

CAN vs. Ethernet



CAN and Ethernet, the early days

Ethernet

- Published 1976
- Commercial products 1980
- IEEE 802.1 1983
- 100 k interfaces sold 1985

CAN

- Published 1986
- Commercial products 1988
- ISO 11898 1993
- Unique features
 - **Non-destructive collision resolution**
 - **System wide data consistency**

CAN vs. Ethernet

COMPARISON CAN/ETHERNET 1988

		CAN	Ethernet Coaxial Bus (ECB)	Ethernet Twisted Pair (ETP)
1	Topology	Bus	Bus	Star
2	Media	Twisted Pair	Thin coaxial	Twisted Pair
3	Bus length	5 km to 40 m	185 m	0 m
4	Drop length	30 cm (or more)	0 cm	250 m
5	Bitrate	10 kbps to 1 Mbps	10 Mbps	1 Mbps
6	Min frame length	47 bit (47 us at 1Mbps)	672 bit (67 us)	672 bit (672 us)
7	Max frame length	160 bit (.16 ms at 1Mbps)	12,240 bit (1.224 ms)	12,240 bit (12.24 ms)
8	Collision resolution without frame loss	Yes	No	No



CAN vs. Ethernet

COMPARISON CAN/ETHERNET 1988

		CAN	Ethernet Coaxial Bus (ECB)	Ethernet Twisted Pair (ETP)
9	Max message collision delay	Predictable (160 us at 1 Mbps) for highest priority message	Unpredictable (more than 1.2 ms)	Unpredictable (but more than 1 2.2 ms)
10	Addressing method	Offline addressing	Source Destination	Source Destination
11	Retransmission on error	Yes	No	No
12	Max retransmission delay on error	187 bit for highest priority message	NA	NA
13	Data consistency guaranteed within the system	Yes	No	No

It is quite clear that CAN was a far better alternative for distributed embedded control systems in 1988!

CAN vs. Ethernet

WHAT HAPPENED WITH CAN AND ETHERNET BETWEEN 1988 AND 2017?

GENERAL TREND

	1988	2017
Transistors/sq mm	3k Lars-Berno Fredriksson 170805	25,000k
Clock frequency	12.5 MHz	8,000 MHz
Chip cores	1	30+
Computing capacity	1	10,000

WHAT HAPPENED WITH CAN AND ETHERNET BETWEEN 1988 AND 2017?**ETHERNET**

Year	Ethernet standard	Description
(1987)	802.3en	1BASE5 1 Mbps over twisted pair in a star topology. Bus topology was abandoned for future Ethernet development.
1990	802.3i	10BASE-T 10 Mbps over twisted pair
1998	802.3y	100BASE-T2 100 Mbps over low quality twisted pair
1999	802.3ab	1000BASE-T Gigabit Ethernet over twisted pair at 1 Gbps
2006	802.3an	10GBASE-T 10 Gigabit Ethernet over unshielded twisted pair (UTP)
2015	802.bw	100BASE-T1 – 100 Mbps Ethernet over a single twisted pair for automotive applications
2016	802.3bp	1000BASE-T1 – Gigabit Ethernet over a single twisted pair, automotive & industrial environments
2017	802.3bz	2.5GBASE-T and 5GBASE-T – 2.5 Gigabit and 5 Gigabit Ethernet over Cat-5/Cat-6 twisted pair

ETHERNET HAS TAKEN ADVANTAGE OF TECHNOLOGICAL ACHIEVEMENTS

WHAT HAPPENED WITH CAN AND ETHERNET BETWEEN 1988 AND 2017?**CAN**

Year	CAN standard	Description
1988	CAN 1.0	Classical CAN, 11 bit identifier
1991	CAN 2.0A/B	Bosch's CAN specification 2.0 published A 11 bit identifier, B 29 bit identifier
1993	ISO 11898	CAN standardized by ISO including a transceiver solution
1994		Transceiver chips commercially available
1995	ISO 11898	29 bit identifier added to the ISO standard
2012	CAN FD	Bosch releases CAN FD 1.0
2015	ISO 11898-1 : 2015	CAN FD (and some other CAN variants) is included in the ISO standard

**CAN HAS NOT TAKEN ADVANTAGE OF TECHNOLOGICAL ACHIEVEMENTS
EXCEPT FOR A HIGHER CLOCK FREQUENCY**

A BIT IS A BIT AND A CABLE IS A CABLE



Modern fourth generation jet
fighters using
MIL-STD-1553 1 Mbps
Less efficient than CAN.



Does a car really need a faster control system than an inherently unstable jet?

A BIT IS A BIT AND A CABLE IS A CABLE

- Higher bitrate → More even impedance within the harness
- Higher bitrate → Active Star topology (HUB)
- Higher bitrate → Echo suppression
- Higher bitrate → More bits per Baud
- Higher bitrate → Forward Error Correction

CAN could run at the same bitrate as Ethernet
if CAN was specified to use the same physical layer technology

**CAN has a vast potential for further enhancement
by taking advantage of modern technology!**



Don't throw the baby out with the bathwater

Bandwidth saving features of CAN

➤ Data consistency

- Broadcasting all messages and every receiver checks for correct reception before accepting the message. Data consistency is a mandatory requirement of any safe security system.

➤ Predictable latency

- CAN can be scheduled in different ways (token passing, message sequence, time, etc.) and different types of schedules can be simultaneously combined. Unschedulable messages can be transmitted at any time.

