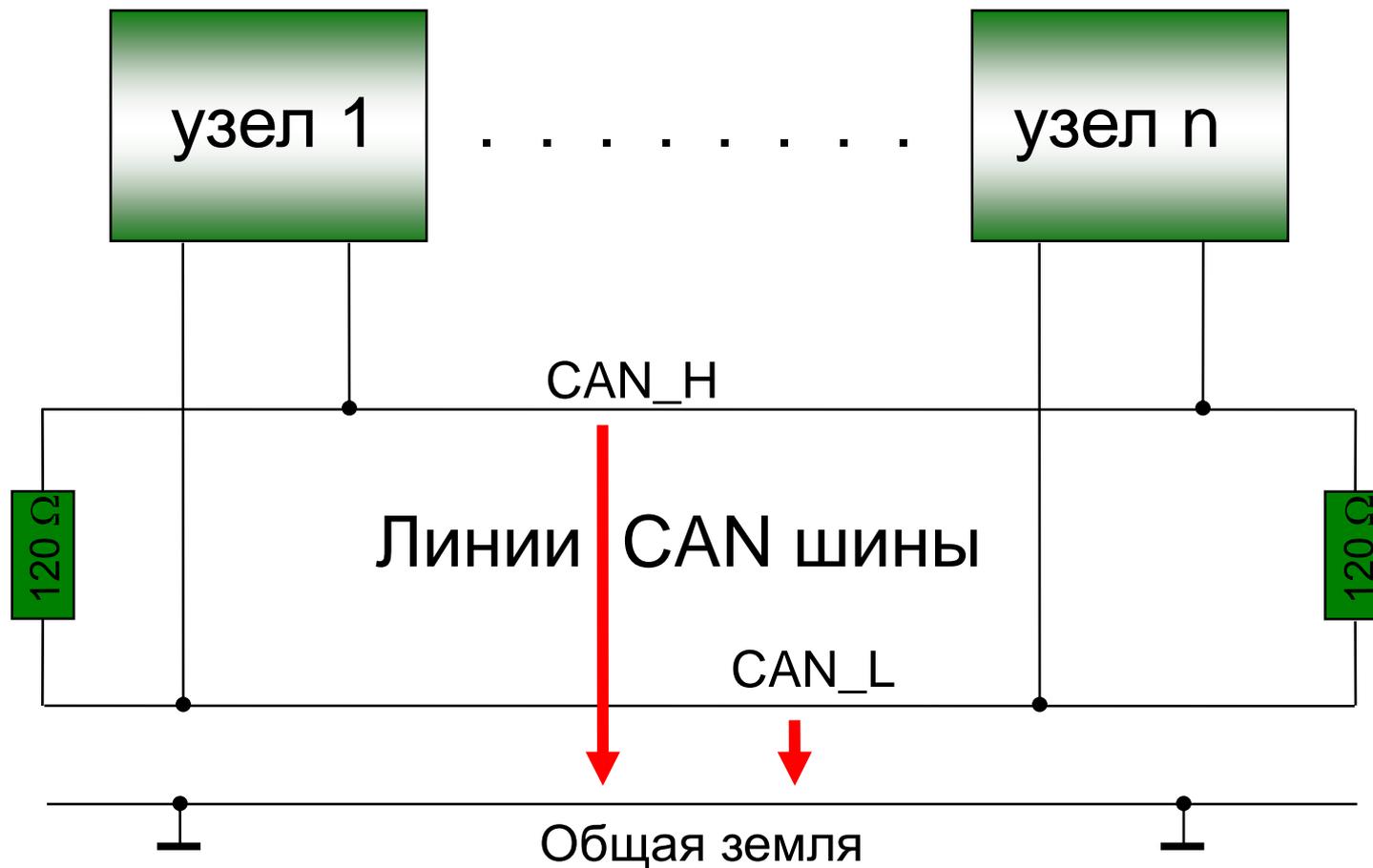
# Взаимодействие уровней протокола



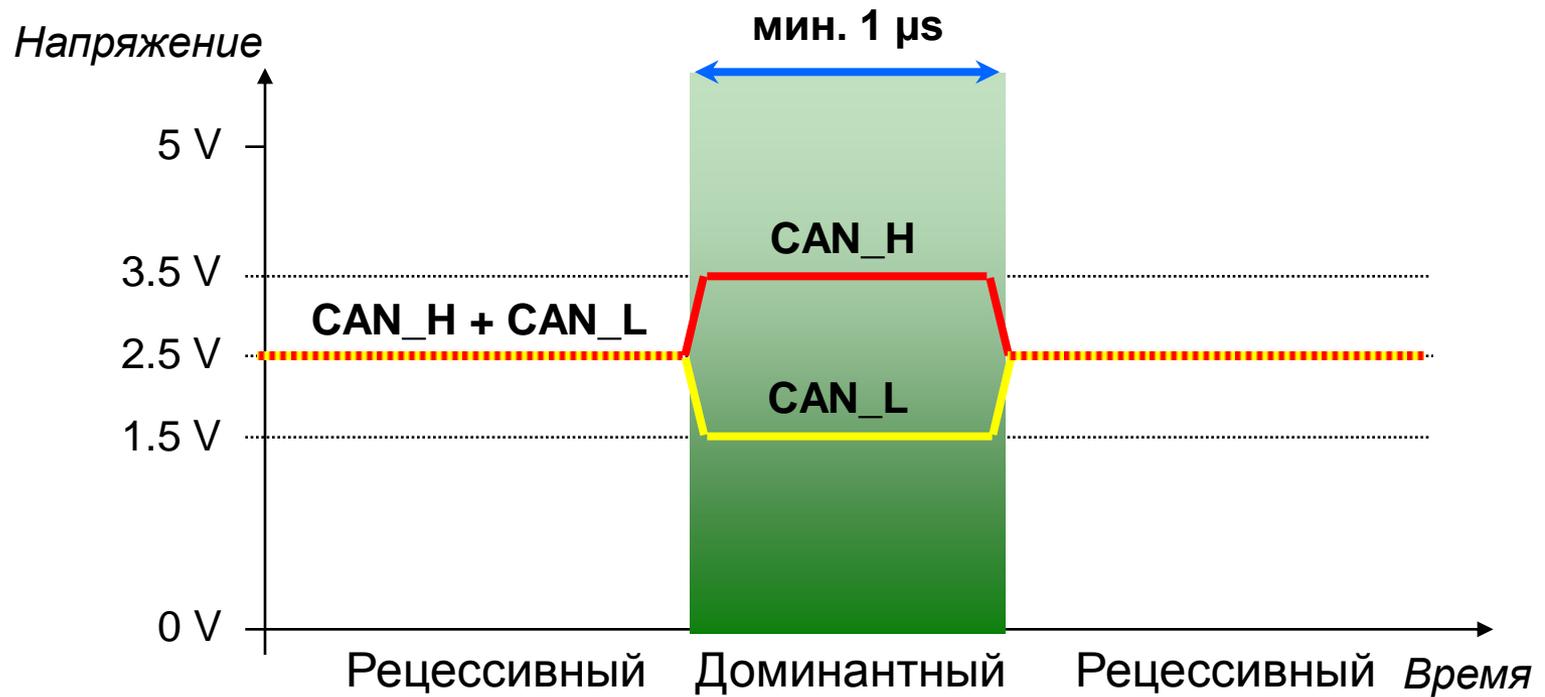
# Уровни физического интерфейса



# Физическая среда ISO 11898-2

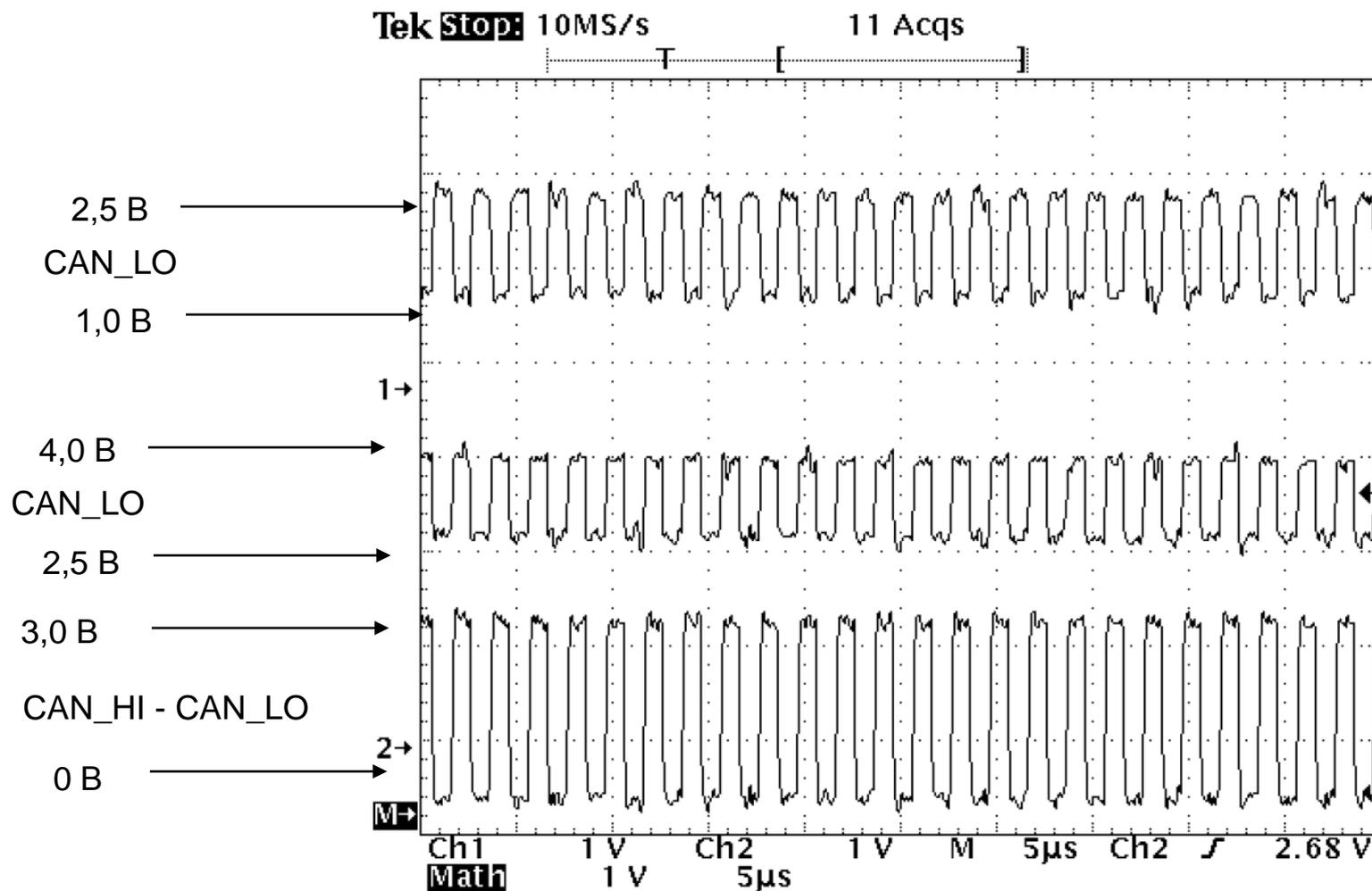


# Уровни сигнала CAN шины

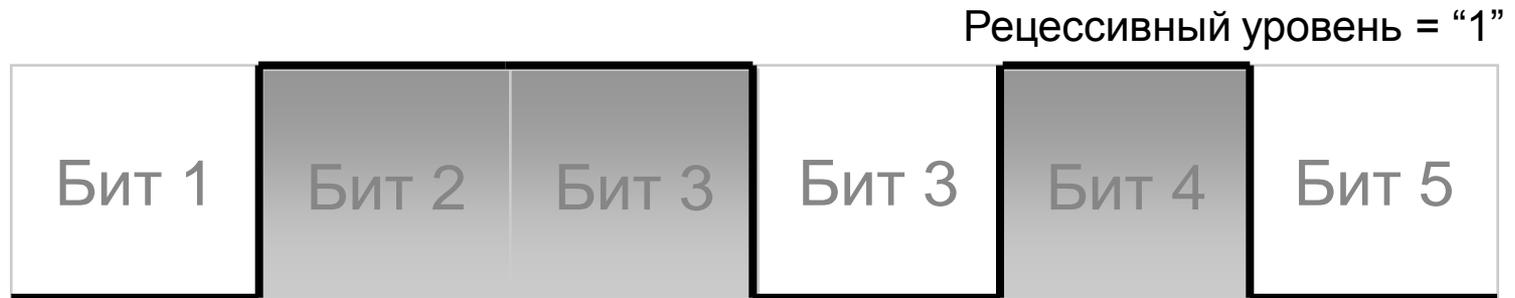


# Уровни сигнала CAN шины

(пример осциллограммы)



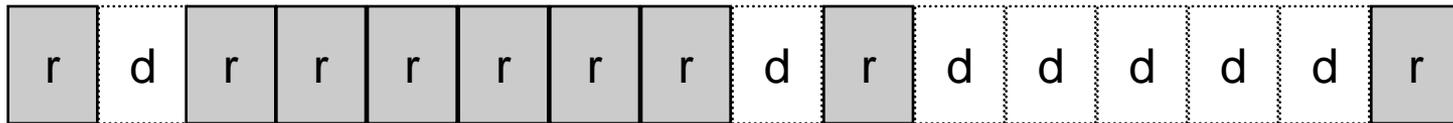
# Кодировка без возврата к нулю (NRZ)



**Замечание:** Уровень NRZ сигнала неизменен в течение каждого битового времени. Это может привести к рассинхронизации различных узлов вследствие смещения опорной частоты.

# Правило бит-стафинга

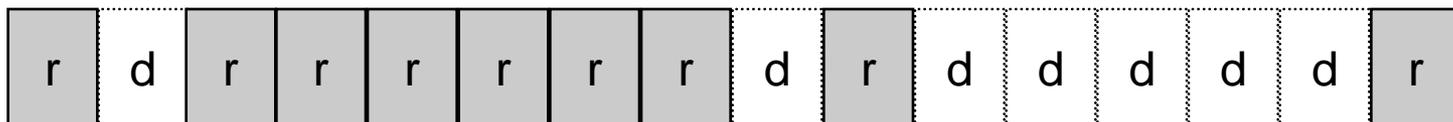
Передаваемая последовательность бит



Вставлены биты стафинга



Принятая после де-стафинга последовательность



# Сегменты битового времени



Sync\_Seg :                    1 квант времени  
Prop\_Seg + Phase\_Seg1: 1 .. 16 квант времени  
Phase\_Seg2:                    1 .. 8 квант времени

# Битовое хронирование, рекомендованное СiА DS-102



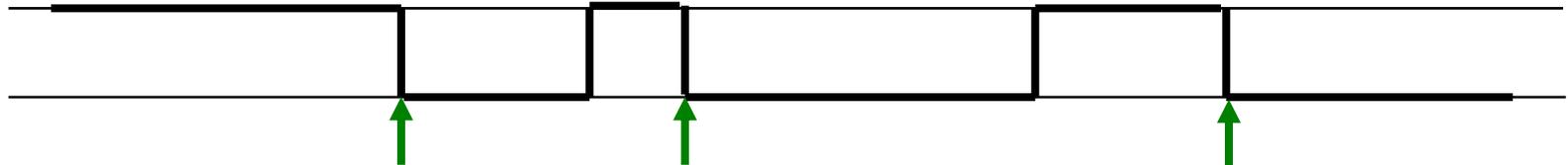
# Битовая синхронизация

- ◆ Жесткая синхронизация по биту начала кадра (SOF) и во время отложенной передачи



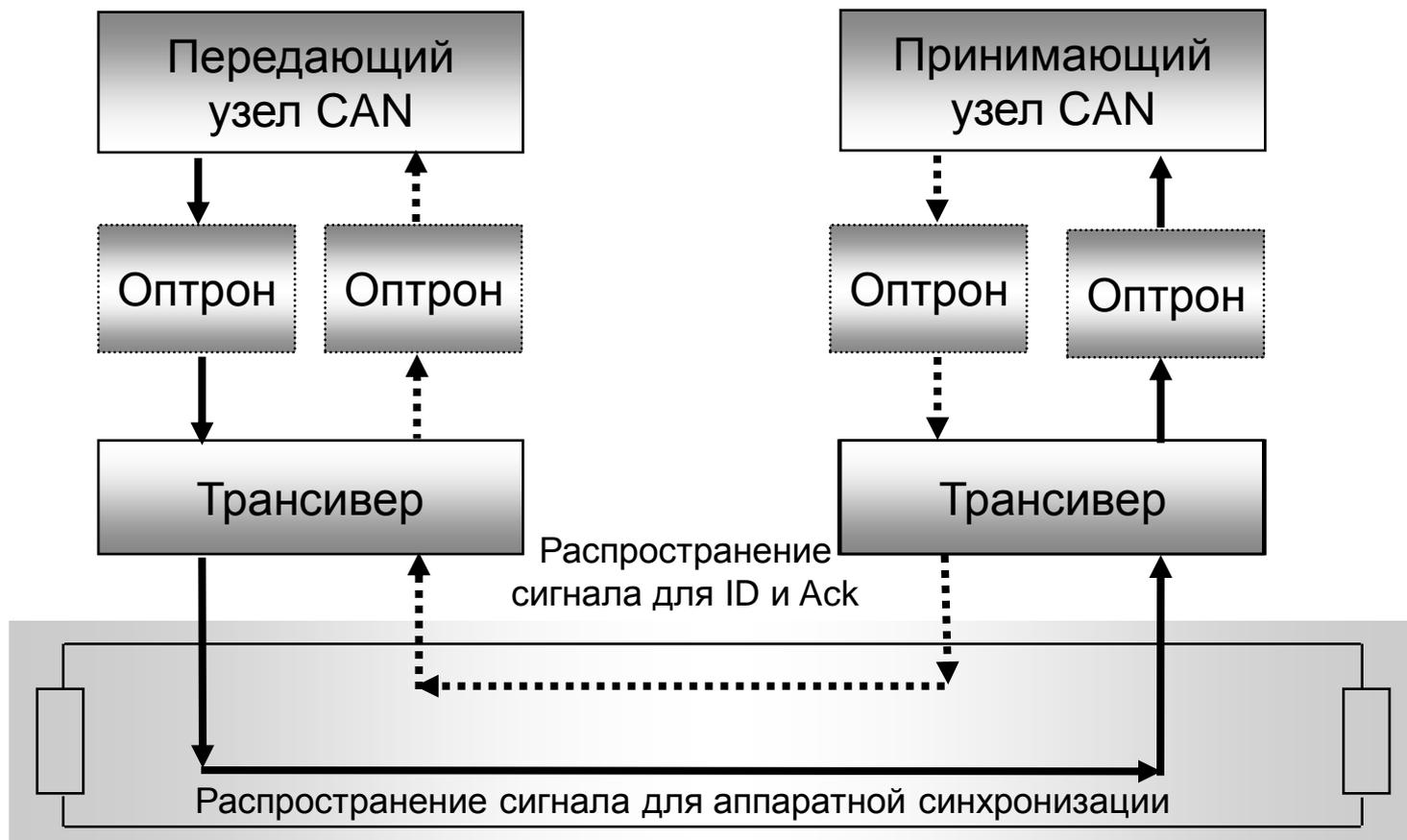
Все узлы синхронизируются по фронту бита SOF