➤ Message avalanches at emergencies avoided

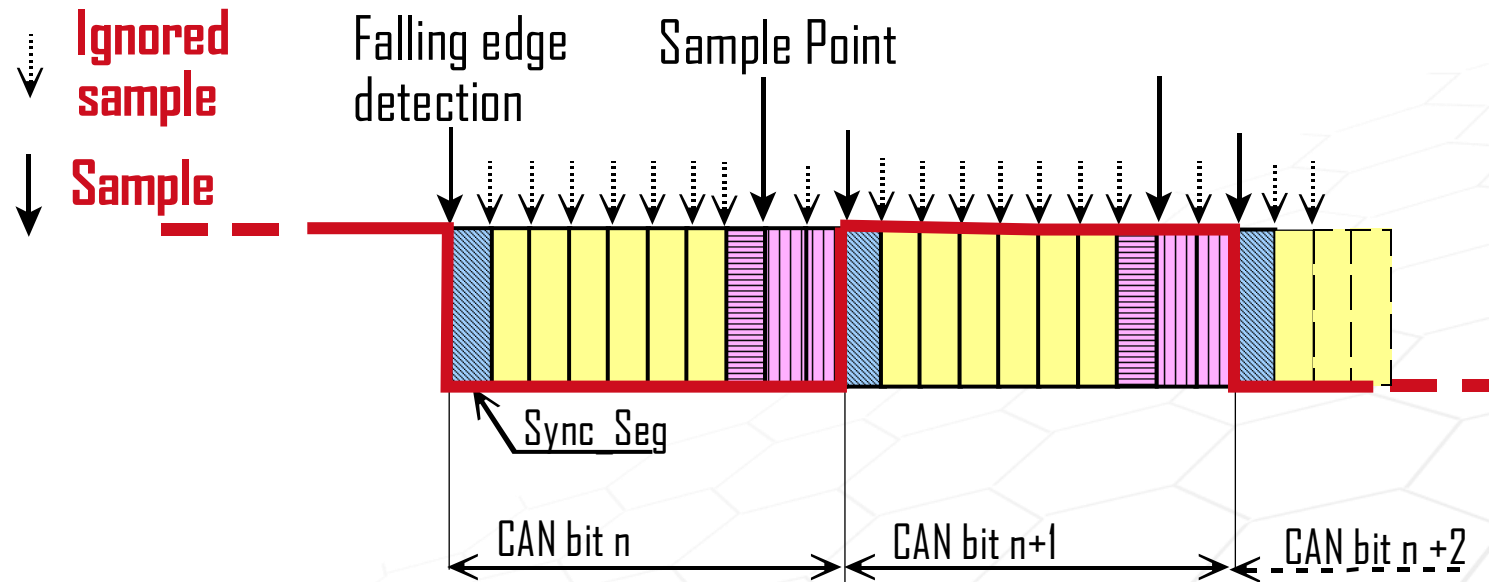
- Using CAN messages with no data for emergency messages, message avalanches at system failures can be easily avoided.



CAN vs. Ethernet

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Classical CAN is only sampling to detect a falling edge and at the Sampling Point. All other samples in a bit are ignored.