- ◆ Ресинхронизация по каждому рецессивно-доминантному фронту

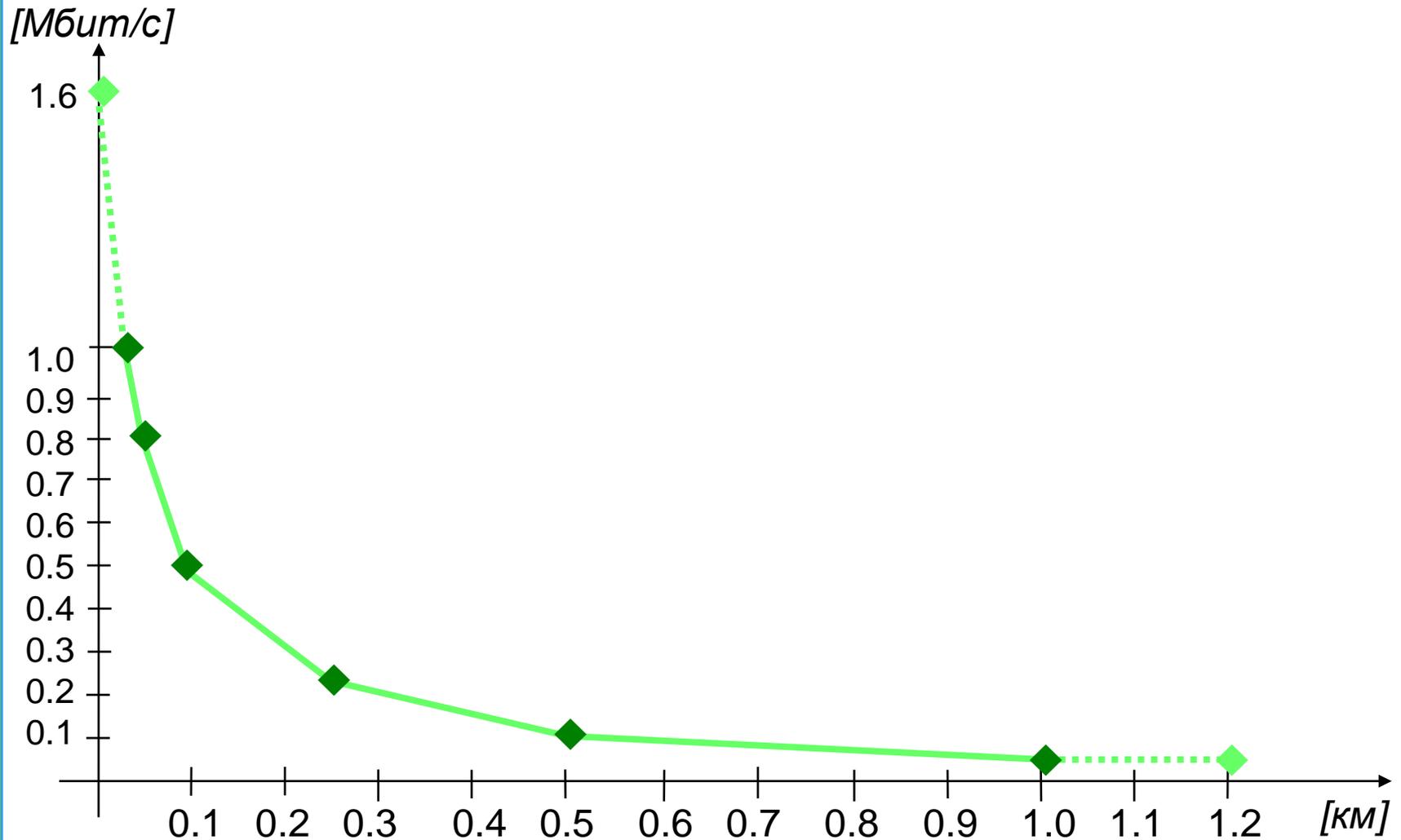


Все узлы автоматически ресинхронизируются

# Распространение сигнала



# Соотношение скорости и длины шины



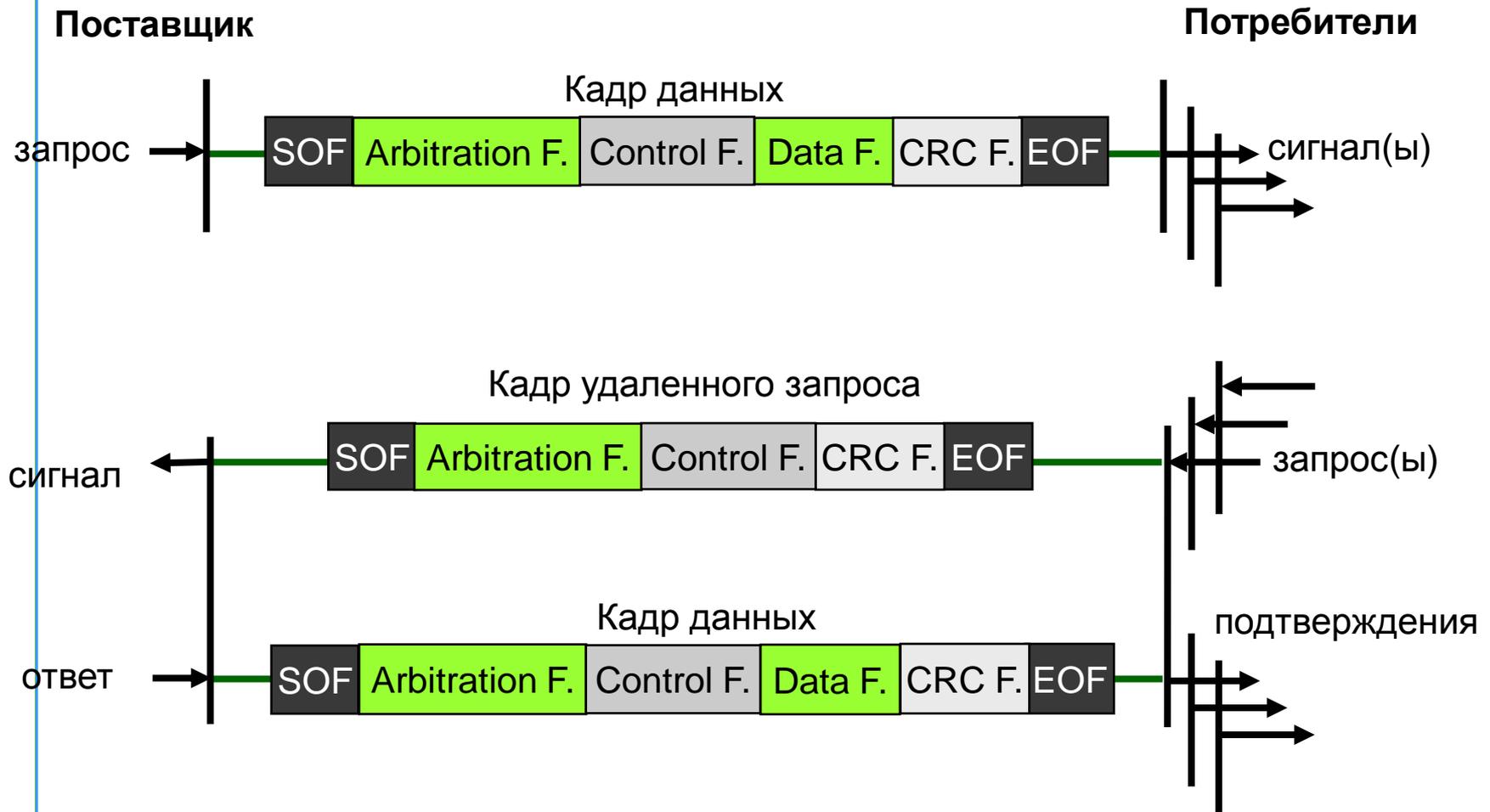
# Практическая длина шины

Скорость	Длина шины	Битовое время
1 Mbit/s	30 m	1 $\mu$ s
800 kbit/s	50 m	1.25 $\mu$ s
500 kbit/s	100 m	2 $\mu$ s
250 kbit/s	250 m	4 $\mu$ s
125 kbit/s	500 m	8 $\mu$ s
62.5 kbit/s	1000 m	20 $\mu$ s
20 kbit/s	2500 m	50 $\mu$ s
10 kbit/s	5000 m	100 $\mu$ s

# Набор скоростей, рекомендованный CiA DS-102

Битовая Скорость  Длина шины	Номи- нальное битовое время $t_b$	Число квантов времени в бите	Длитель- ность кванта времени $t_q$	Положение точки выборки	BTR 0 at 16 MHz (82C200)	BTR 1 at 16 MHz (82C200)
1 Mbit/s 25 m	1 $\mu$ s	8	125 ns	6 $t_q$ (750 ns)	00h	14h
800 kbit/s 50 m	1.25 $\mu$ s	10	125 ns	8 $t_q$ (1 $\mu$ s)	00h	16h
500 kbit/s 100 m	2 $\mu$ s	16	125 ns	14 $t_q$ (1.75 $\mu$ s)	00h	1Ch
250 kbit/s 250 m	4 $\mu$ s	16	250 ns	14 $t_q$ (3.5 $\mu$ s)	01h	1Ch
125 kbit/s 500 m	8 $\mu$ s	16	500 ns	14 $t_q$ (7 $\mu$ s)	03h	1Ch
50 kbit/s 1000 m	20 $\mu$ s	16	1.25 $\mu$ s	14 $t_q$ (17.5 $\mu$ s)	09h	1Ch
20 kbit/s 2500 m	50 $\mu$ s	16	3.125 $\mu$ s	14 $t_q$ (43.75 $\mu$ s)	18h	1Ch
10 kbit/s 5000 m	100 $\mu$ s	16	6.25 $\mu$ s	14 $t_q$ (87.5 $\mu$ s)	31h	1Ch

# Канальный уровень: кадры данных



В соответствии с ISO 11898-1 (канальный уровень CAN)

# Канальный уровень: определения и понятия

## Master-Slave

Ведущий-Ведомый

Poling (Опрос)

Данные выдаются тогда и только тогда когда их запросит Ведущий

Логически централизованные СУ

Альтернатива – событийно ориентированная модель

## Multimaster

Producer - Consumer

Поставщик - Потребители

Данные «товар» выдаются в сеть в независимости от наличия потребителя на них

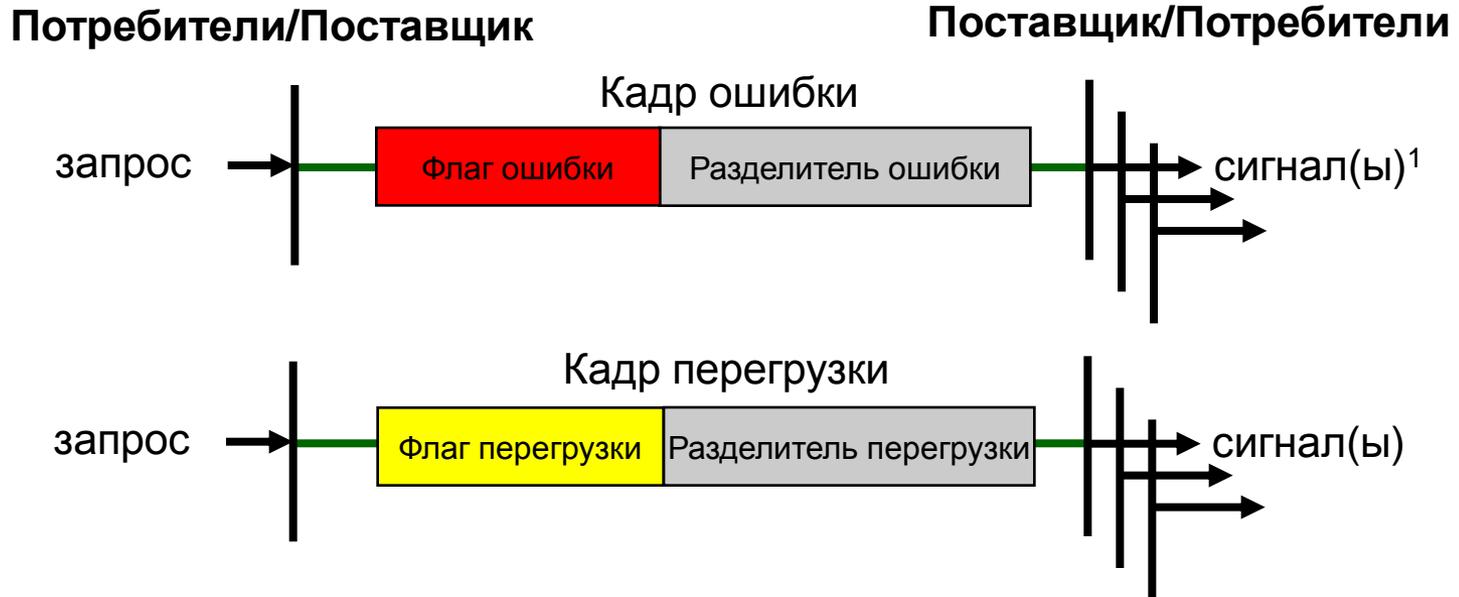
Децентрализованные, сетевые СУ

Клиент-Сервер

Service requester-  
Service provider

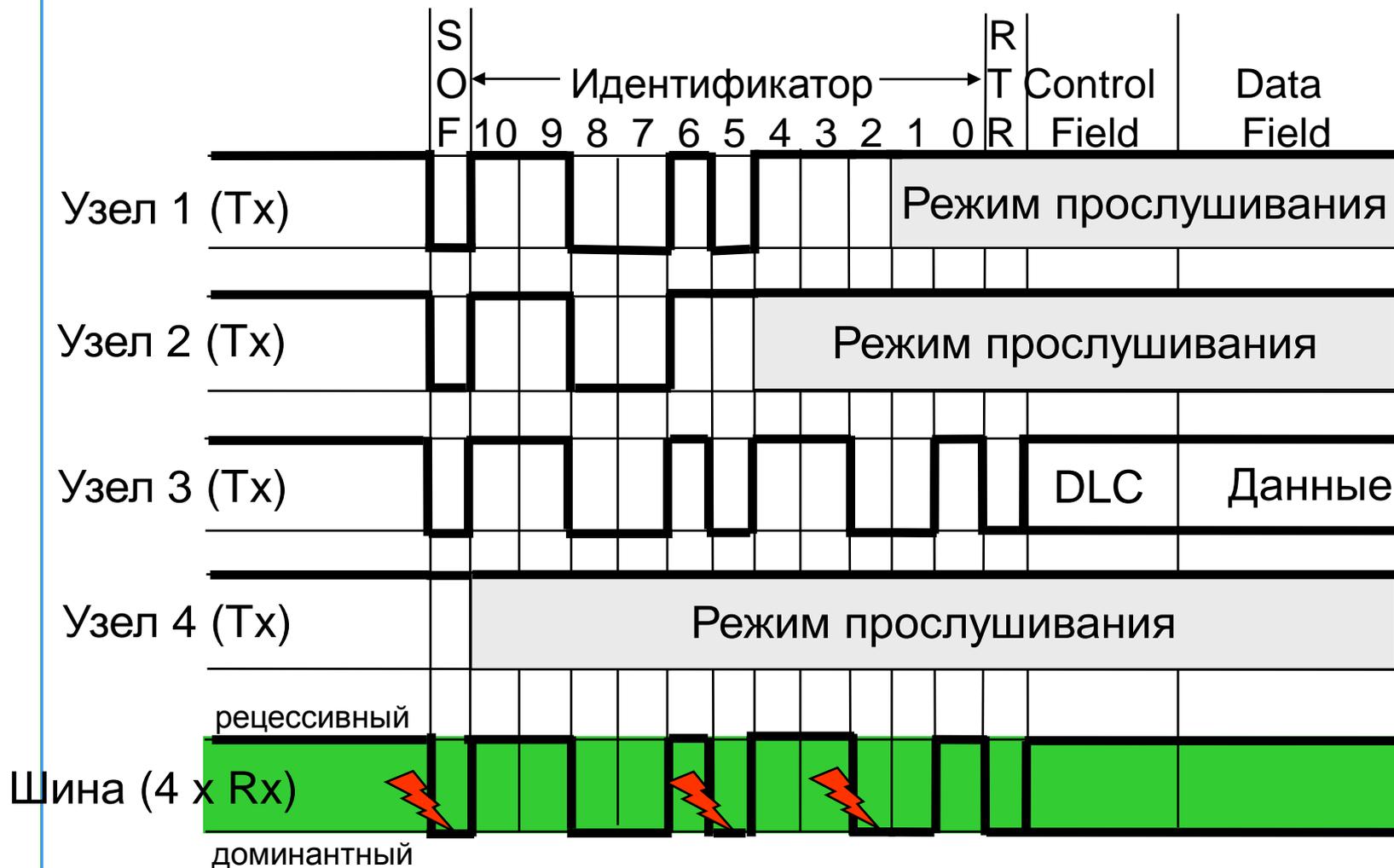
Услуги без запроса не предоставляются

# Канальный уровень: кадры ошибки и перегрузки

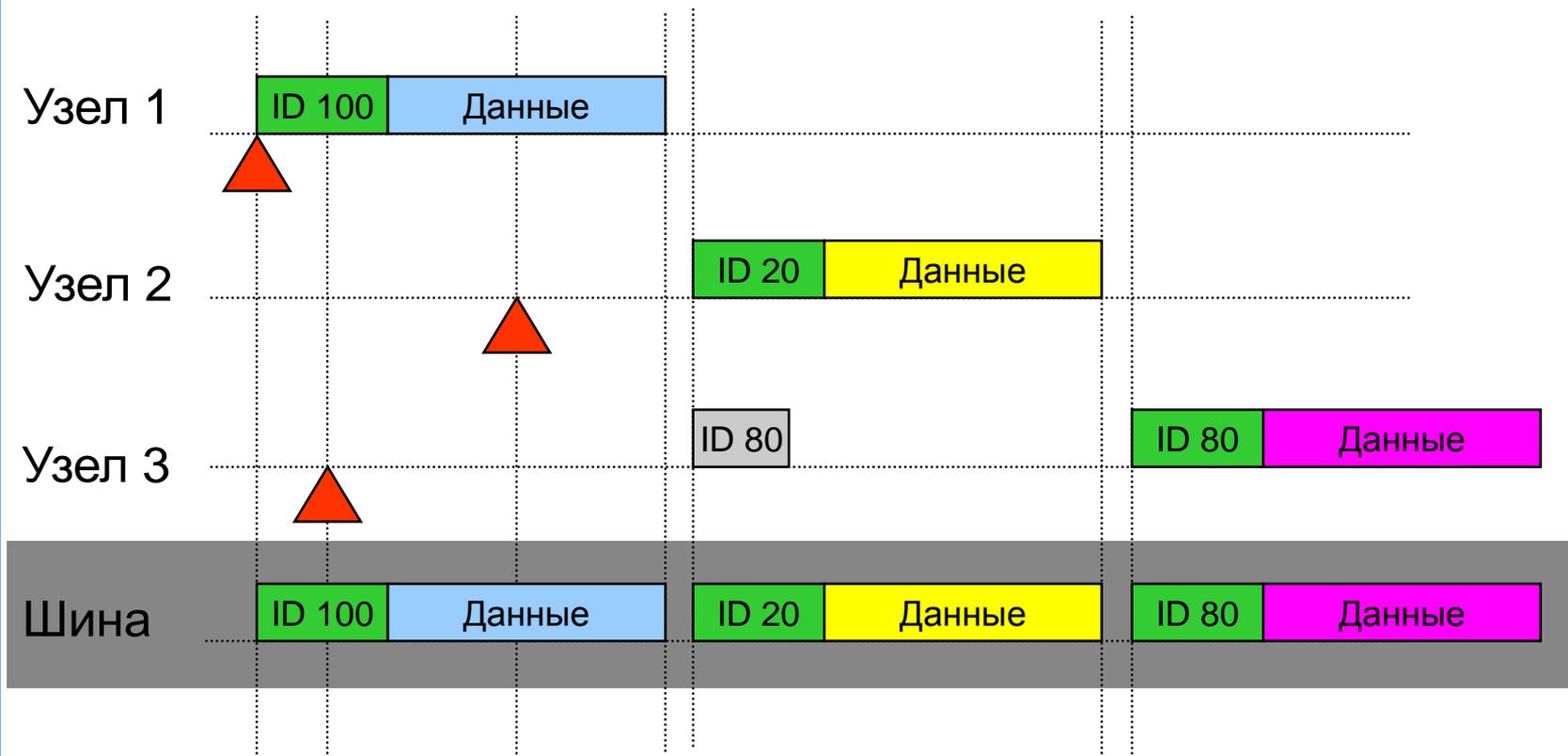


<sup>1</sup> Потребитель игнорирует последний кадр;  
поставщик автоматически осуществляет повторную передачу.

# Исключение столкновений



# Арбитраж шины



Запрос передачи кадра

# Кадр данных CAN



*Поле арбитража:*

12 бит для 11 битового идентификатора  
(основной формат кадра)

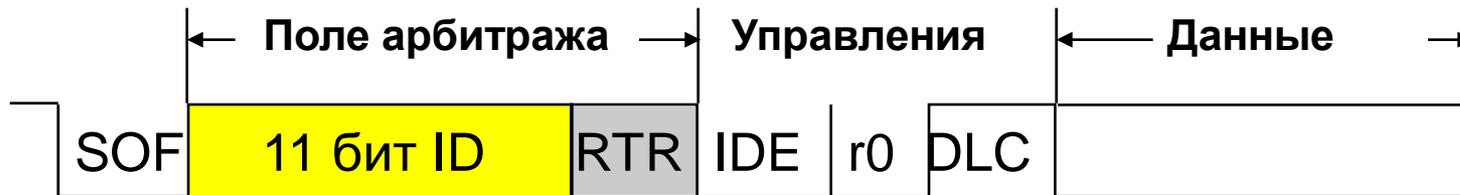
32 бита для 29 битового идентификатора  
(расширенный формат кадра)

Минимальная длина кадра: **47** бит

Максимальная длина кадра: **131** бит

# Поле арбитража

## Основной формат кадра



## Расширенный формат кадра



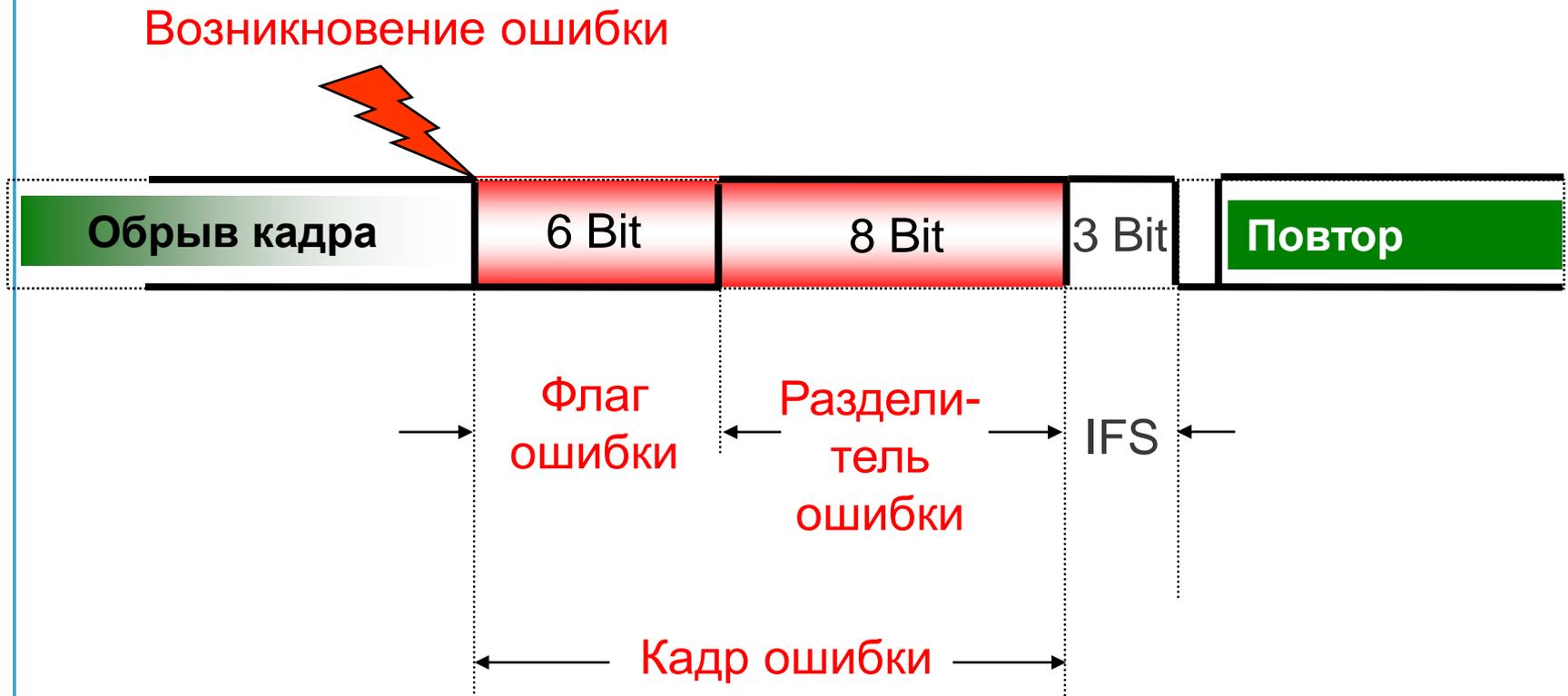
**Особенности:** более длинные кадры (на 20 бит плюс стаф-биты) увеличивают латентность шины и ухудшают качество CRC контроля

# Кадр удаленного запроса CAN

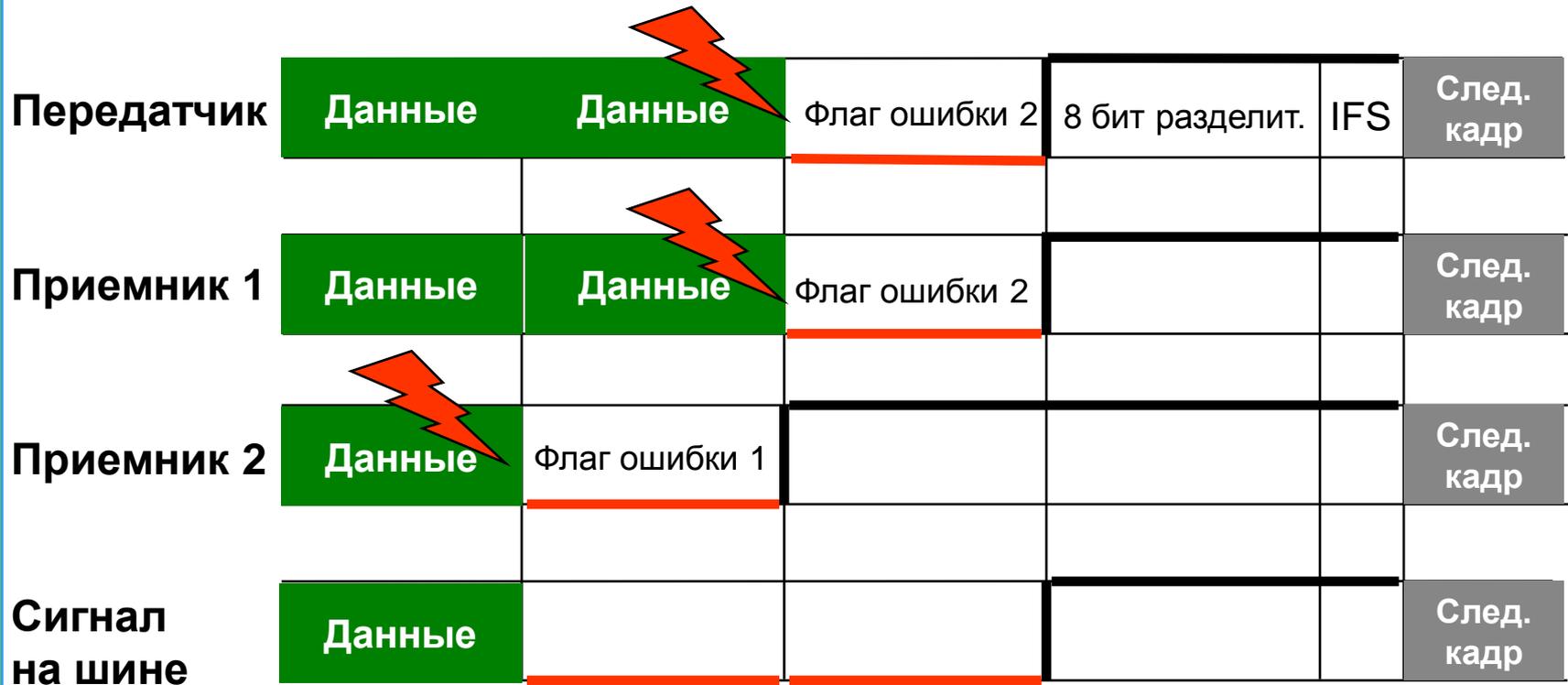


- ◆ CAN контроллер, имеющий приемный буфер или FIFO, отвечает на удаленный запрос только под управлением центрального процессора.
- ◆ CAN контроллер, поддерживающий стандартный алгоритм хранения сообщений, отвечает на удаленный запрос автоматически без вмешательства центрального процессора.
- ◆ CAN контроллер, поддерживающий продвинутый алгоритм хранения сообщений, отвечает на удаленный запрос автоматически *и опционально* – под управлением центрального процессора.