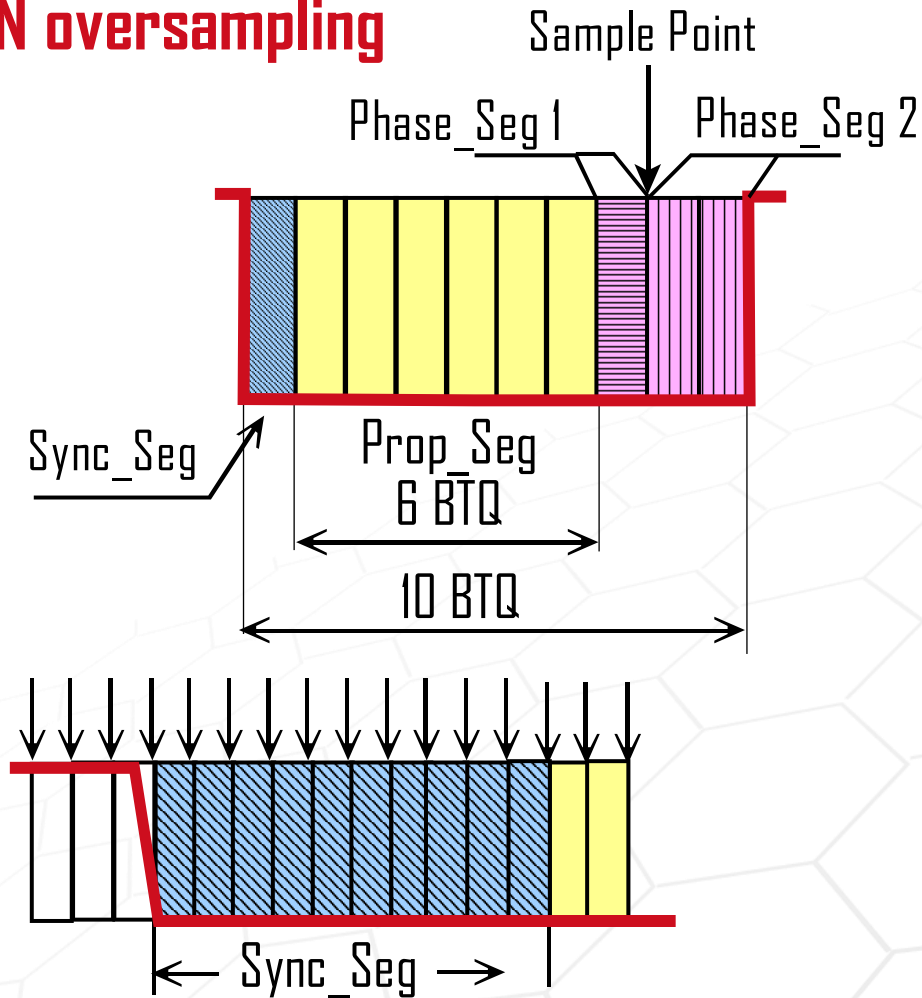


Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

CAN oversampling

EXAMPLE

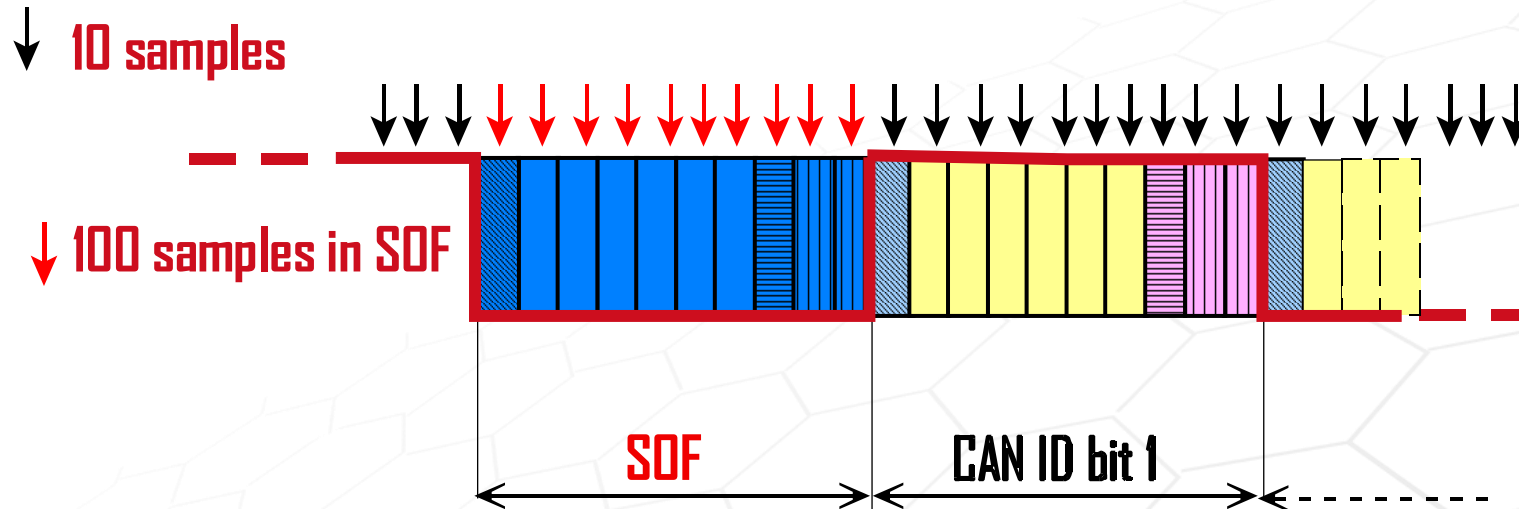
- Classical CAN Bit
 - 10 BTQ
 - 6 BTQ Prop_Seg
- 10 samples per BTQ
- 100 samples per bit