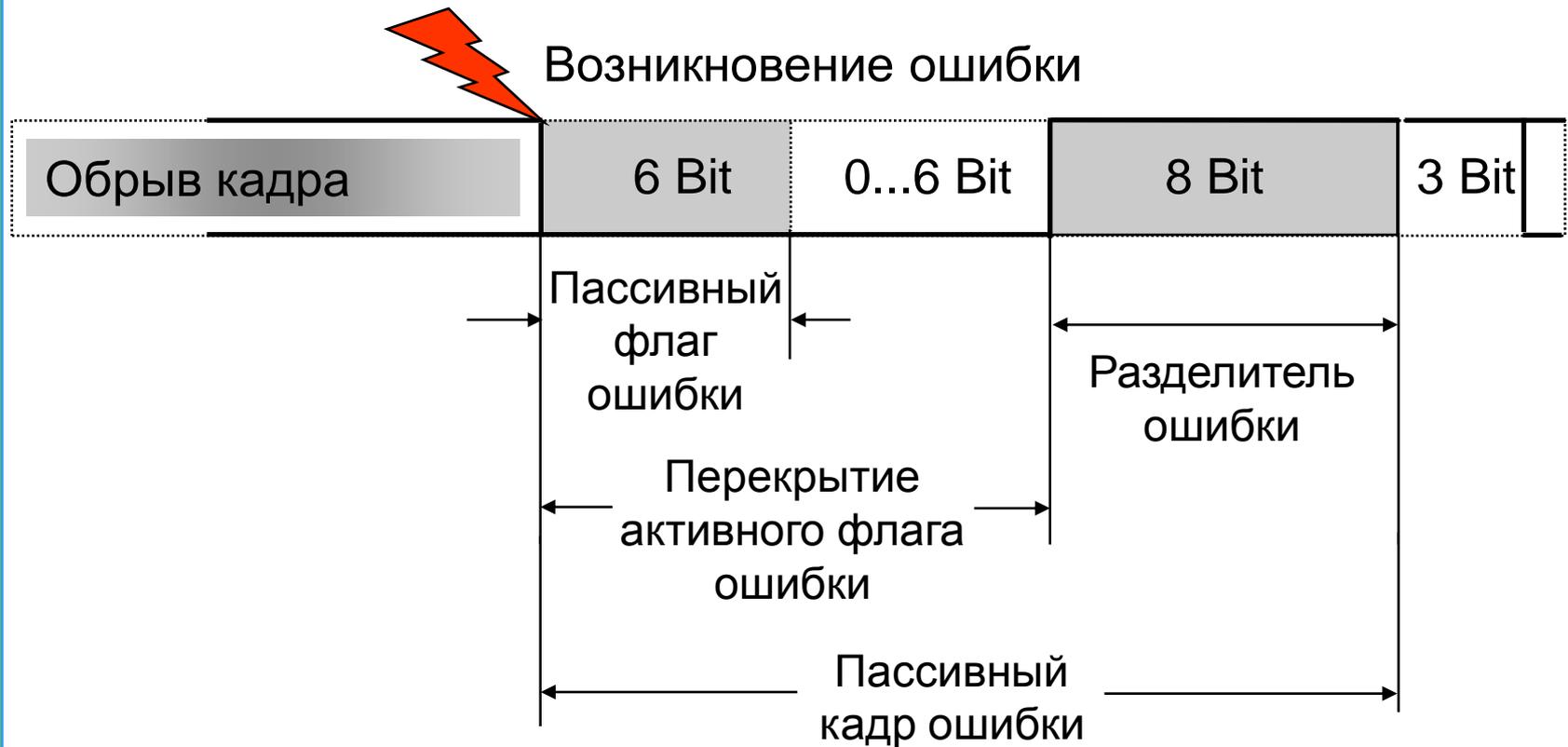
# Активный кадр ошибки



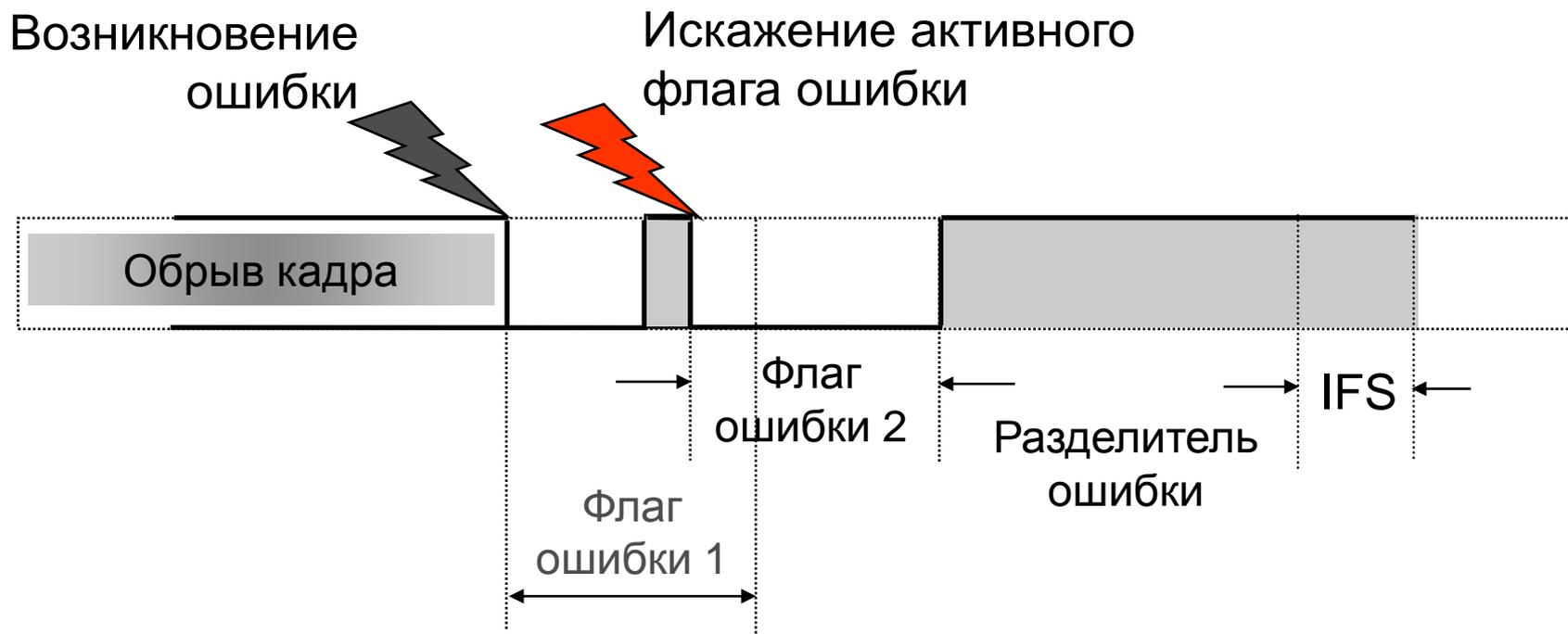
# Глобализация локальной ошибки



# Пассивный кадр ошибки



# Искажение кадра ошибки



**Замечание:**

При искажении бита в разделителе ошибки передается кадр перегрузки.

# Обнаружение ошибок

Вероятность **не** обнаружения ошибки в стандартном CAN кадре:

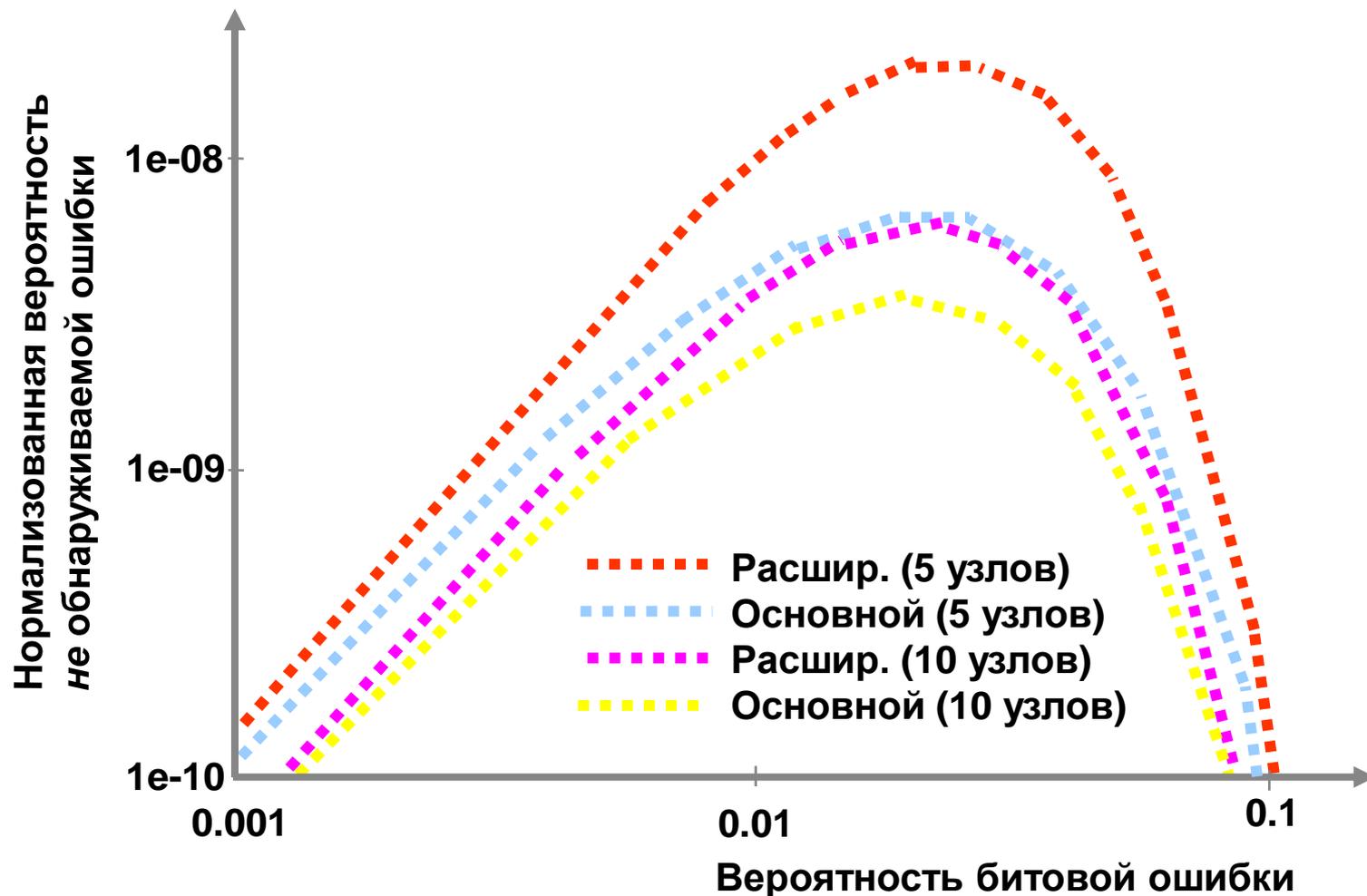
$$\rho < 4.7 \times 10^{-11} \times \text{коэфф. ошибок}$$

*Пример:* 1 битовая ошибка за каждые 0.7  
с,

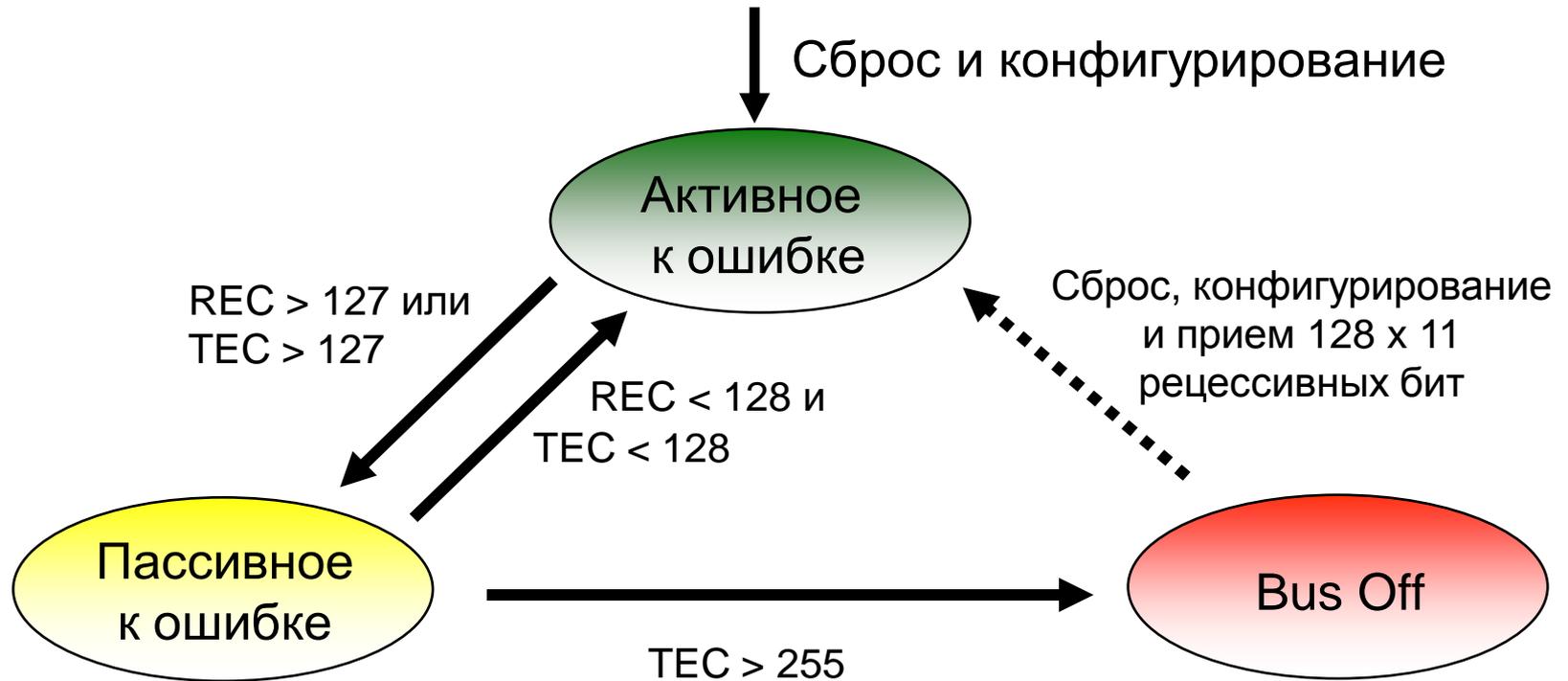
500 кбит/с, 8ч в день, 365 дней в году.

Статистическая вероятность:  
1 **не** обнаруженная ошибка за 1000 лет

# Характеристики обнаружения ошибок



# Состояния ошибки CAN узла

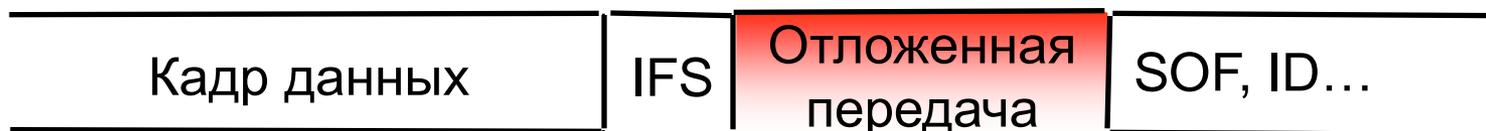


REC: Счетчик ошибок приема

TEC: Счетчик ошибок передачи

# Отложенная передача

Иницируется пассивным к ошибке узлом



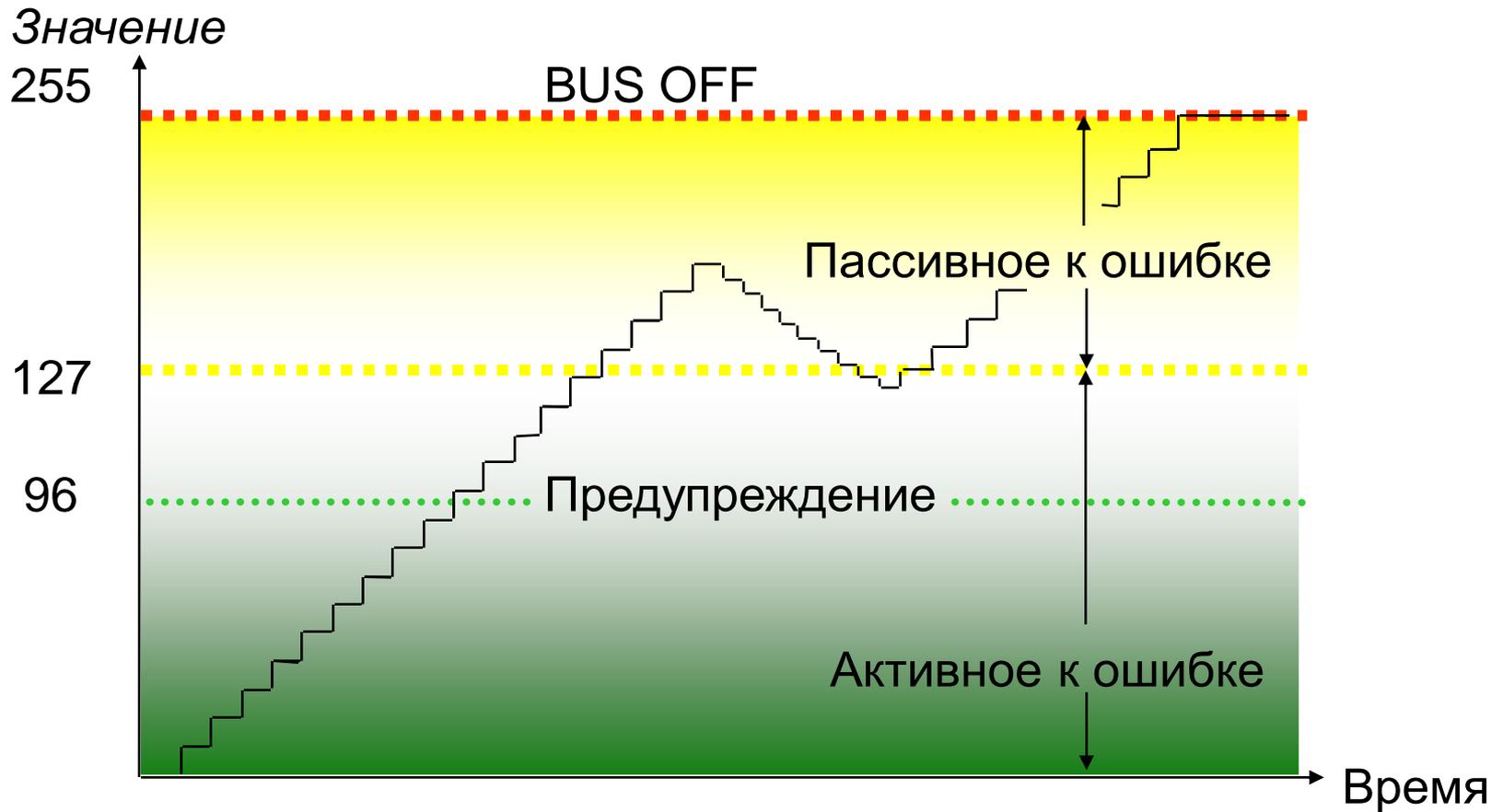
Иницируется другими узлами



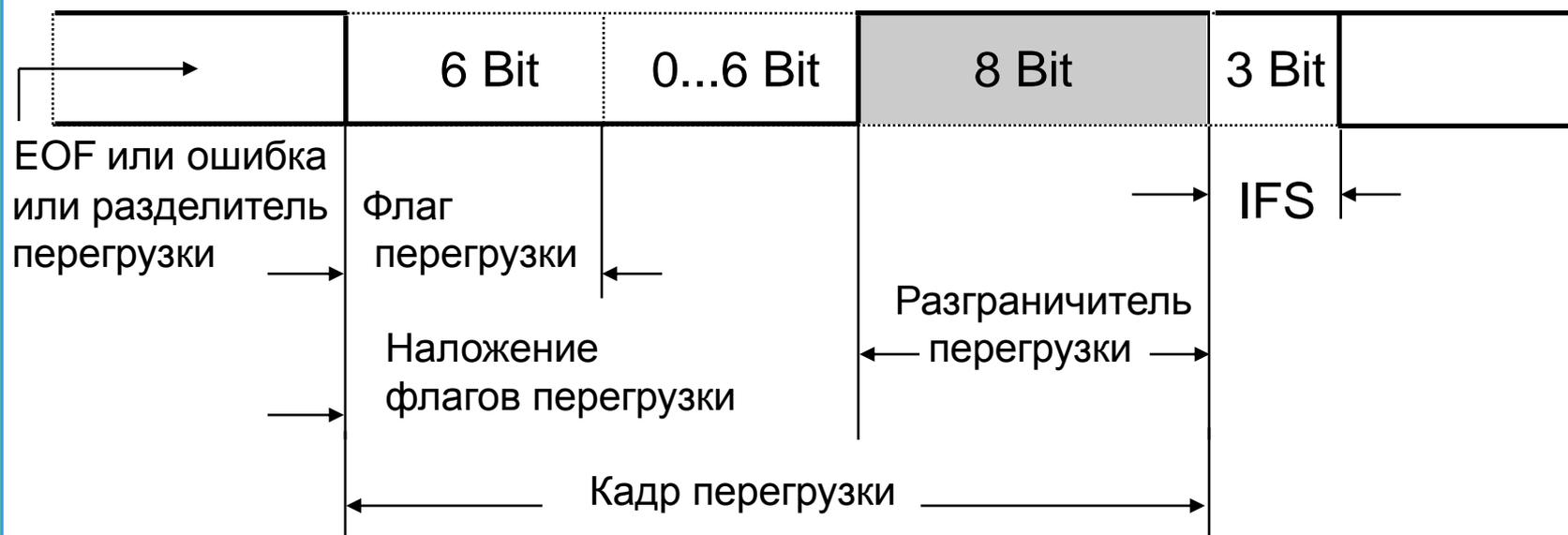
Межкадровый промежуток (IFS) = 3 бита

Отложенная передача = 8 бит

# Счетчик ошибок передачи



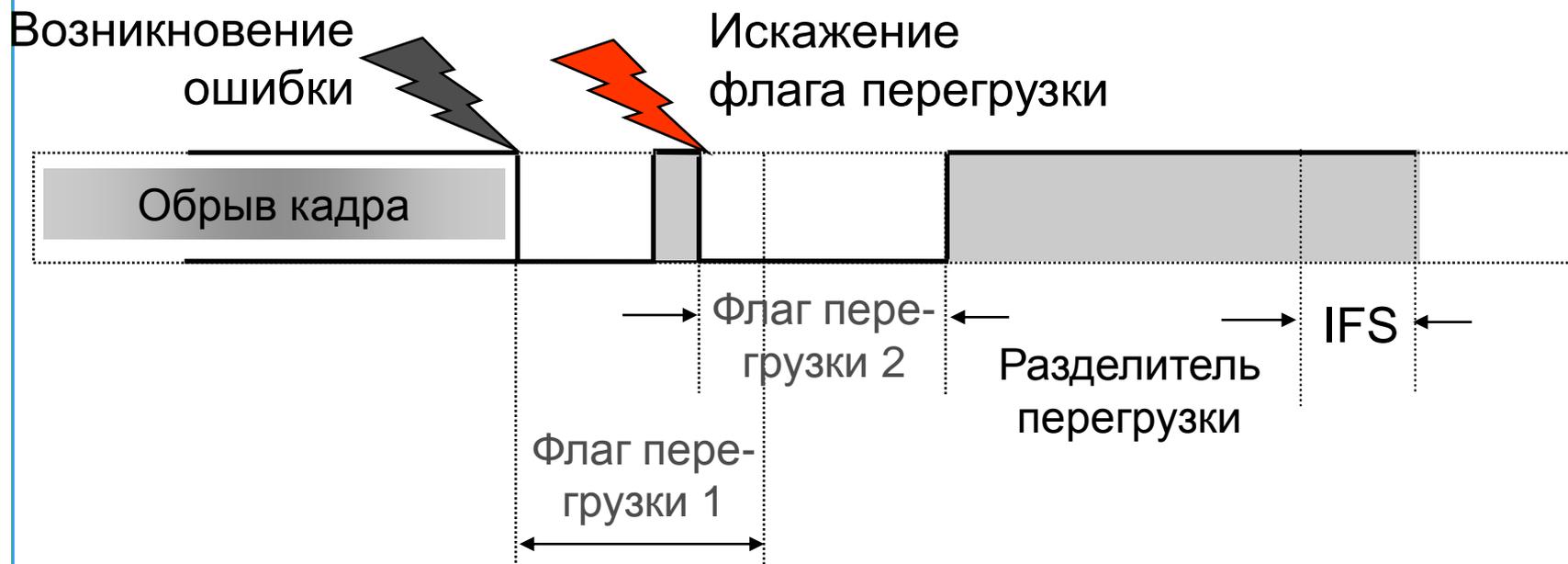
# Кадр перегрузки



## **Замечание:**

*Кадры перегрузки не вызывают повторной передачи кадров.*

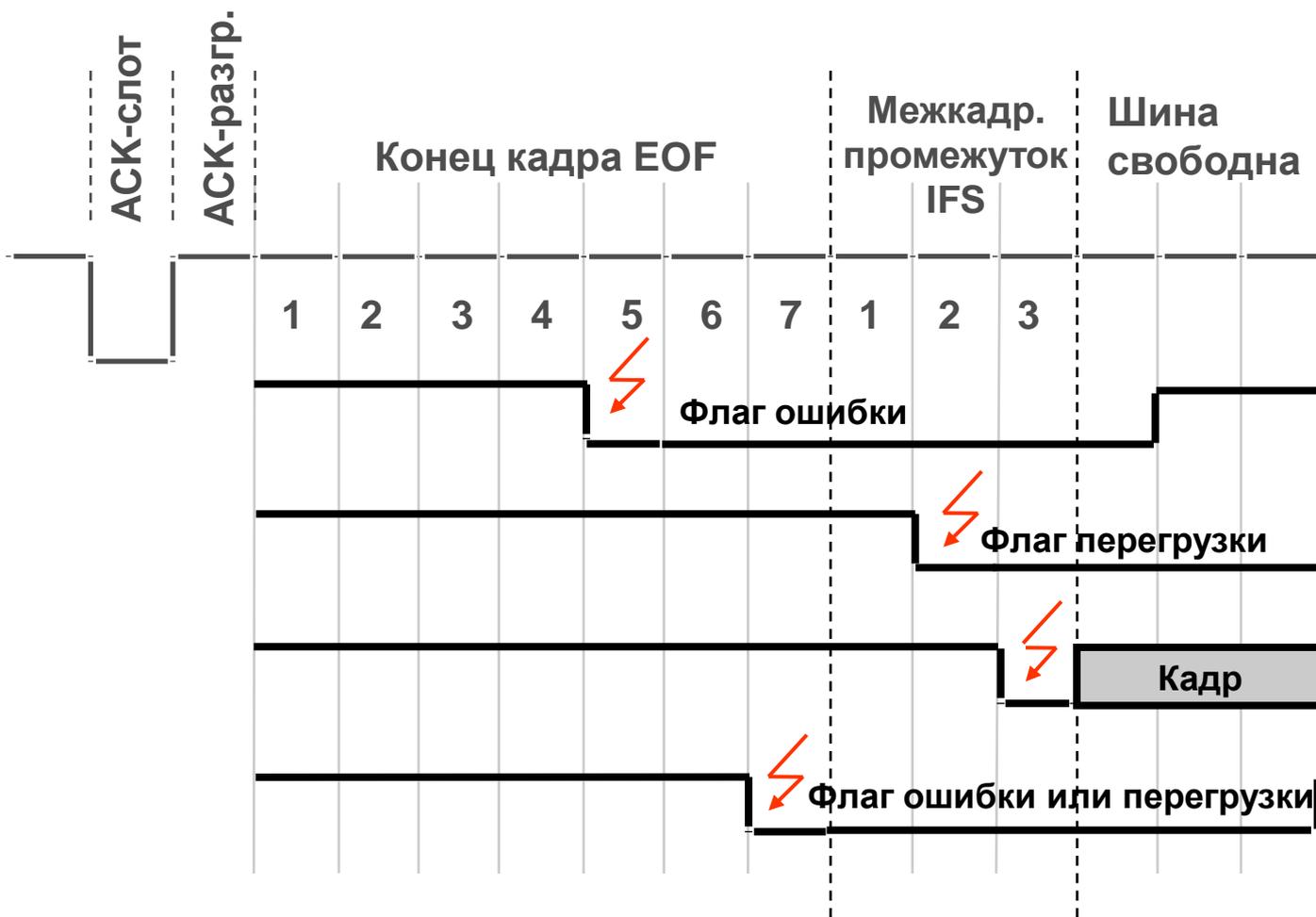
# Искажение кадра перегрузки



## Замечание:

При искажении бита в разделителе перегрузки активные к ошибке узлы передают кадр перегрузки. Каждый узел может передать не более двух кадров перегрузки подряд.

# Локальная ошибка в EOF и IFS

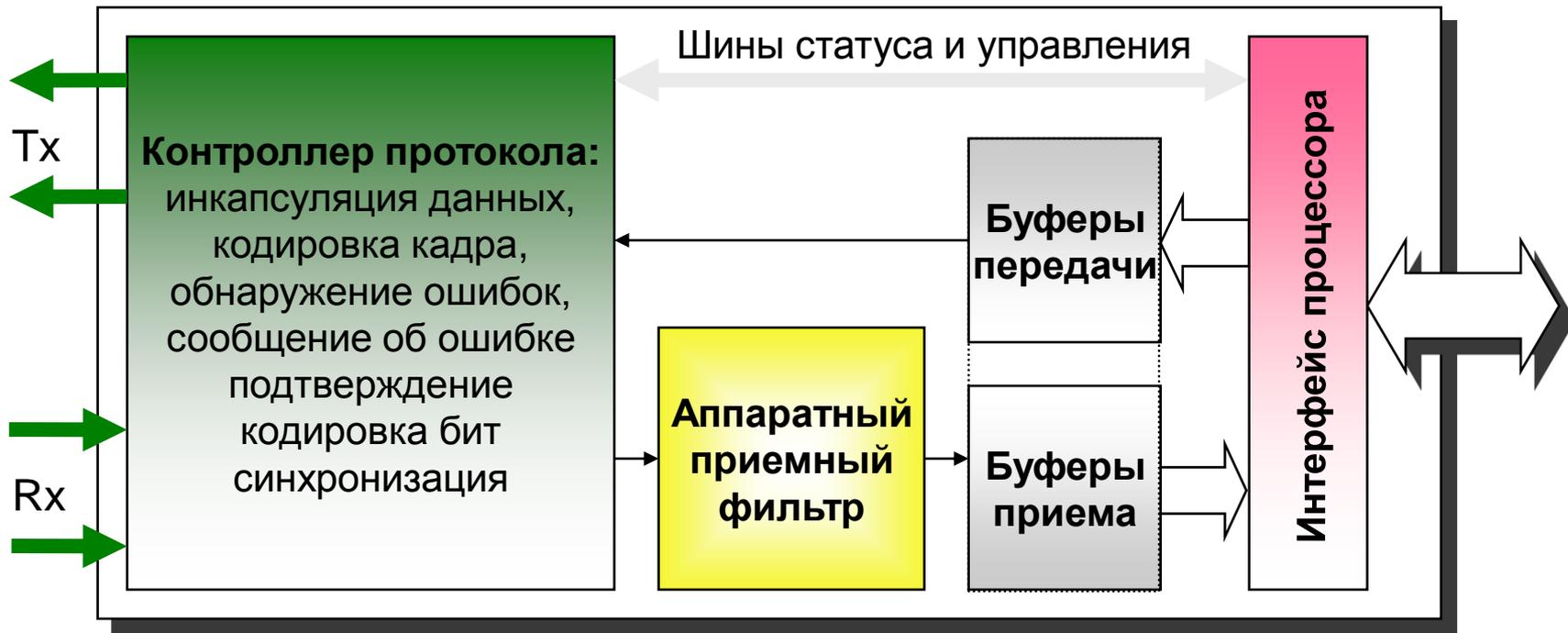


# Дублированный прием сообщений

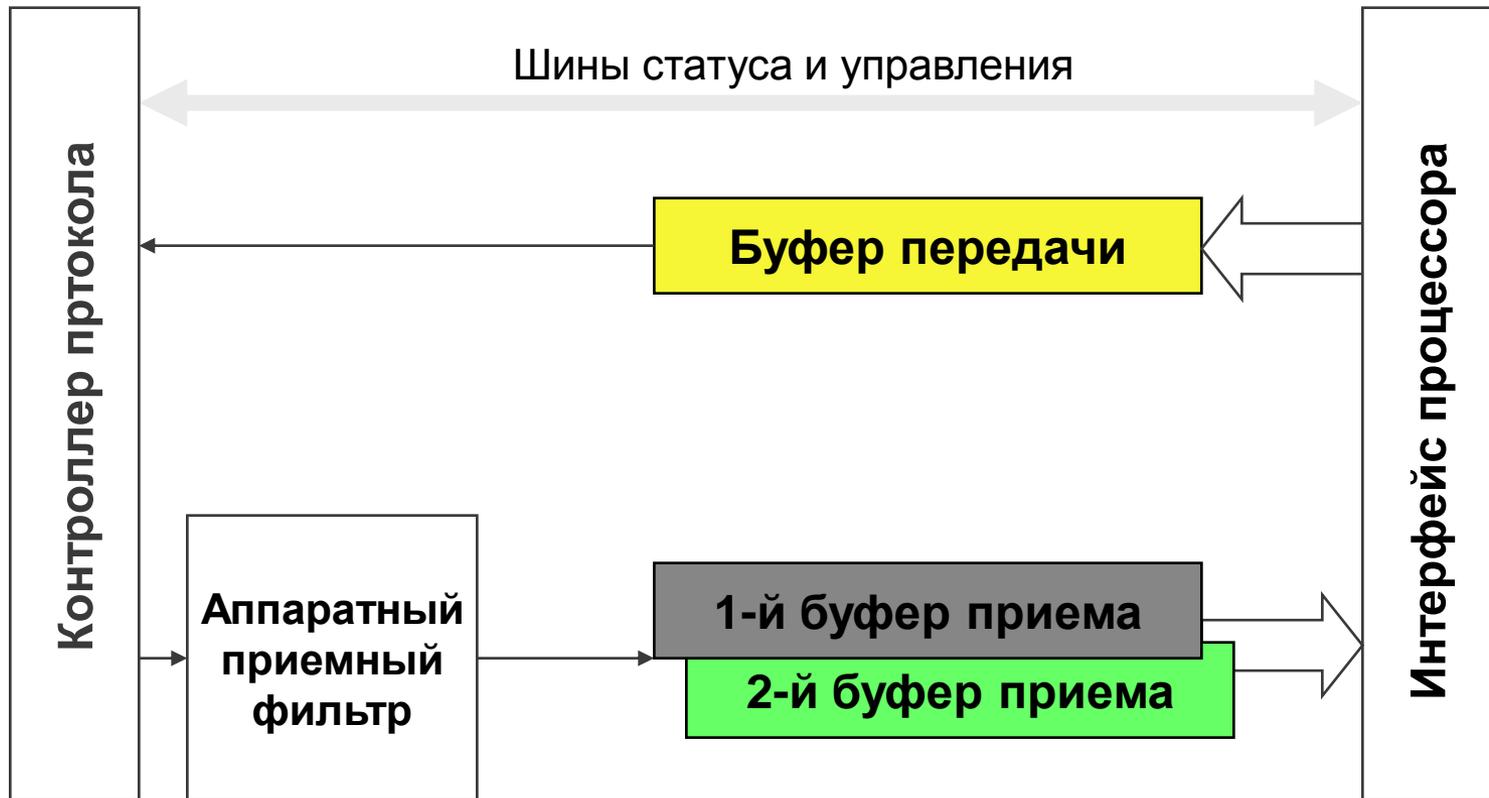
CAN протокол обеспечивает очень высокую вероятность целостности и сохранности сообщений. Но возможно дублирование сообщений вследствие ошибки одного бита вблизи конца кадра (EOF). Поэтому, если CAN используется в среде с высоким уровнем помех:

- ◆ не используйте мерцающих сообщений;
- ◆ не передавайте сообщений, содержащих относительные величины (изменение температуры и т.п.);
- ◆ используйте защищенные протоколы или нумерацию кадров для передачи сегментированных данных.