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

CAN oversampling

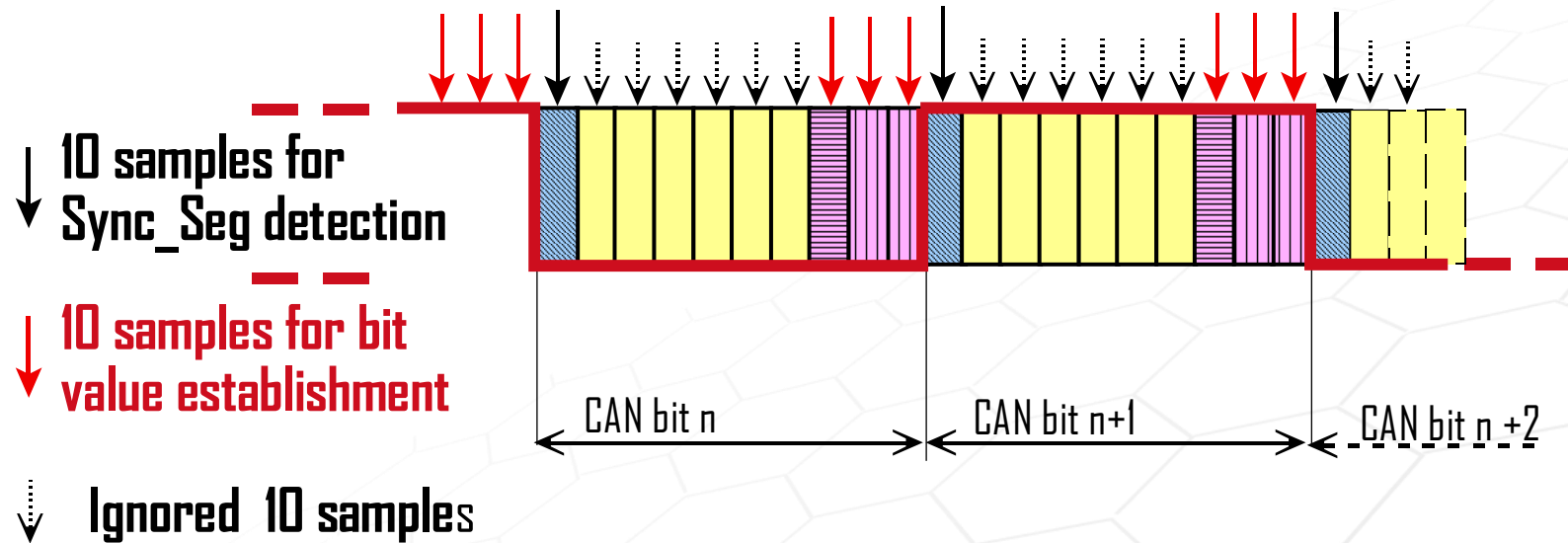
Better SOF detection



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

CAN oversampling

More accurate bit value reading by 30 samples vs 1



CAN vs. Ethernet

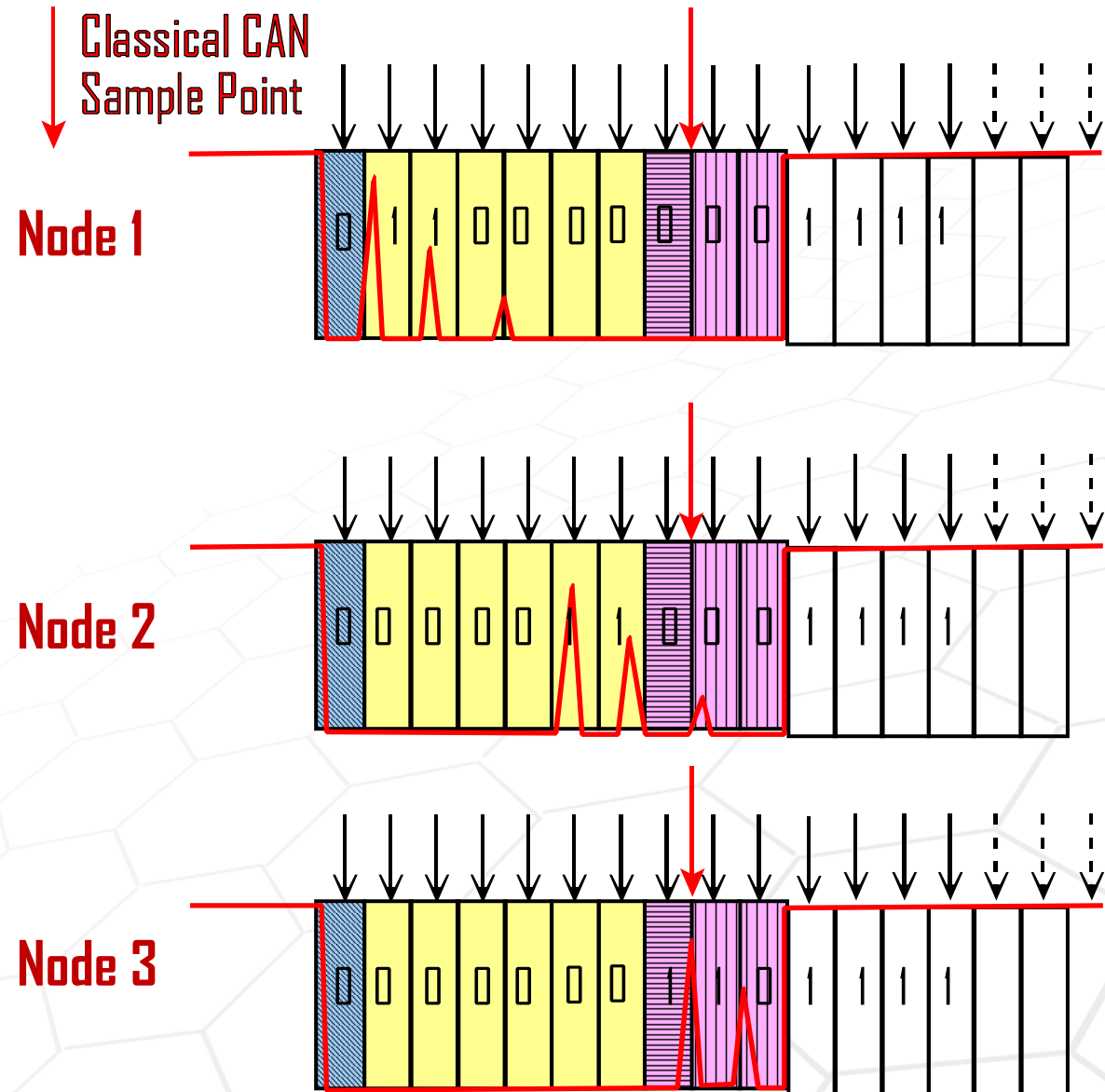
Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