# Общая архитектура CAN контроллеров



# Классическая буферизация сообщений



# Внутренняя инверсия приоритета

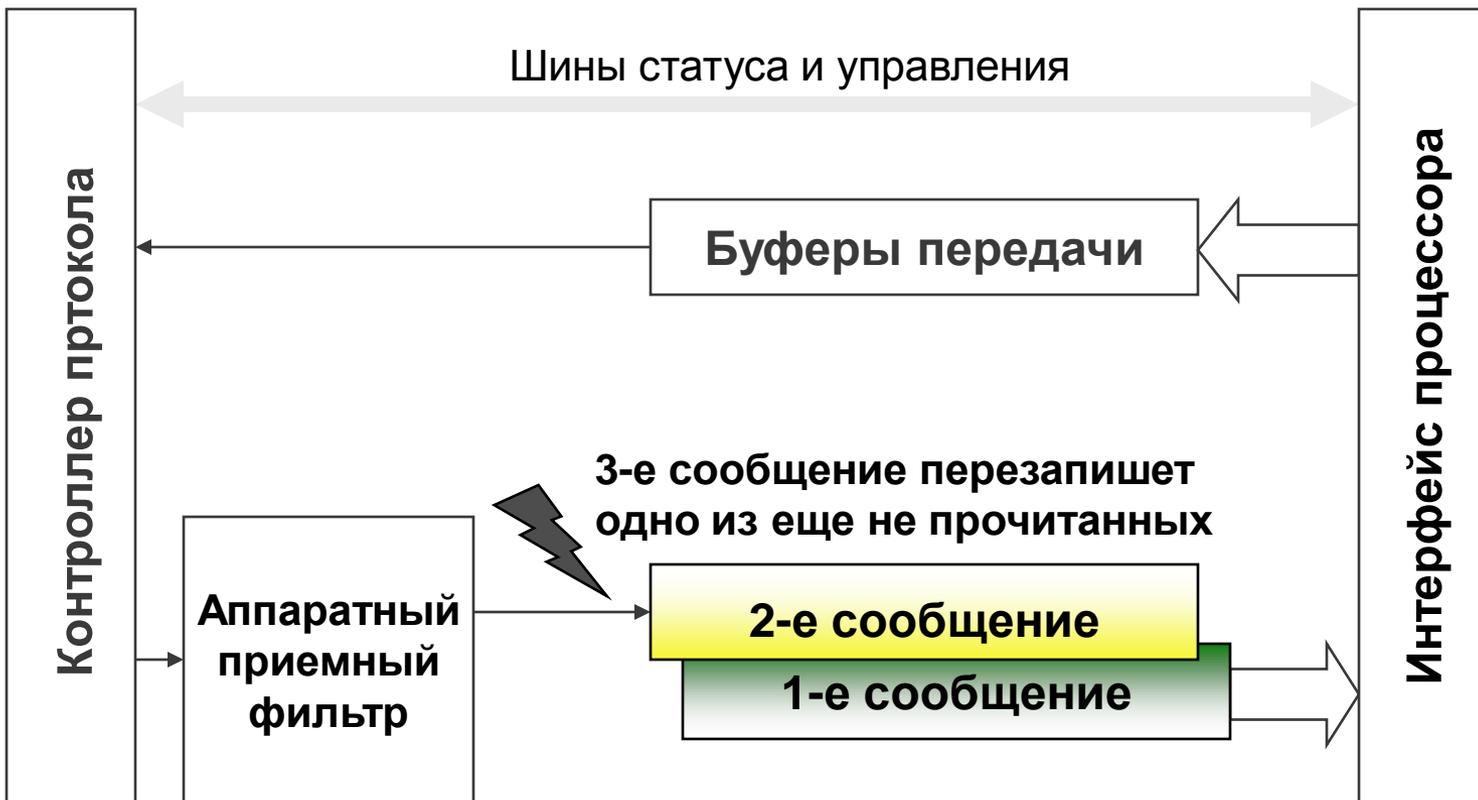


Высоко приоритетное  
сообщение

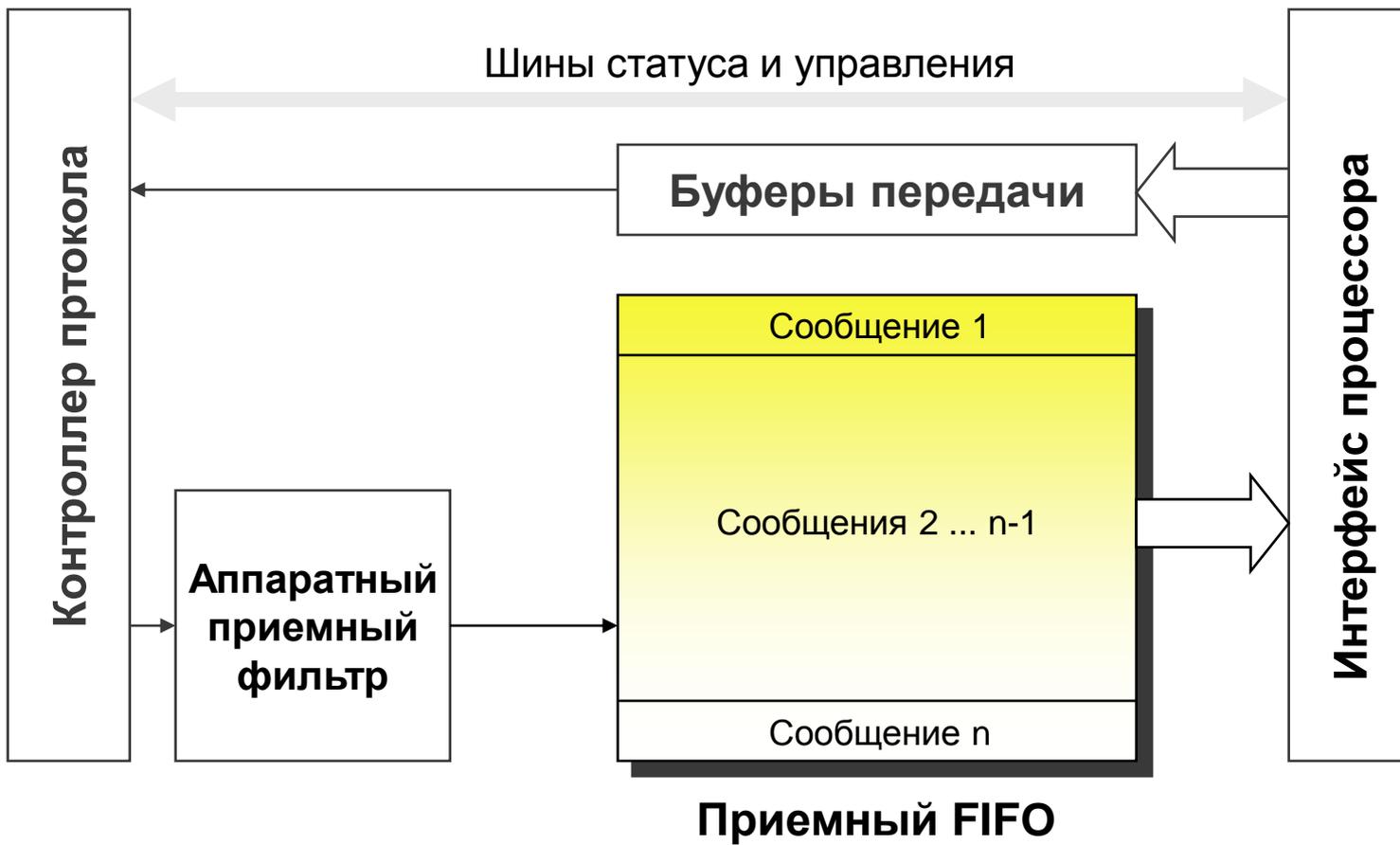
# Три буфера передачи



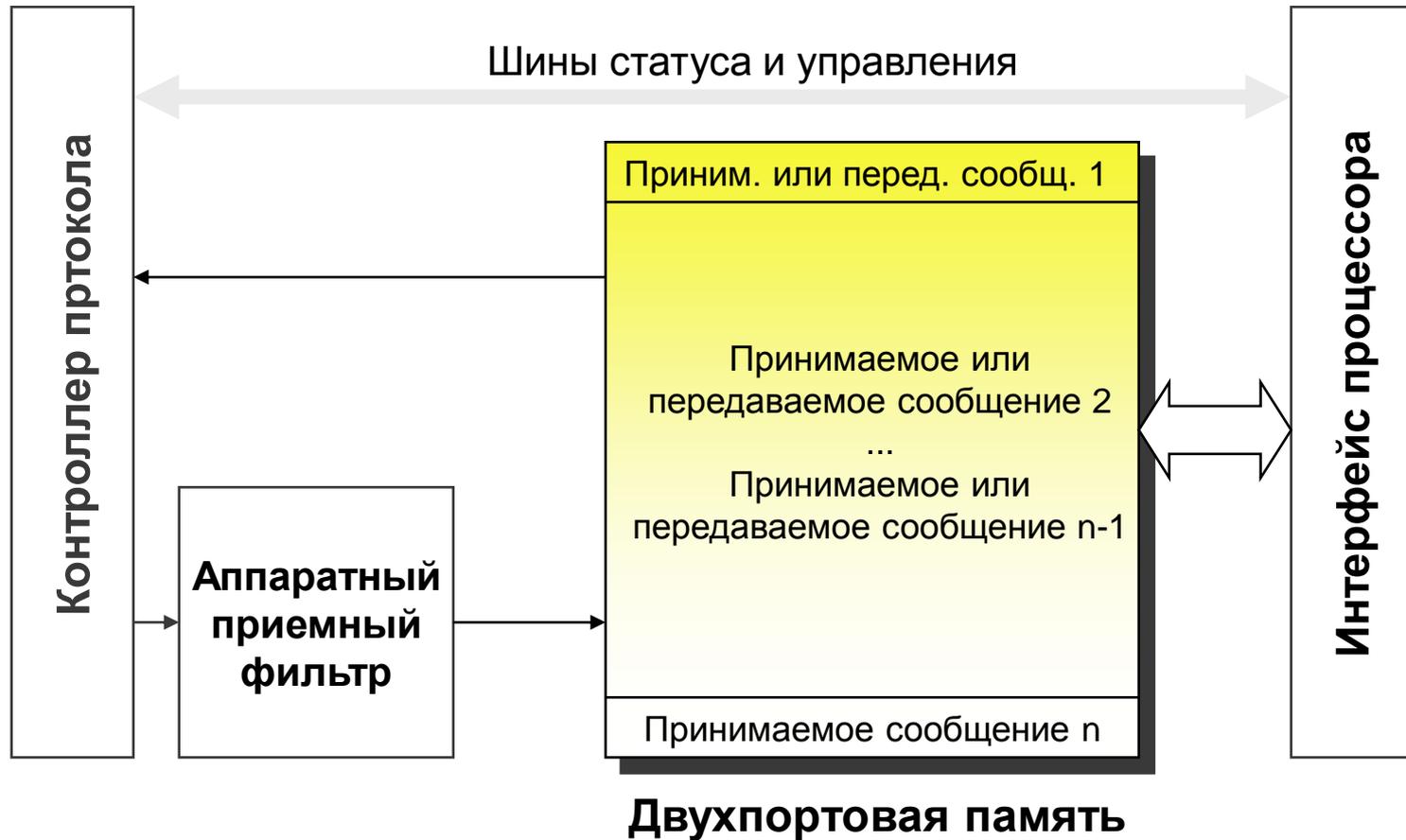
# Переполнение сообщений



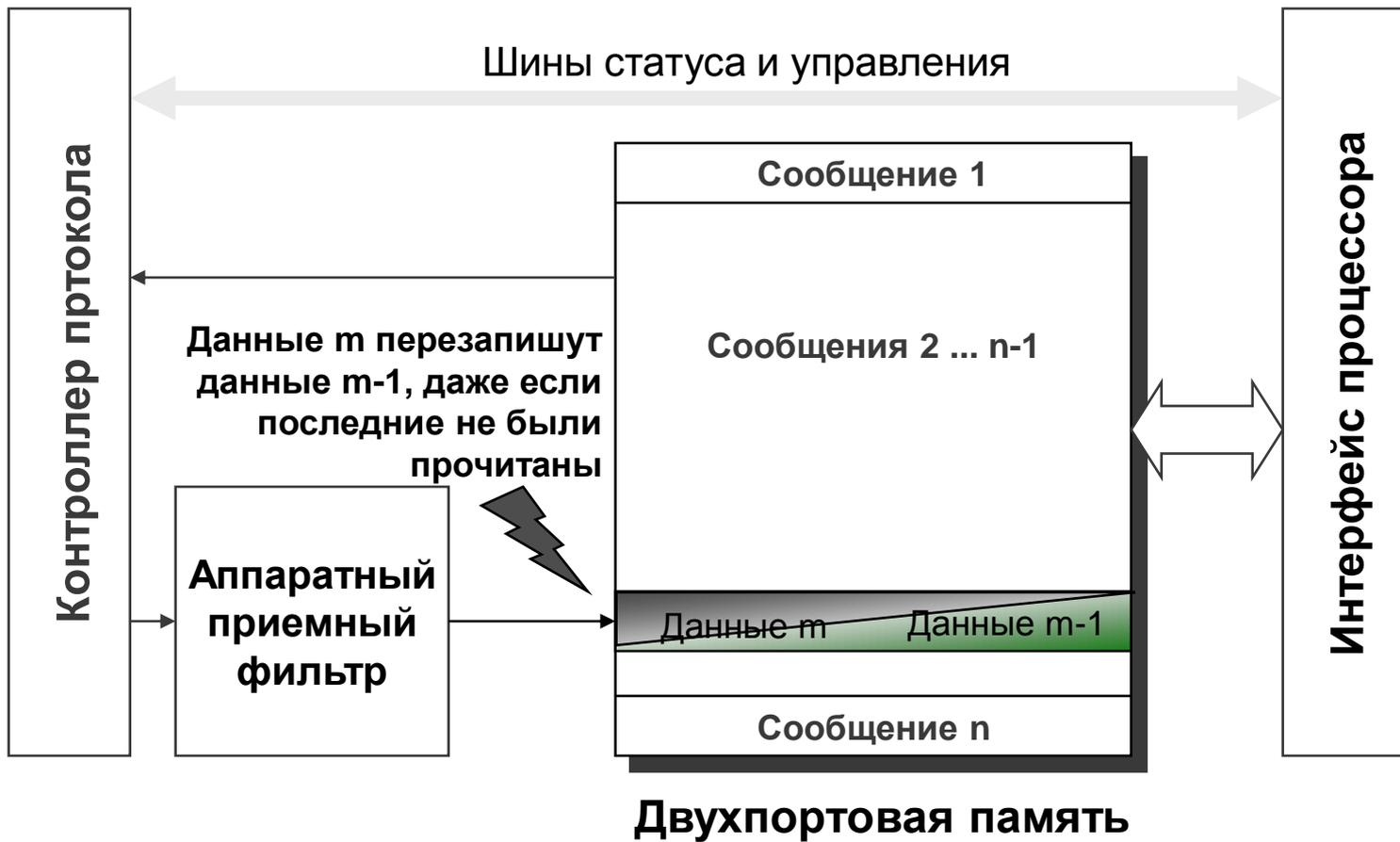
# Приемные FIFO



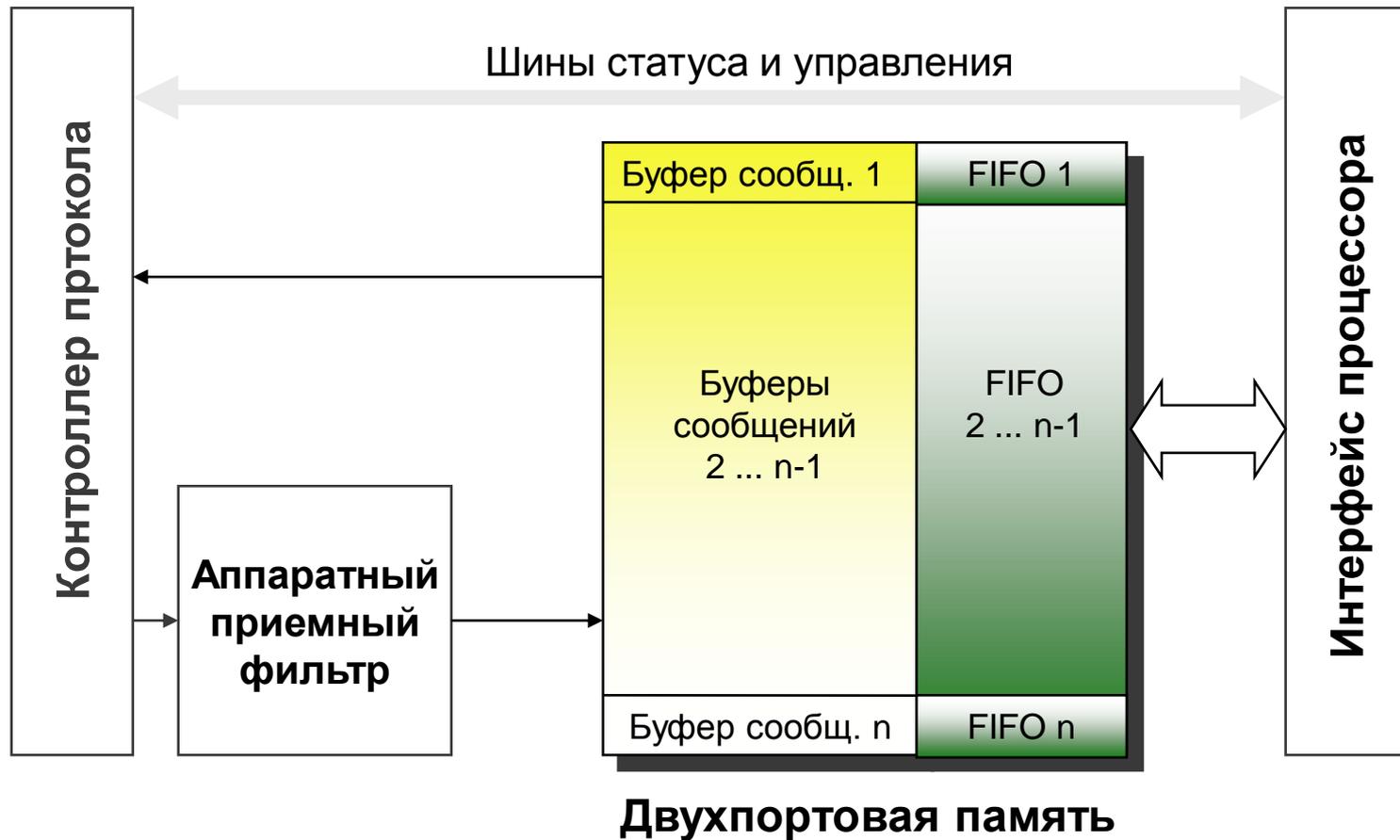
# Классическое хранение сообщений



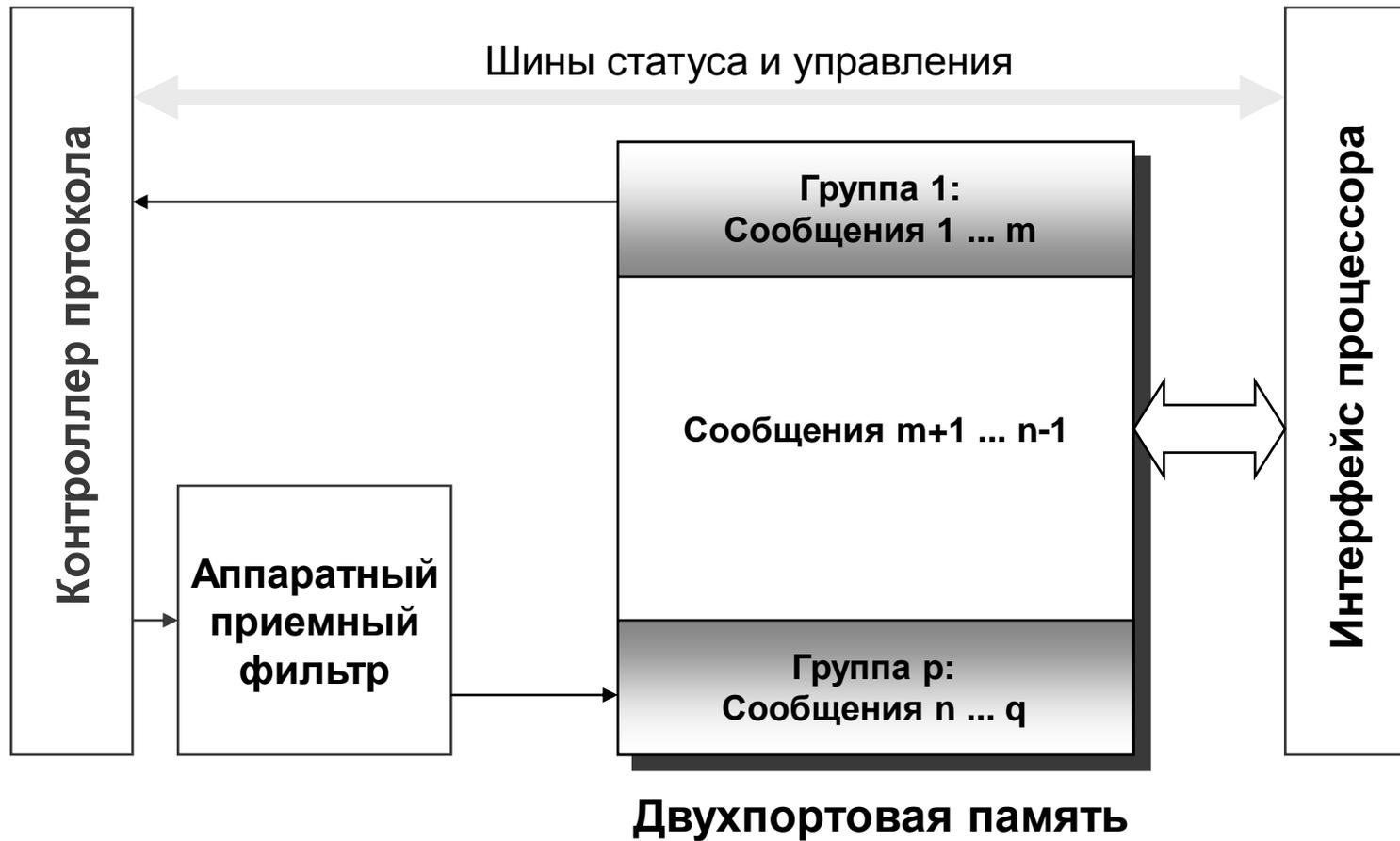
# Перезапись данных



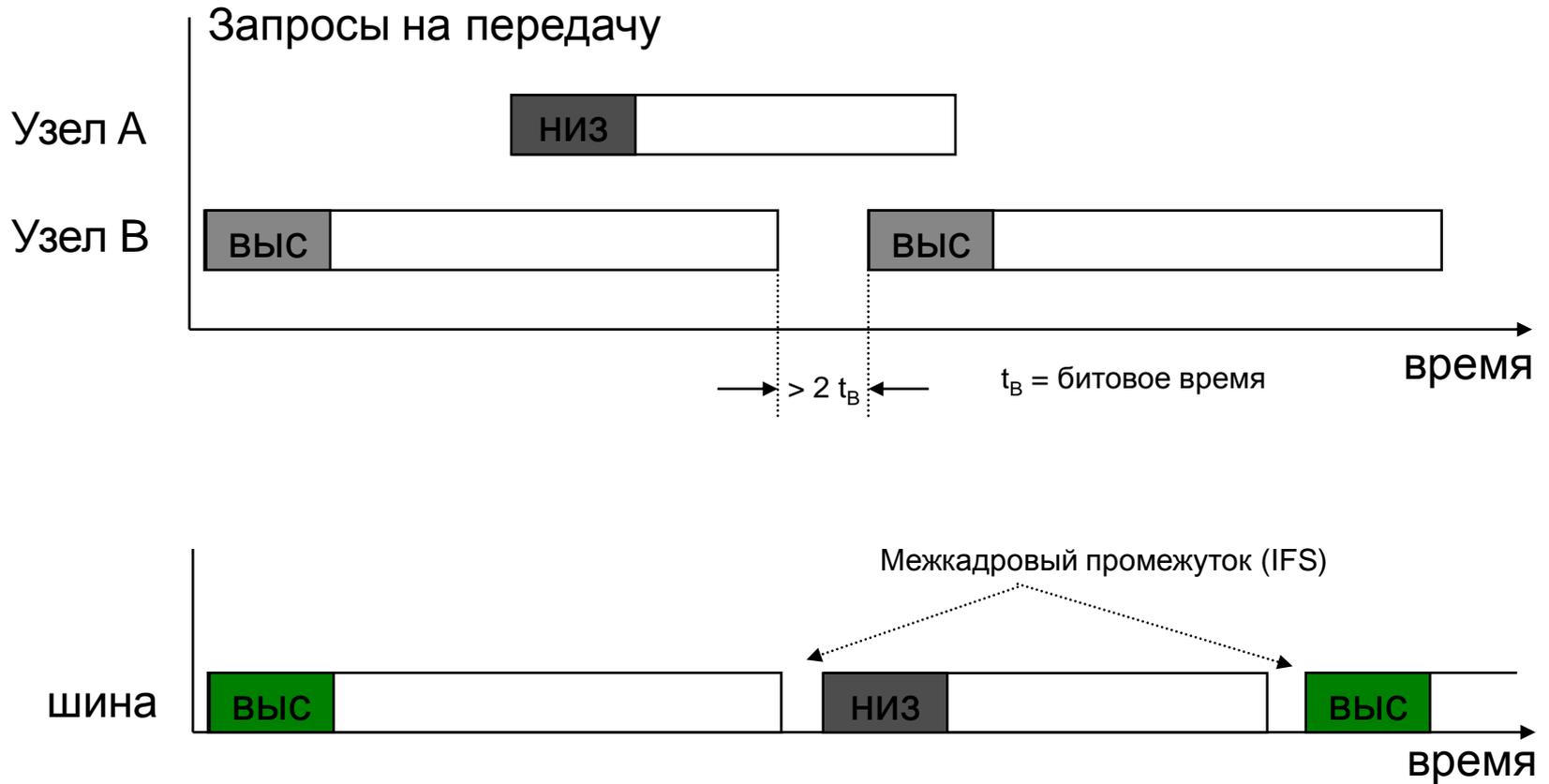
# Хранение сообщения в FIFO



# Группирование буферов сообщений



# Внешняя инверсия приоритета



# Прикладной уровень CANopen

Модули В/В



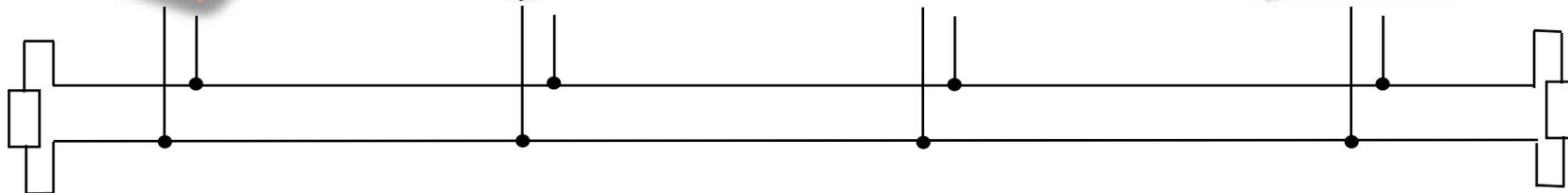
Сервоприводы



ПИД регуляторы

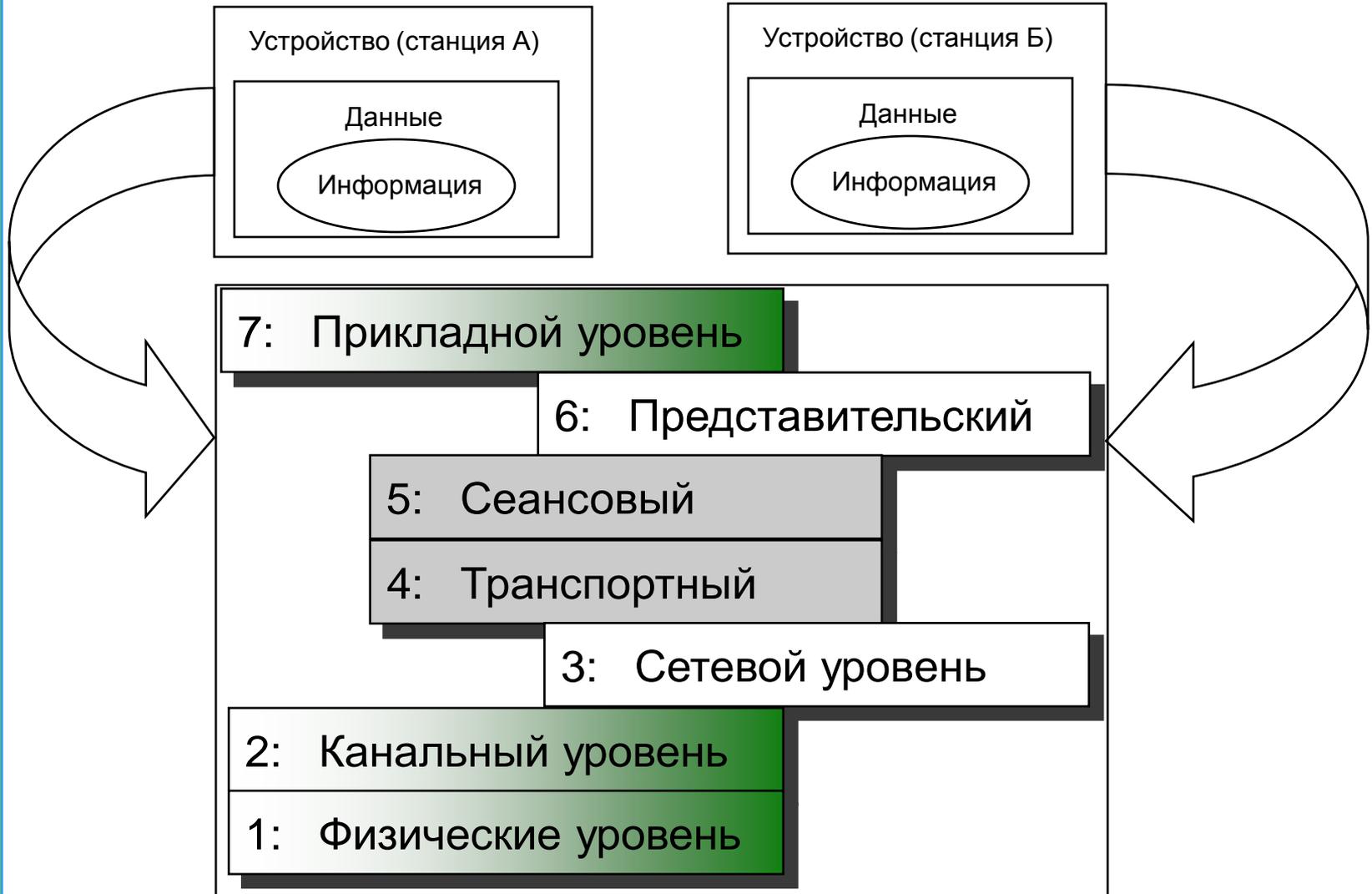


Консоль оператора

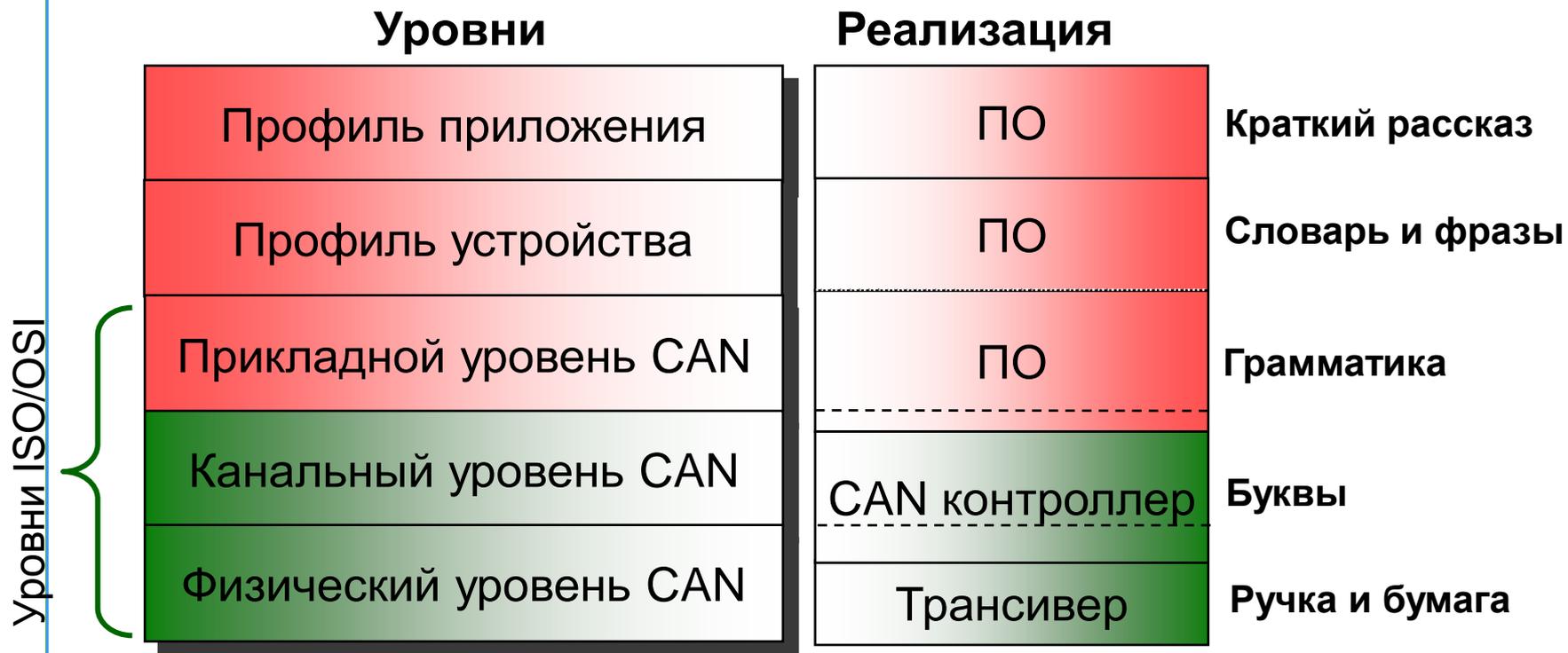


Пример CAN-bus/CANopen сети

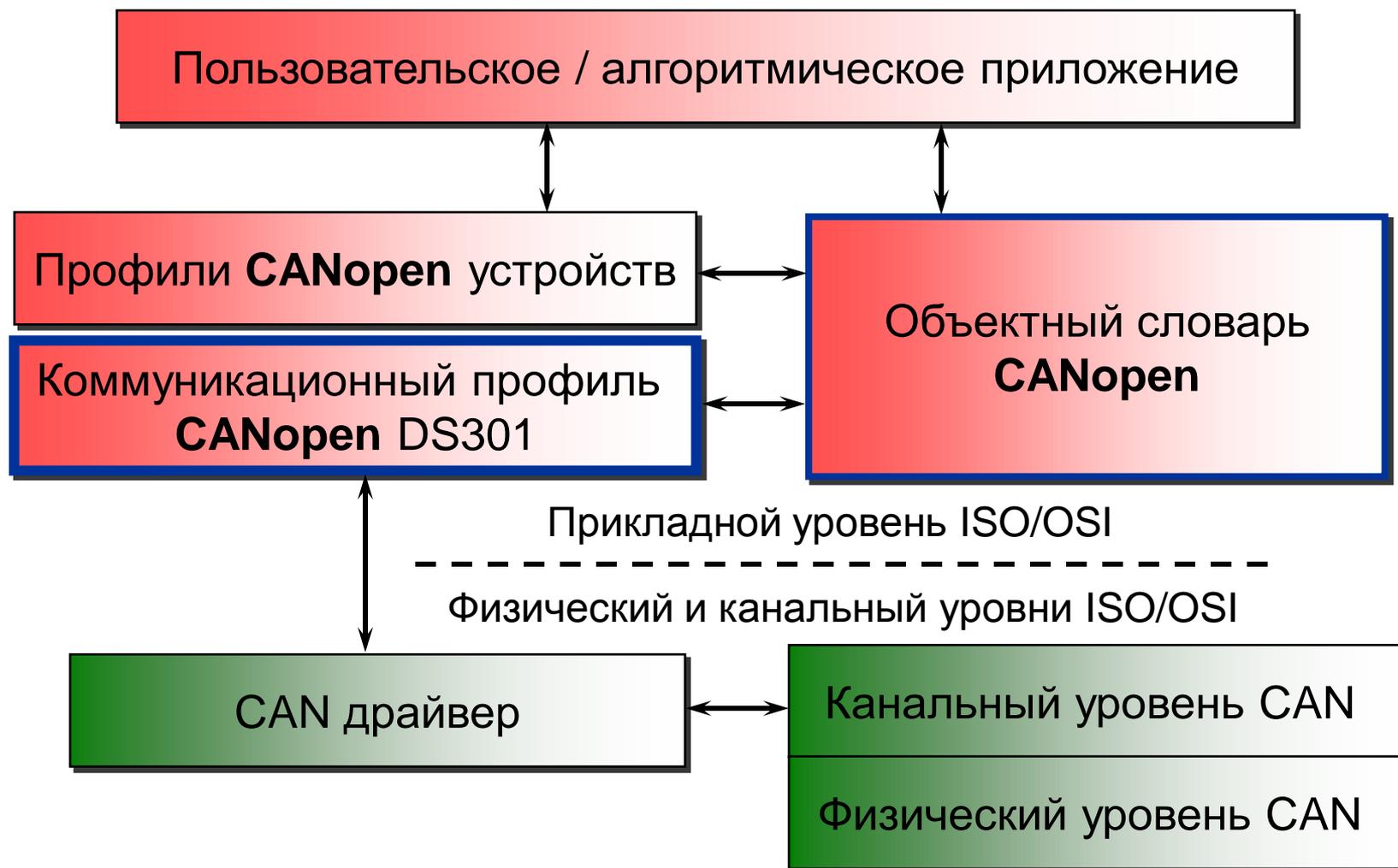
# Эталонная модель ISO/OSI



# Эталонная модель CAN



# Эталонная модель CANopen



# *Коммуникационные протоколы*

- Протокол PDO (объект данных процесса)
- Протокол SDO (сервисный объект данных)
- Протоколы специальных объектов:
  - Протокол синхронизации SYNC
  - Протокол временной метки TIME
  - Протокол срочного сообщения EMCY
- Протоколы управления сетью:
  - NMT протокол
  - Boot-Up протокол (загрузка узла)
  - Протокол сердцебиения



# Предопределенное распределение идентификаторов

Объект	Код функции (двоичный)	Итоговый COB-ID	Индекс объектного словаря
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (639h)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h
LSS slave	1111	2020	-
LSS master	1111	2021	-

# Объектный словарь CANopen

ОС - наиболее важный компонент любого CANopenустройства, поскольку в нем описано функции устройства и способ его поведения в сети !

- В ОС в определенном порядке сгруппированы объекты
- Некоторые из объектов должны присутствовать в обязательном порядке
- Каждый узел сети CANopen должен иметь (имеет) свой ОС
- EDS файл описывает стандартным образом ОС CANopen устройства

# Структура объектного словаря

Индекс (hex)	Объект
0000	Зарезервирован
0001-009F	Определение типов данных
00A0-0FFF	Зарезервированы для будущего применения
1000-1FFF	Коммуникационные объекты (профиль)
2000-5FFF	Профили особых устройств
6000-9FFF	Профили стандартизированных устройств
A000-AFFF	Параметры сети (описывает поведение устройства в сети )
B000-FFFF	Зарезервированы для будущего применения

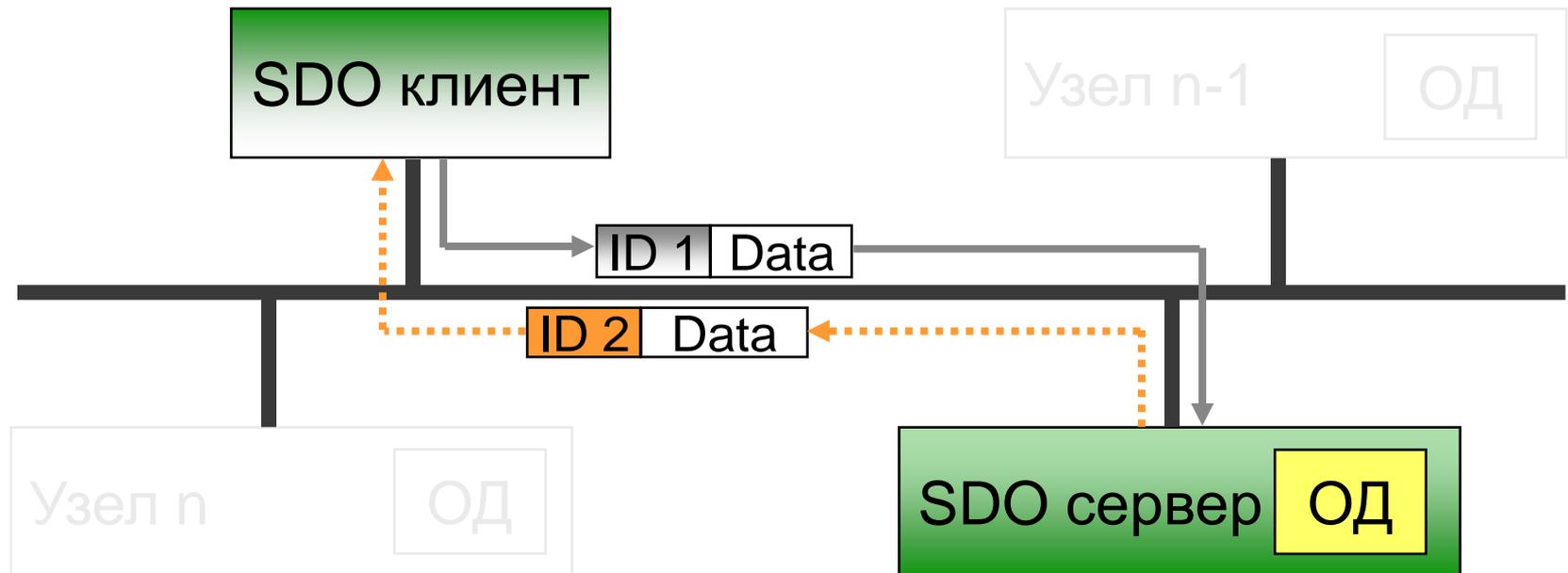
# Коммуникационные объекты

Index (hex)	Object	Name	Type	Acc.	M/O
1000	VAR	device type	Unsigned32	ro	M
1001	VAR	error register	Unsigned8	ro	M
1002	VAR	manufacturer status register	Unsigned32	ro	O
1003	ARRAY	pre-defined error field	Unsigned32	ro	O
1004	Reserved for compatibility reasons				
1005	VAR	COB-ID SYNC-message	Unsigned32	rw	O
1006	VAR	communication cycle period	Unsigned32	rw	O
1007	VAR	synchronous window length	Unsigned32	rw	O
1008	VAR	manufacturer device name	Vis-String	c	O
1009	VAR	manufacturer hardware version	Vis-String	c	O
100A	VAR	manufacturer software version	Vis-String	c	O
100B	Reserved for compatibility reasons				
100C	VAR	guard time	Unsigned32	rw	O
100D	VAR	life time factor	Unsigned32	rw	O
100E	Reserved for compatibility reasons				
100F	Reserved for compatibility reasons				
1010	VAR	store parameters	Unsigned32	rw	O
1011	VAR	restore default parameters	Unsigned32	rw	O
1012	VAR	COB-ID time stamp	Unsigned32	rw	O
1013	VAR	high resolution time stamp	Unsigned32	rw	O
1014	VAR	COB-ID Emergency	Unsigned32	rw	O
1015	VAR	Inhibit Time Emergency	Unsigned16	rw	O
1016	ARRAY	Consumer Heartbeat Time	Unsigned32	rw	O
1017	VAR	Producer Heartbeat Time	Unsigned16	rw	O
1018	RECORD	identity object	Identity	ro	M



# Сервисный объект данных (SDO)

Обмен «равный к равному»



SDO протоколы:

- ▶ Ускоренный
- ▶ Сегментированный
- ▶ Блочный

# Формат ускоренного SDO сообщения



Младший  
байт кадра

Старший  
байт кадра

Каждое ускоренное или сегментированное SDO сообщение и каждый блок данных для блочного протокола:

- ▶ Либо подтверждается
- ▶ Либо отвергается

**Задействованы и передаются всегда все 8 байт данных CAN-кадра!**

# Сегментированная передача SDO

Клиент

Сервер



# SDO параметр в объектном словаре

Индекс	Суб-индекс	Описание	Тип данных
12XXh	0h	Число записей	Unsigned8
	1h	COB-ID клиент → сервер	Unsigned32
	2h	COB-ID сервер → клиент	Unsigned32
	3h	Номер узла клиент/сервер	Unsigned8



Старший бит

Младший бит

**31:** SDO существует (0), не существует (1)

**30:** зарезервирован (всегда 0)

**29:** стандартный формат кадра (0), расширенный формат кадра (1)

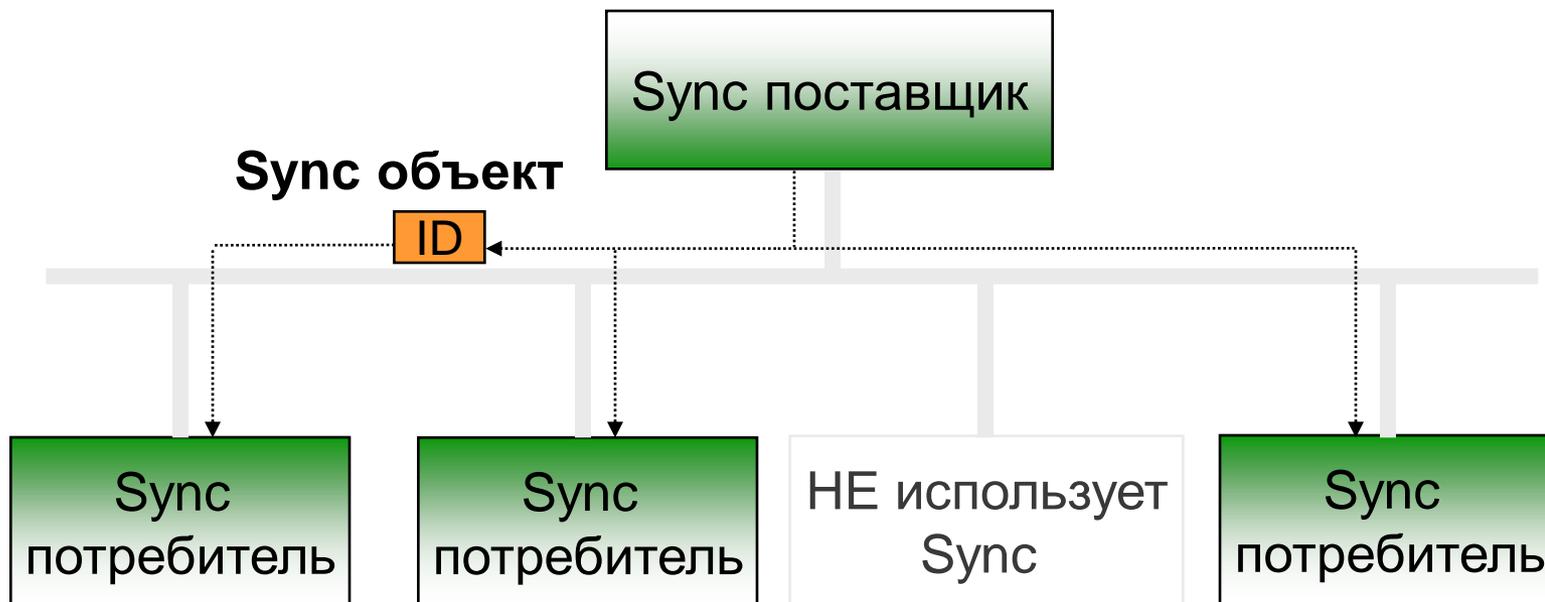
# Набор SDO параметров

SDO параметры сервера (22H)					
1200	RECORD	1 SDO параметр сервера	SDO параметр	ro	O
1201	RECORD	2 SDO параметр сервера	SDO параметр	rw	M/O**
.....	.....	.....	.....	.....	.....
127F	RECORD	128 SDO параметр сервера	SDO параметр	rw	M/O**
SDO параметры клиента (22H)					
1280	RECORD	1 SDO параметр клиента	SDO параметр	rw	M/O**
1281	RECORD	2 SDO параметр клиента	SDO параметр	rw	M/O**
.....	.....	.....	.....	.....	.....
12FF	RECORD	128 SDO параметр клиента	SDO параметр	rw	M/O**
1300		зарезервирован			
.....	.....	.....	.....	.....	.....
13FF		зарезервирован			

\*\* M/O (Mandatory/Optional)

Если устройство поддерживает SDO, то определение соответствующего параметра в Объектном Словаре является обязательным.

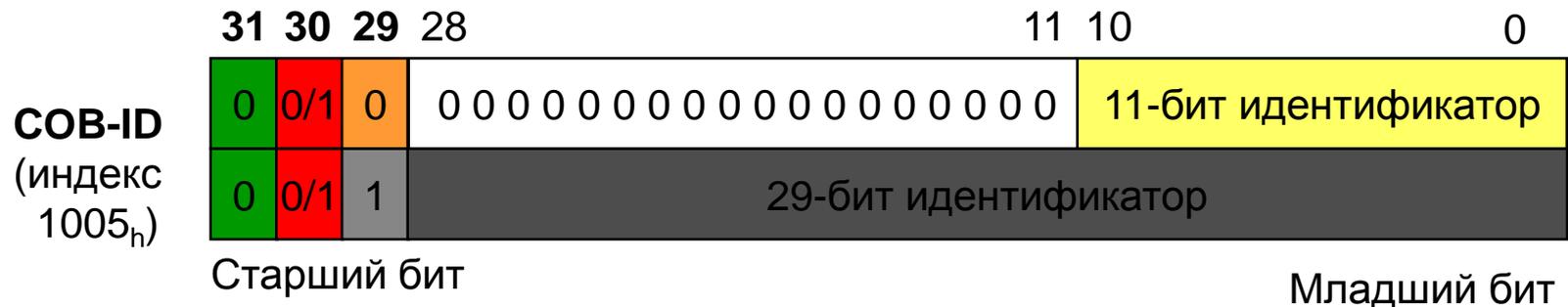
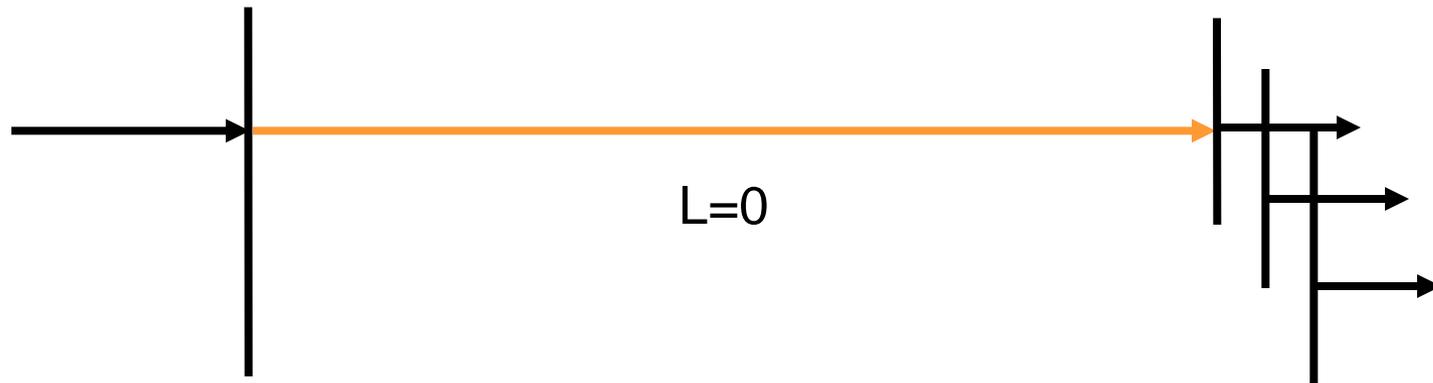
# Объект синхронизации Sync