CAN oversampling

Signal quality checking

Detection of disturbance propagation

Classical CAN bit error



KVASER

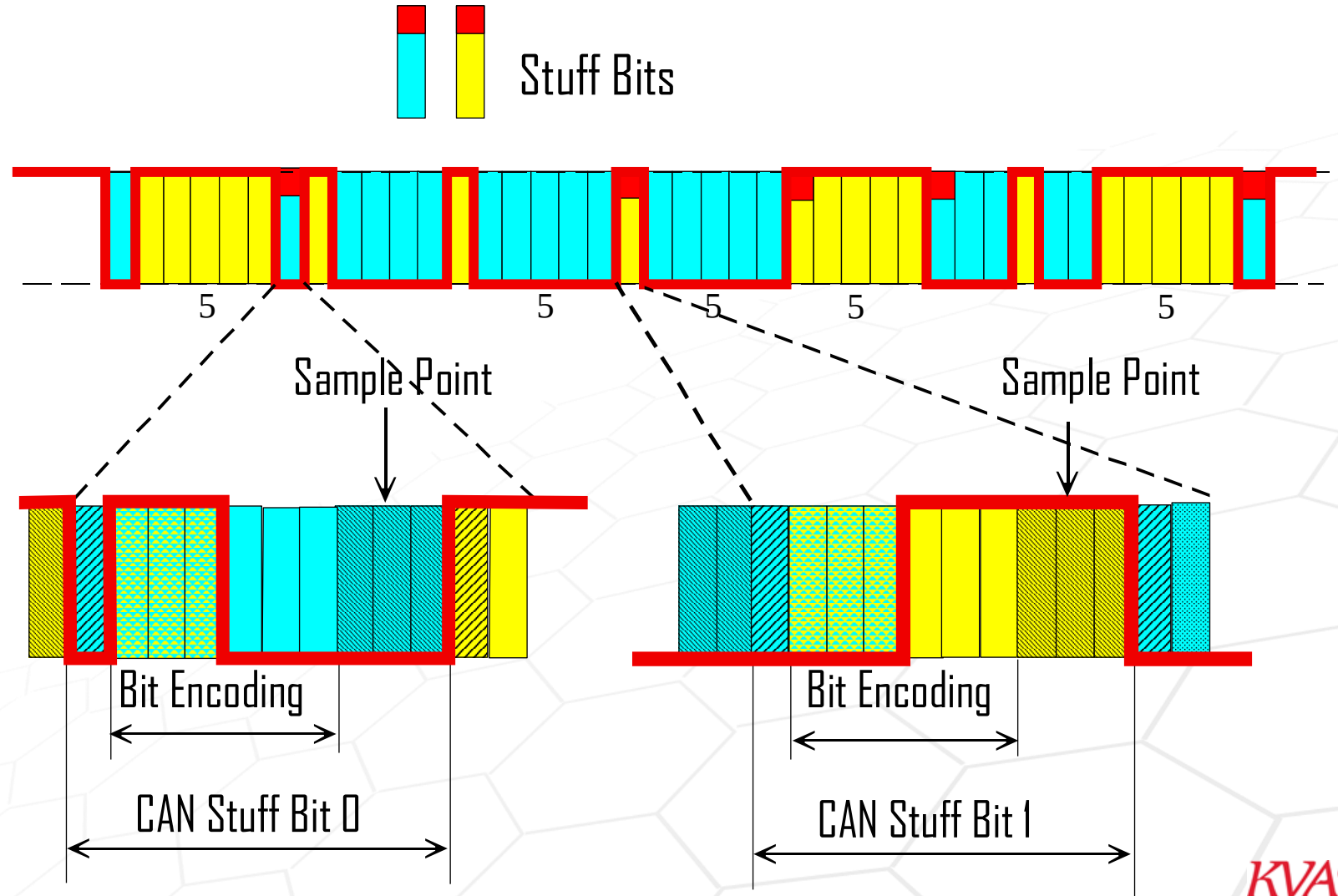
CAN vs. Ethernet

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Bit encoding

Encoding stuff bits

Hamming distance is always 6!

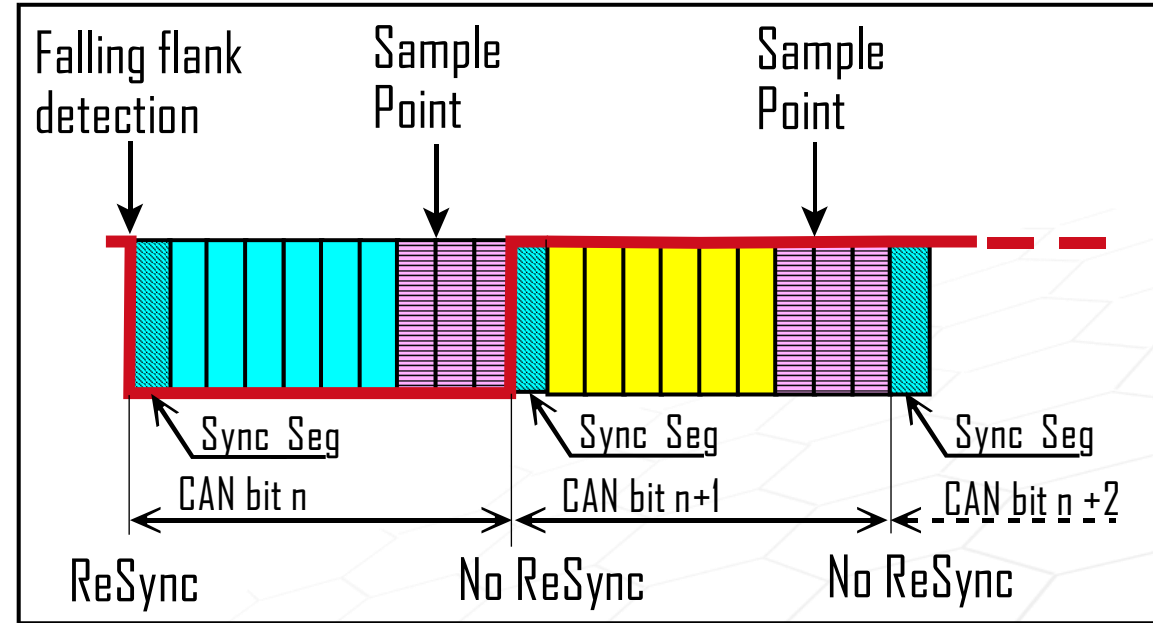


CAN vs. Ethernet

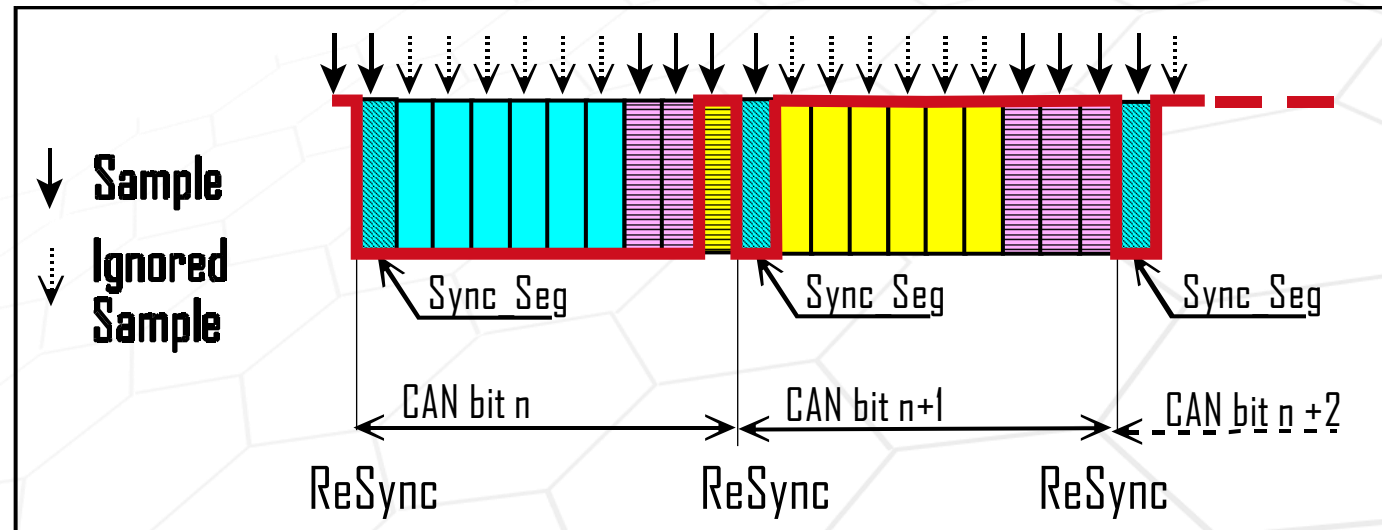
Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Modern transceiver designs allows for more falling edges

Classical CAN
ReSync at
falling edges



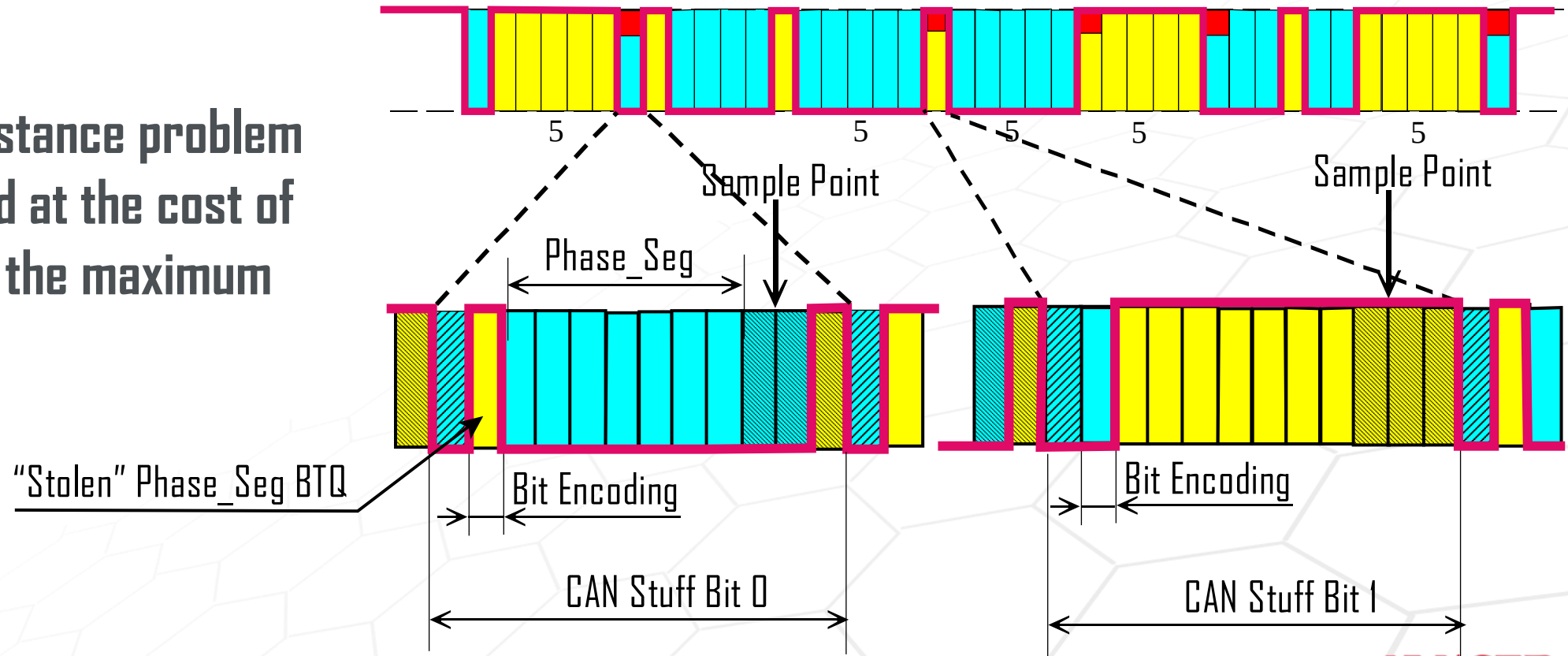
Enhanced
Classical CAN
ReSync at
every bit