# Sync протокол

Sync-поставщик

Sync-потребители

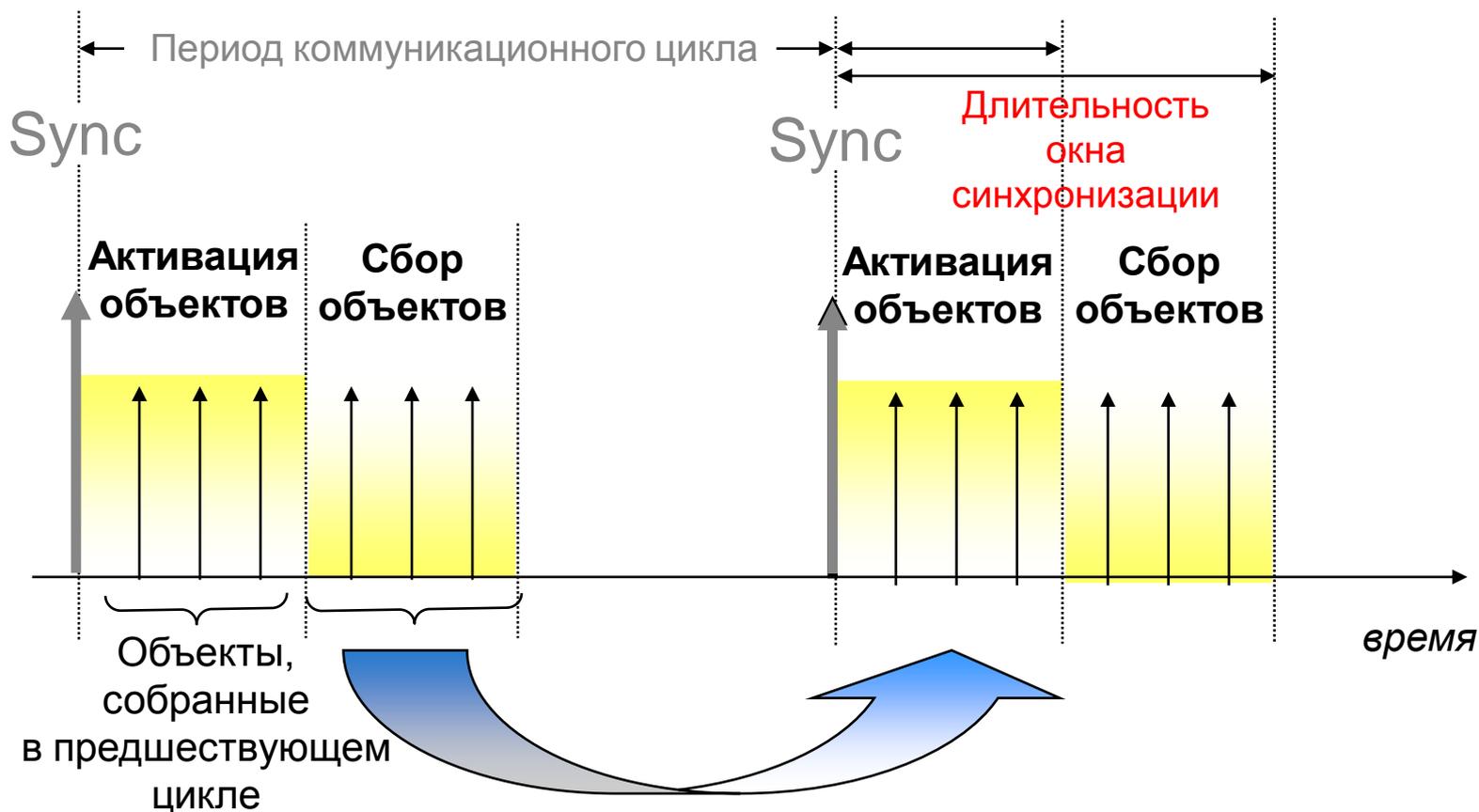


**31:** не используется

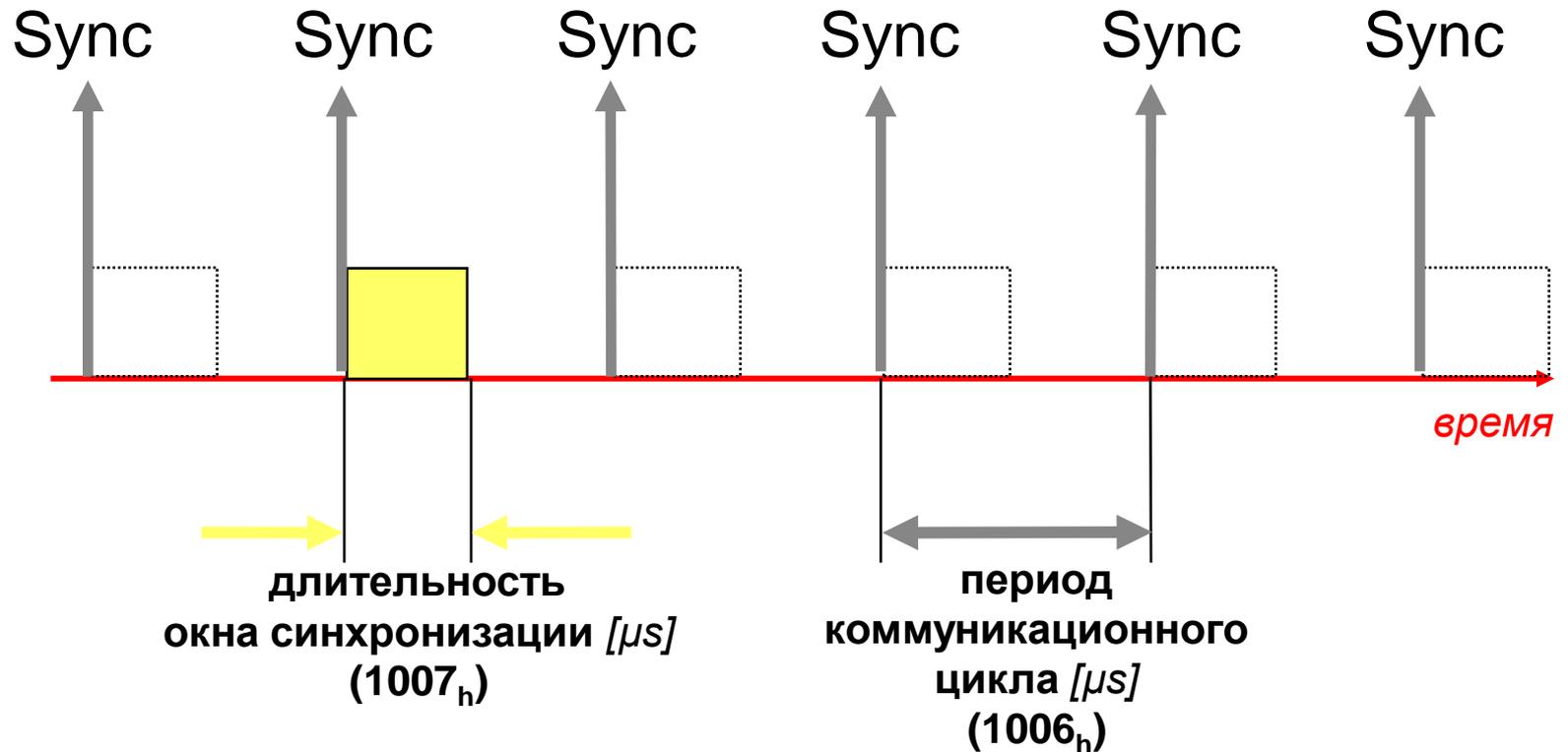
**30:** Sync источник (1), Sync потребитель (0)

**29:** стандартный формат кадра (0), расширенный формат кадра (1)

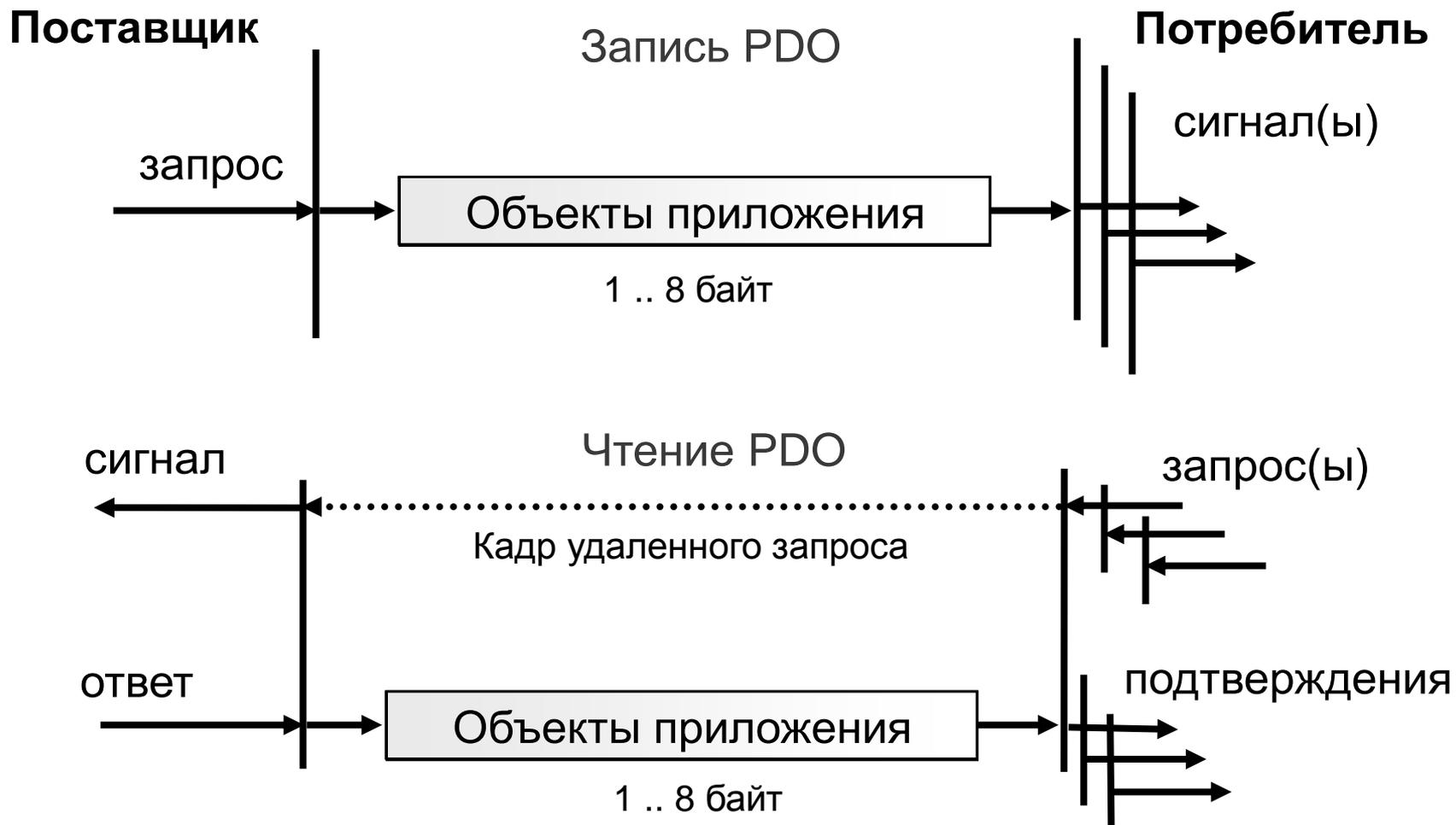
# Синхронные операции



# Определения интервалов Sync



# Объект данных процесса PDO



# Набор PDO параметров

Коммуникационный параметр принимаемого PDO (20H)					
1400	RECORD	1 принимаемый PDO параметр	PDOCommPar	rw	M/O*
1401	RECORD	2 принимаемый PDO параметр	PDOCommPar	rw	M/O*
.....	.....	.....	.....	.....	.....
15FF	RECORD	512 принимаемый PDO параметр	PDOCommPar	rw	M/O*
Параметр отображения принимаемого PDO (21H)					
1600	ARRAY	Отображение 1 принимаемого PDO	PDOMapping	rw	M/O*
1601	ARRAY	Отображение 2 принимаемого PDO	PDOMapping	rw	M/O*
.....	.....	.....	.....	.....	.....
17FF	ARRAY	Отображение 512 принимаемого PDO	PDOMapping	rw	M/O*
Коммуникационный параметр передаваемого PDO (20H)					
1800	RECORD	1 передаваемый PDO параметр	PDOCommPar	rw	M/O*
1801	RECORD	2 передаваемый PDO параметр	PDOCommPar	rw	M/O*
.....	.....	.....	.....	.....	.....
19FF	RECORD	512 передаваемый PDO параметр	PDOCommPar	rw	M/O*
Параметр отображения передаваемого PDO (21H)					
1A00	ARRAY	Отображение 1 передаваемого PDO	PDOMapping	rw	M/O*
1A01	ARRAY	Отображение 2 передаваемого PDO	PDOMapping	rw	M/O*
.....	.....	.....	.....	.....	.....
1BFF	ARRAY	Отображение 512 передаваемого PDO	PDOMapping	rw	M/O*

- Если устройство поддерживает PDO, соответствующие записи объектного словаря для коммуникационного параметра и параметра отображения являются обязательными. Они могут быть доступны только по чтению.

# Коммуникационный PDO параметр

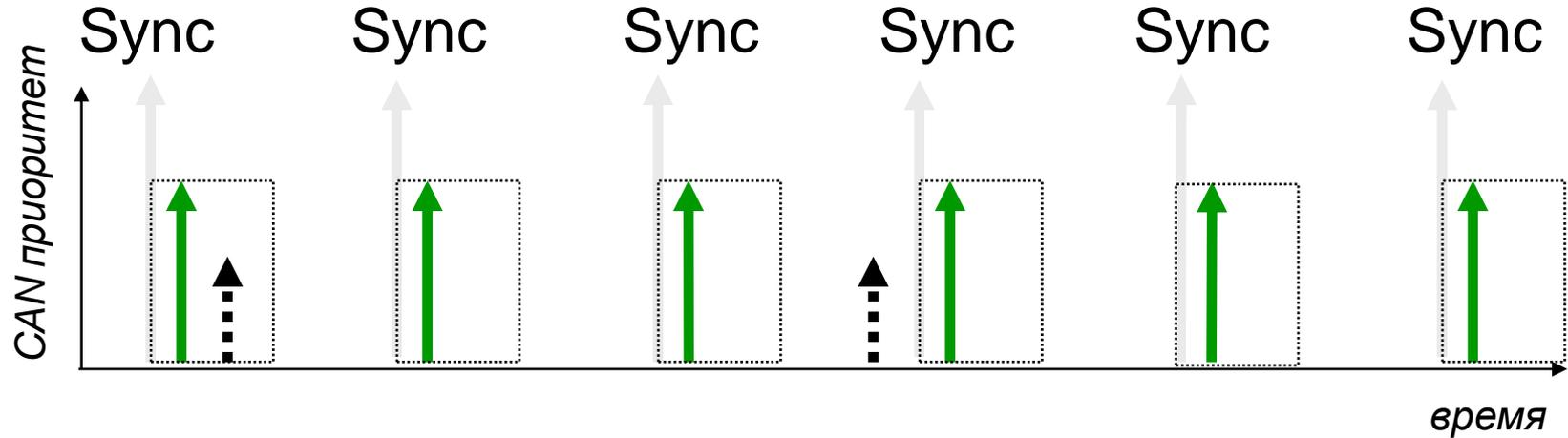
Индекс	Суб-индекс	Описание	Тип данных
1400h	0h	число записей	Unsigned8
...	1h	COB-ID	Unsigned32
15FFh	2h	тип передачи	Unsigned8
1800h	3h	время запрета	Unsigned16
...	4h	зарезервирован	Unsigned8
19FFh	5h	таймер события	Unsigned16



# Режимы PDO



# Синхронные PDO

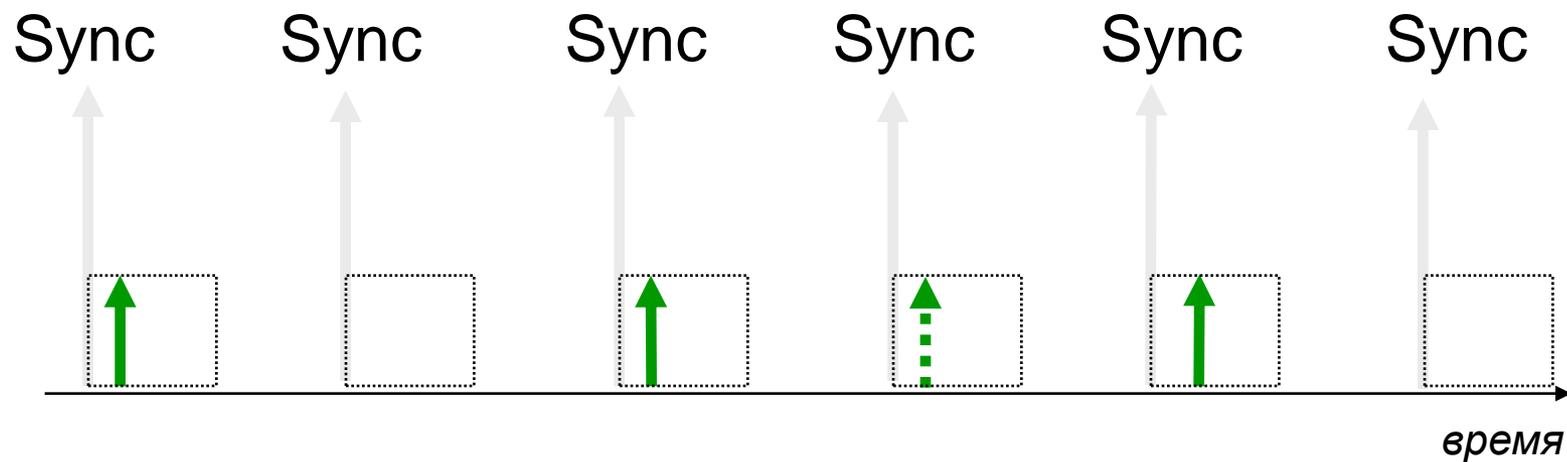


синхронные PDO



асинхронные PDO

# Циклические и ациклические PDO



синхронные циклические PDO



синхронные ациклические PDO

# Параметр PDO отображения

Индекс	Суб-индекс	Описание	Тип данных
1600h	0h	число записей	Unsigned8
...	1h	1 объект	Unsigned32
17FFh	2h	2 объект	Unsigned32
1A00h	..	..	..
...	40h	64 объект	Unsigned32
1BFFh			

## Отображение PDO объекта

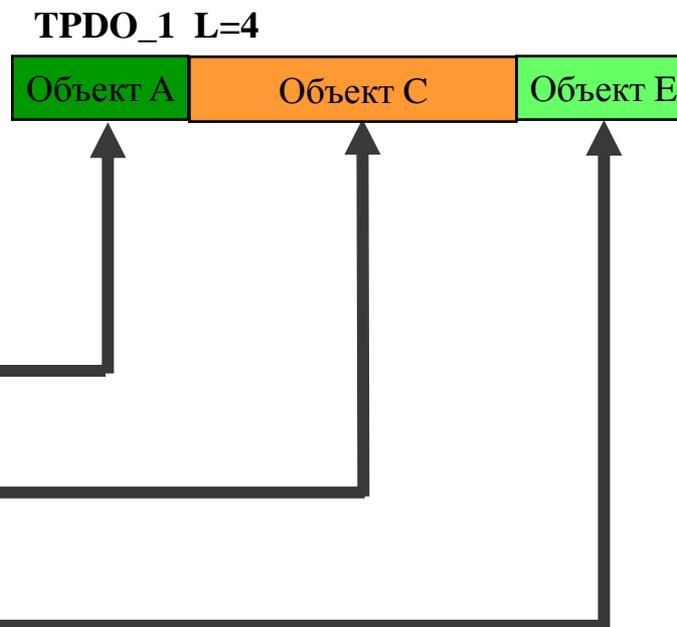
31	16 15	8 7	0
Индекс объекта	Суб-индекс	Длина в битах	

# Отображение объектов в PDO

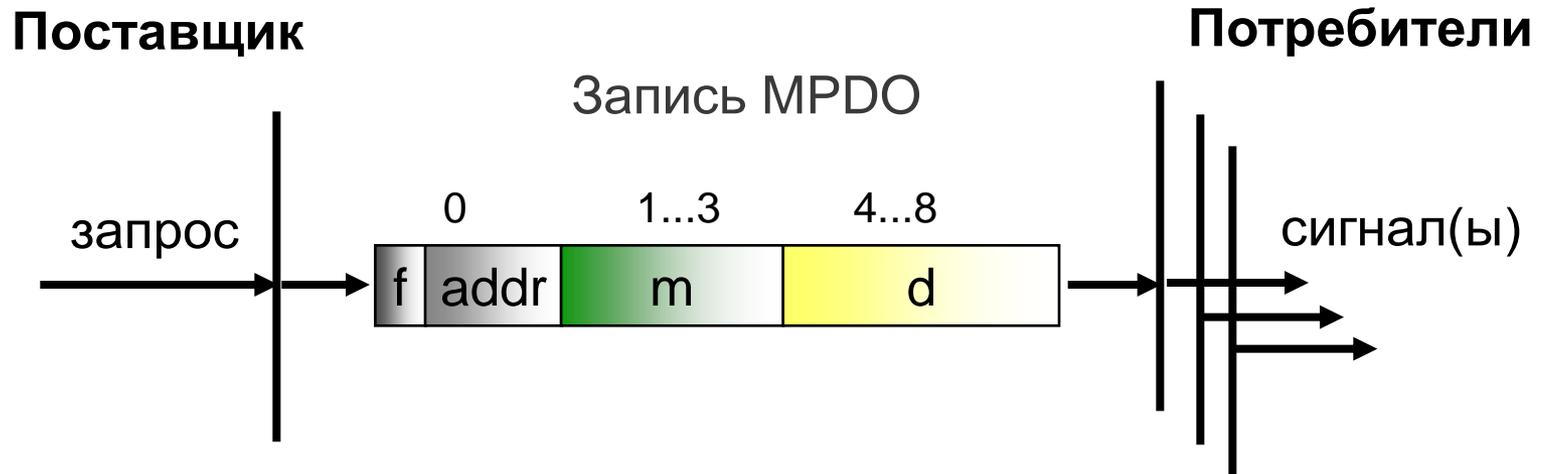
## Объектный словарь

Индекс	Суб	Содержит объект	
1A00h	01h	2000h 01h	8
1A00h	02h	2001h 00h	16
1A00h	03h	2003h 01h	8

Индекс	Суб	Содержит объект
2000h	01h	Объект А
2000h	02h	Объект В
2001h	00h	Объект С
2002h	00h	Объект D
2003h	01h	Объект Е
2003h	02h	Объект F
2003h	03h	Объект G



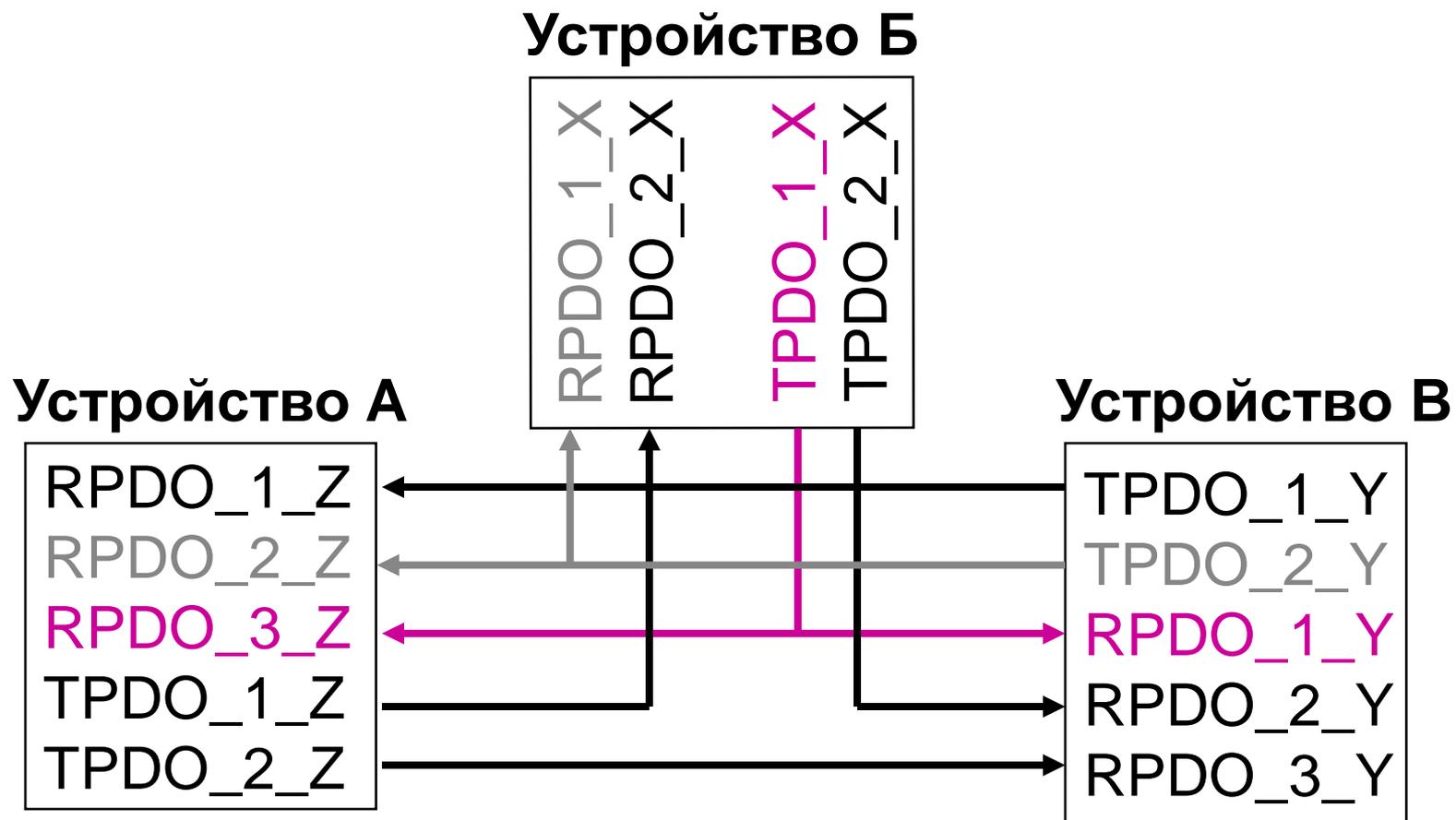
# Мультиплексированный PDO



**f = 0 (Режим адреса источника)**  
**addr = номер узла поставщика**  
**m = индекс и субиндекс (мультиплексор)**  
**d = данные**