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Bonus

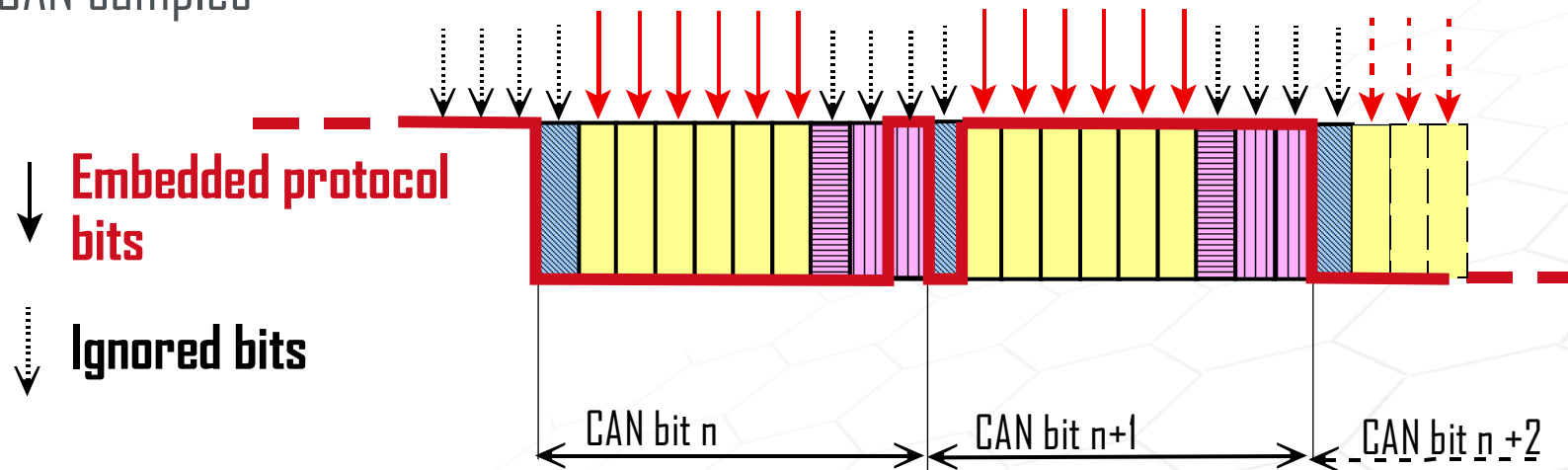
The Hamming distance problem ultimately solved at the cost of a few meters of the maximum bus length



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Use the ignored CAN samples for another protocol that in turn ignores the CAN samples

An embedded protocol



The RTR bit is used to get a falling flank for a hard resynchronization at each receiving node

Different embedded protocols for different CAN IDs makes hacking more difficult

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

An embedded protocol

Protocols for different purposes can be embedded

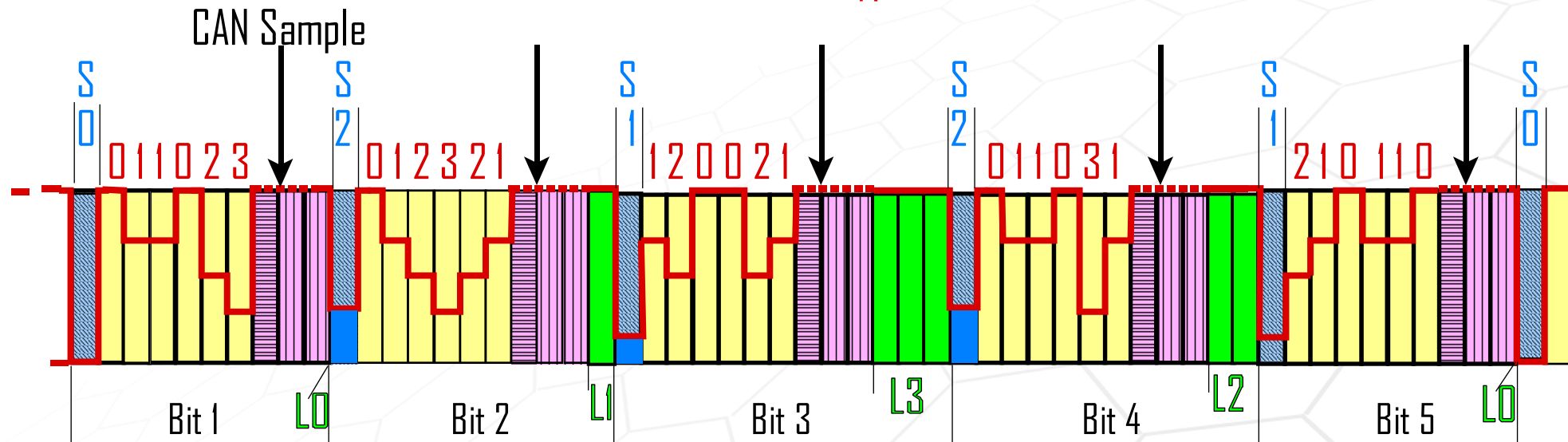
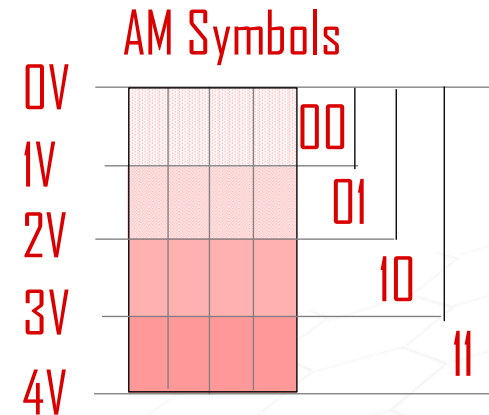
- **Authentication of the transmitter**
- **Encryption key transmission**
- **File transfers**
- **Etc.**

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

More bandwidth into the embedded protocol

Embedded codes

- 4 level AM Prop_Seg coding
- 4 level bit length coding
- 3 level Sync_Seg coding



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Even more bandwidth into the embedded protocol