**f = 1 (Режим адреса назначения)**  
**addr = номер узла потребителя**  
**m = индекс и субиндекс (мультиплексор)**  
**d = данные**

# Связывание PDO



Получаем CAN сеть, которая «живет сама по себе»

БЕЗ «ВЕДУЩЕГО»

# Протокол временной метки

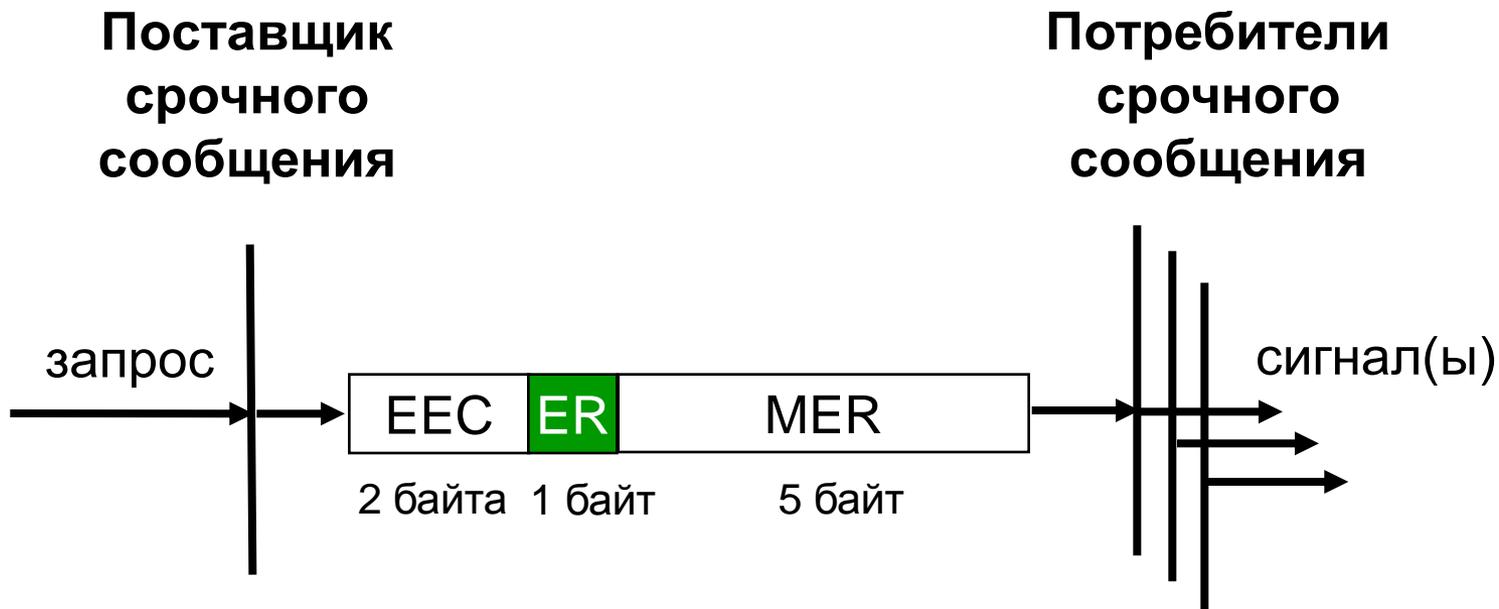
Поставщик  
временной метки

Потребители  
временной метки



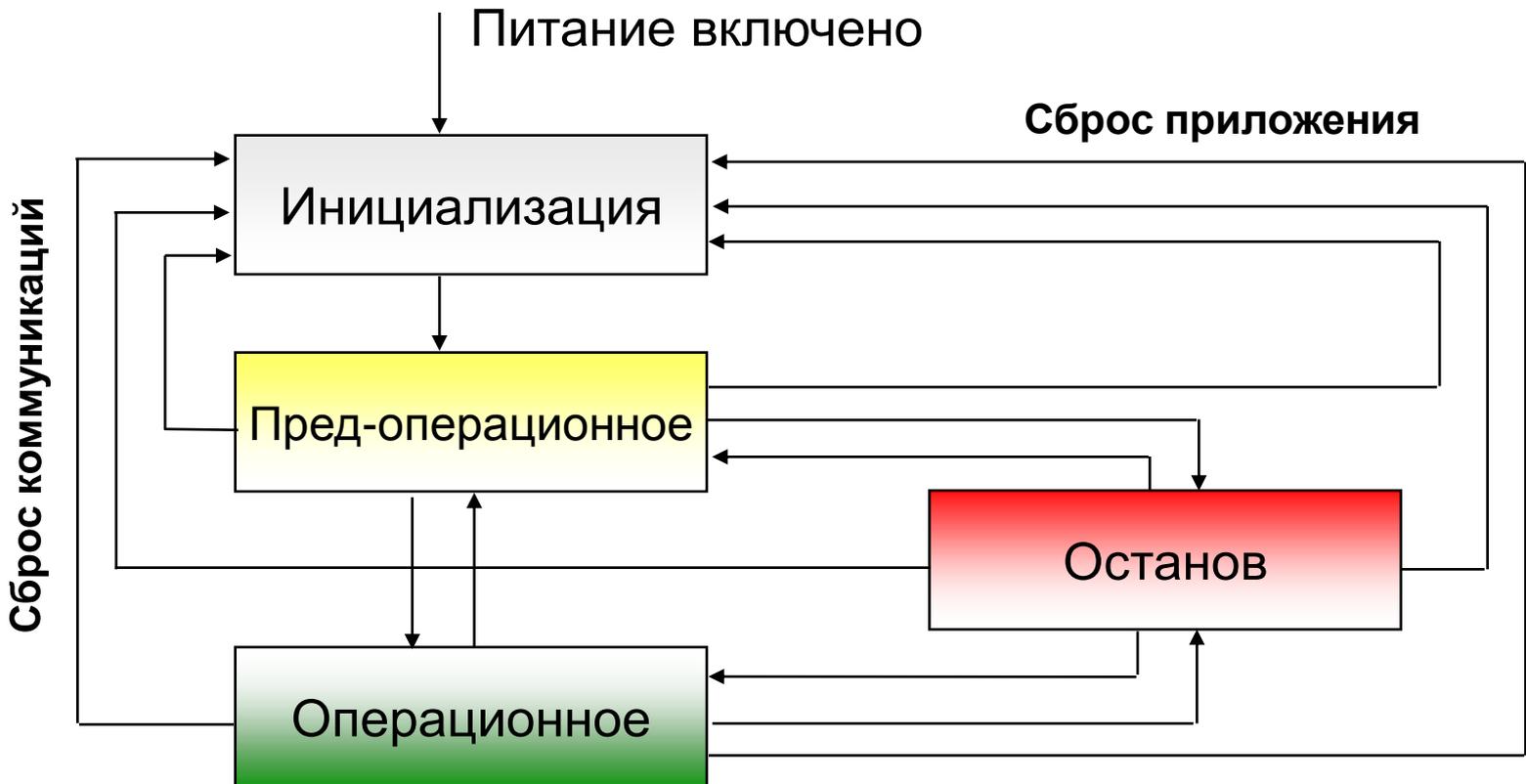
STRUCT OF  
UNSIGNED28 ms, (после полуночи)  
VOID4 reserved\_1,  
UNSIGNED16 days, (с 1 января 1984 г.)

# Протокол срочного сообщения

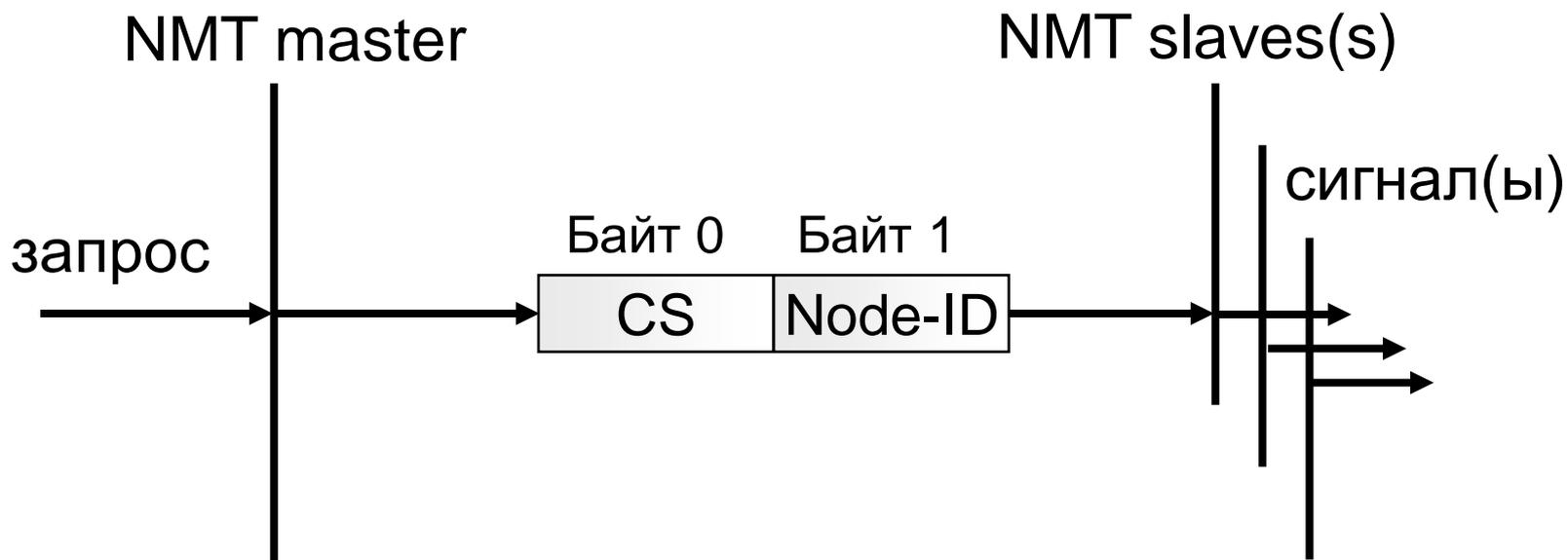


**ЕЕС** Код ошибки срочного сообщения  
**ER** Регистр ошибки ( $1001_h$ )  
**MER** Особый код ошибки

# Диаграмма состояний NMT slave



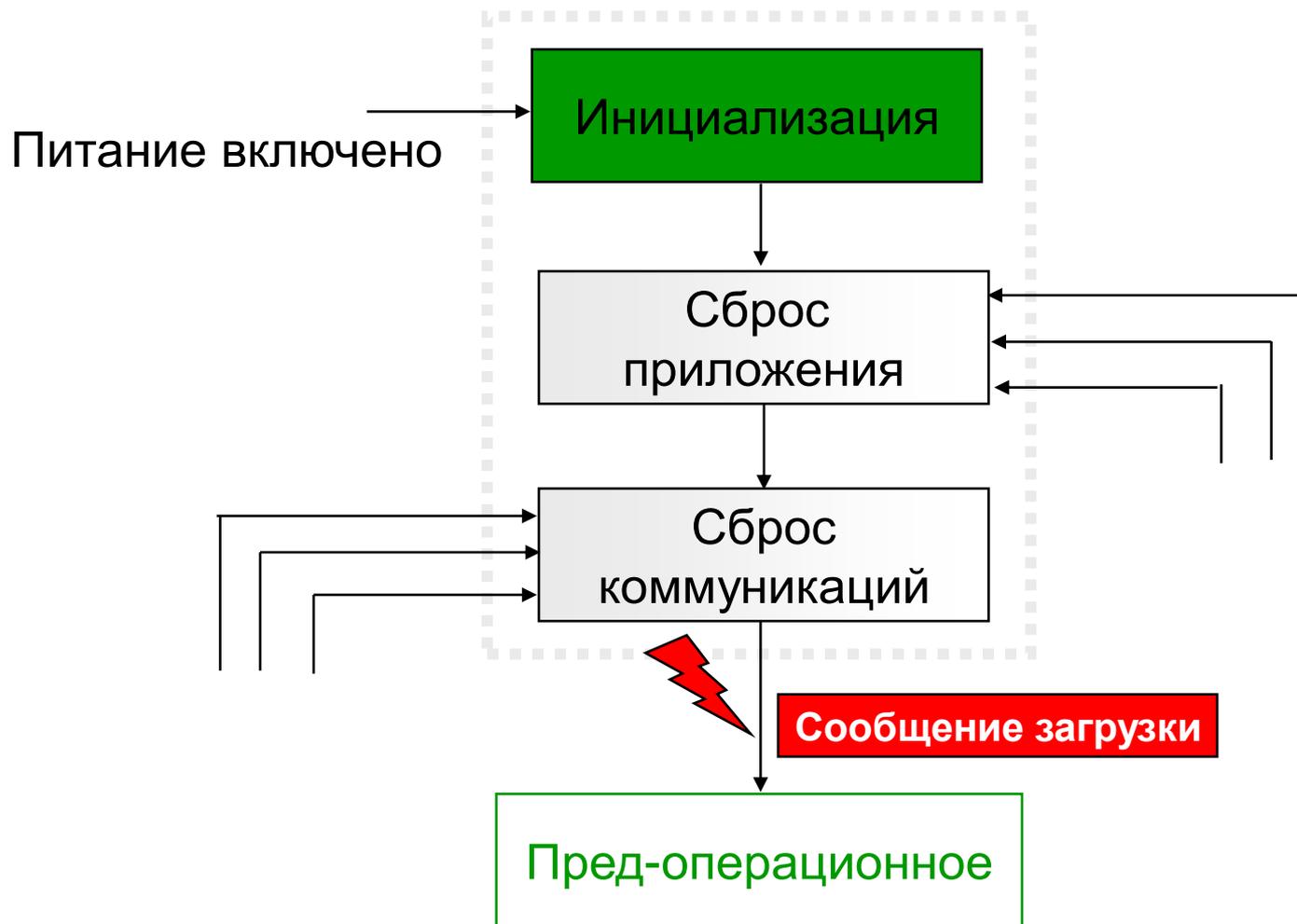
# ***NMT протокол***



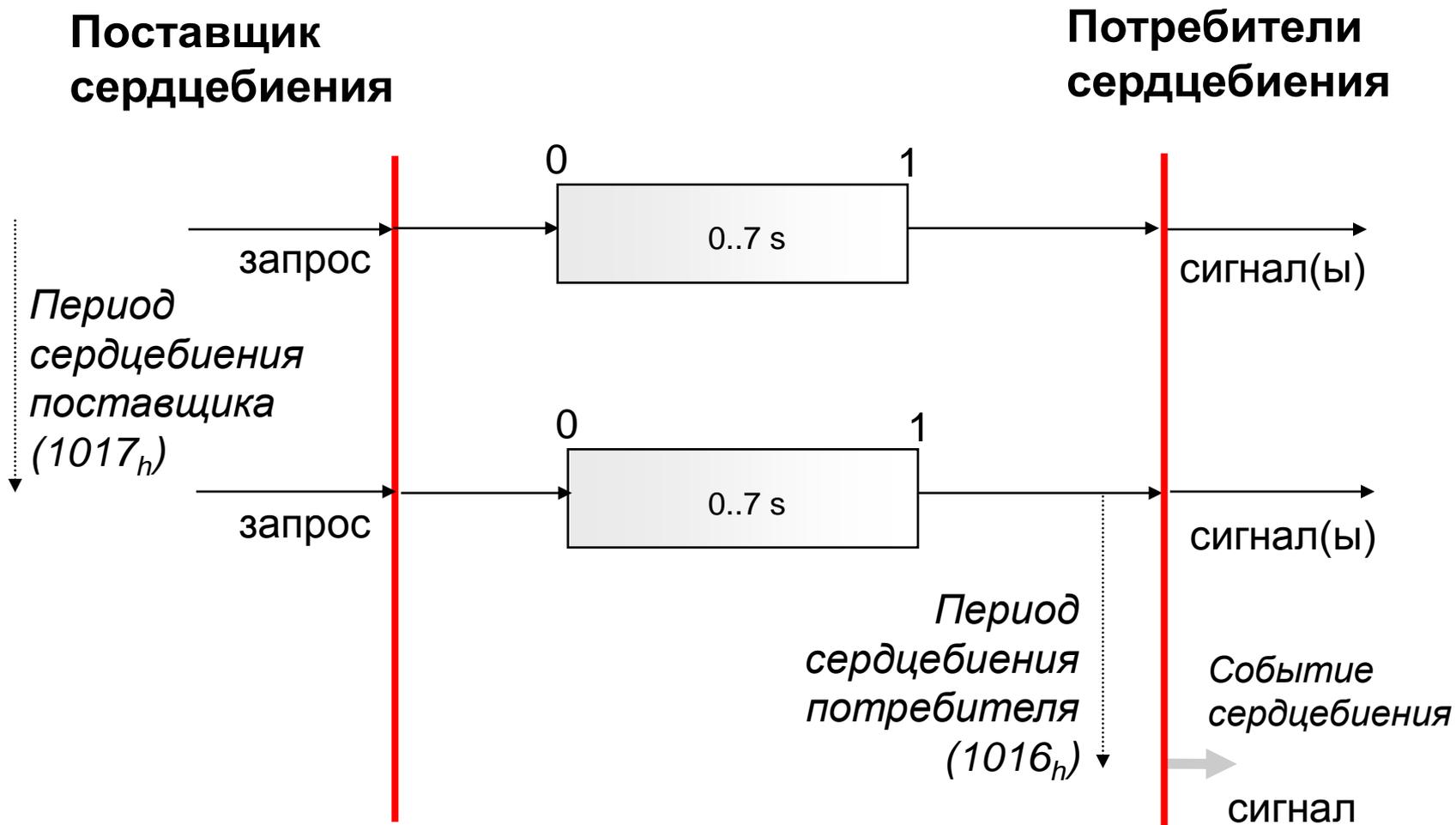
**Код команды CS задает новое состояние узла Node-ID**

Start Remote Node (CS=1),  
Stop Remote Node (CS=2),  
Enter Pre-Operational (CS=128),  
Reset Node (CS=129) and  
Reset Communication (CS=130).

# Состояние инициализации



# Протокол сердцебиения



# *Коммуникационные протоколы (повтор)*

- Протокол PDO (объект данных процесса)
- Протокол SDO (сервисный объект данных)
- Протоколы специальных объектов:
  - Протокол синхронизации SYNC
  - Протокол временной метки TIME
  - Протокол срочного сообщения EMCY
- Протоколы управления сетью:
  - NMT протокол
  - Boot-Up протокол (загрузка узла)
  - Протокол сердцебиения

**В этих абстракциях описывается поведение сети !**



# CAN-технология и обучение современным технологиям автоматизации

## CAN-технология – составные части

- разработка и изготовление аппаратных средств автоматизации для построения систем с CAN-bus
- разработка и изготовление интерфейсных устройств CAN-bus для универсальных компьютерных платформ (ISA, PC104, PCI, USB... )
- программное обеспечение для реализации протоколов верхнего уровня
- формализация и описание прикладных систем в терминах и категориях протоколов верхнего уровня
- средства диагностики и конфигурирования сетей CAN
- обучение, проведение семинаров, выпуск литературы

# CAN-технология и обучение современным технологиям автоматизации

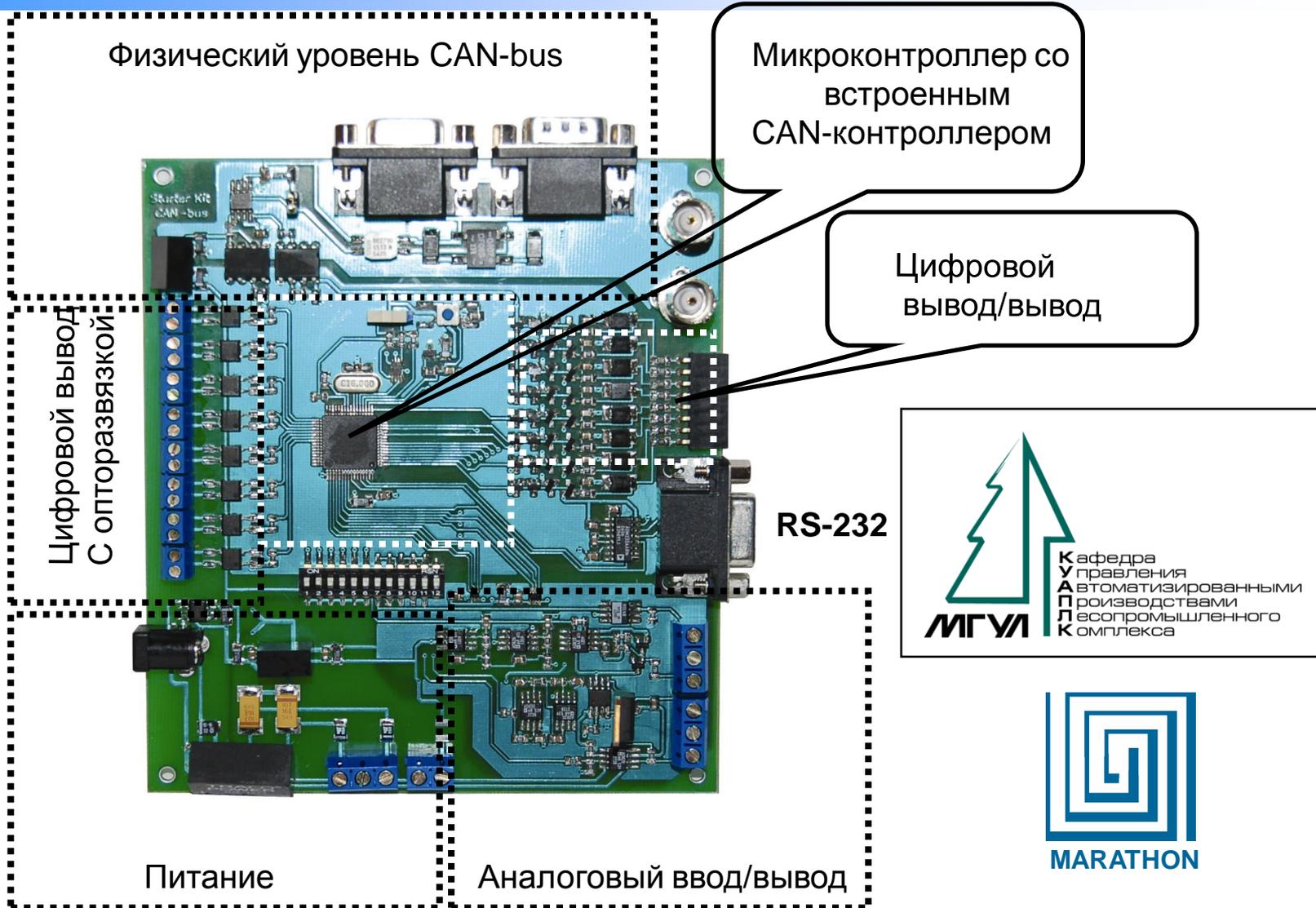
## Почему CAN ?

- технология устоявшаяся, 15 лет с момента начала широкого внедрения
- одобрена мировым сообществом через принятие международных стандартов (ISO 11898, EN 50235)
- универсальность технологии, очень широкая область применения
- востребованность как компонентов, так и специалистов на растущем рынке средств автоматизации

## Вывод для ВУЗов России:

**Освоение российским инженерами технологии CAN и интеграция российских разработок на мировом рынке средств автоматизации является единственным шансом для сохранения и существования в России разработок в области автоматизации.**

# CAN-bus стартер-кит для обучения студентов



Кафедра  
Управления  
Автоматизированными  
Производствами  
Лесопромышленного  
Комплекса

**МГУИЛ**



**Спасибо за внимание !**

**Ваши вопросы,  
пожалуйста !**