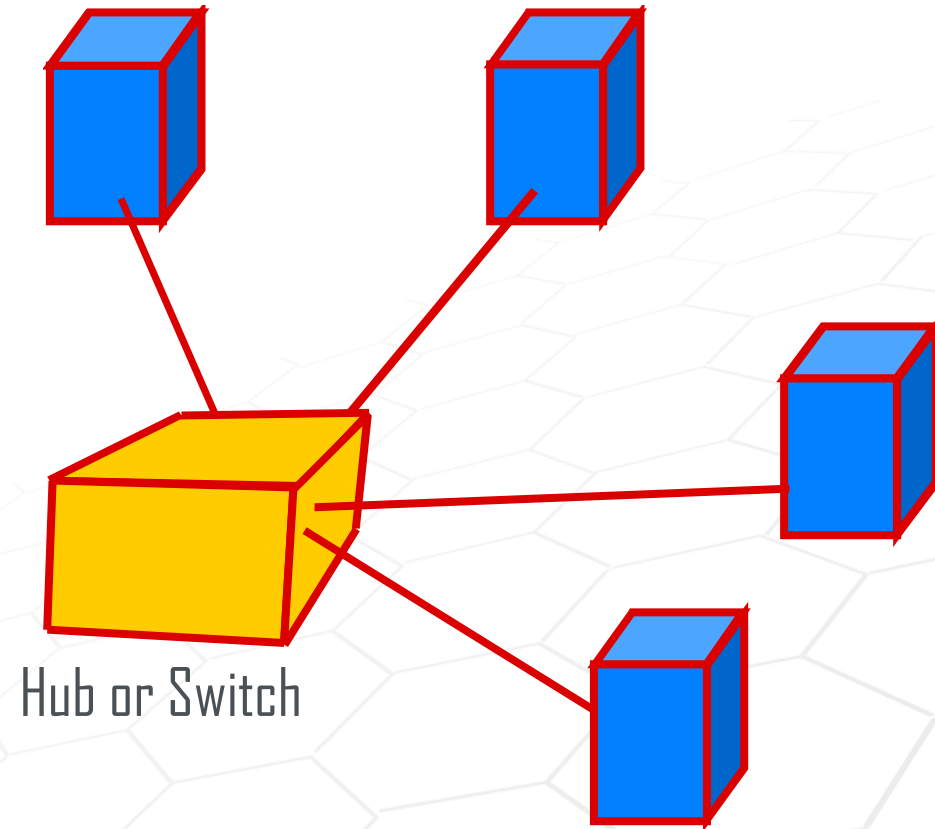
Modern modulation and coding techniques could bring more bandwidth.

Would be good to have a selection of techniques for different needs.

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

A star topology for CAN

- High bandwidth for CAN embedded protocols
- CAN mode for hard real time and dependability
- Ethernet mode for maximum bandwidth



Proven since the early 90s

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

Ethernet and CAN on the same transceiver chip

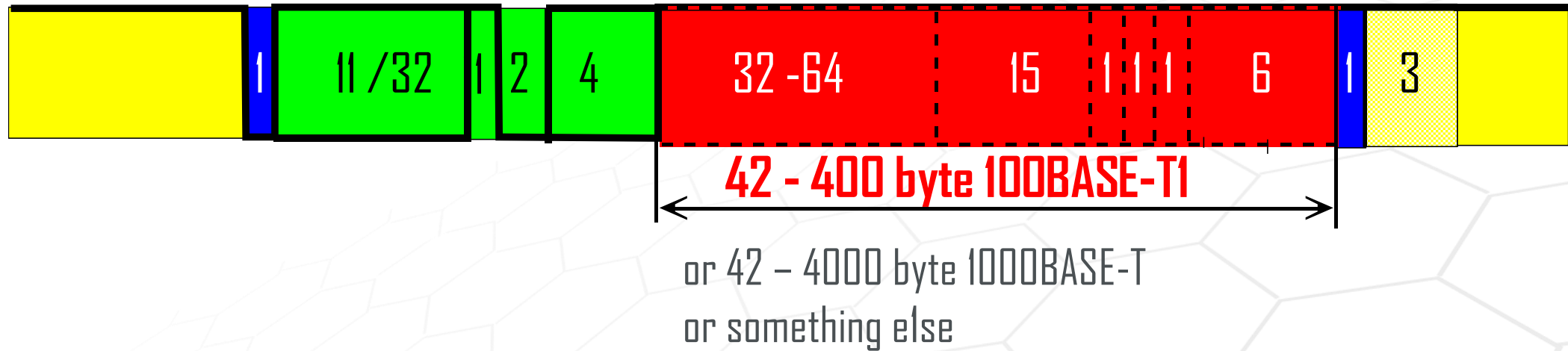
- Put all CAN nodes but one into Silent Mode
- The remaining CAN node switches to Ethernet mode on a CAN command
- **Communicate with the remaining node**
- Switch the remaining node into CAN mode by an Ethernet command
- Put the CAN nodes into Communication Mode



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

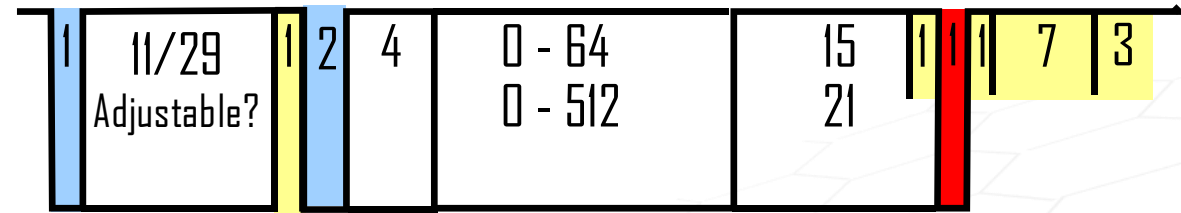
Dynamic change of protocol

- Certain CAN identifiers are assigned for Ethernet communication
- CAN nodes not having the CAN Ethernet ID set for reception turn into Silent Mode after the DLC field.
- All CAN nodes turn to Run Mode after EOF



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

A new network architecture

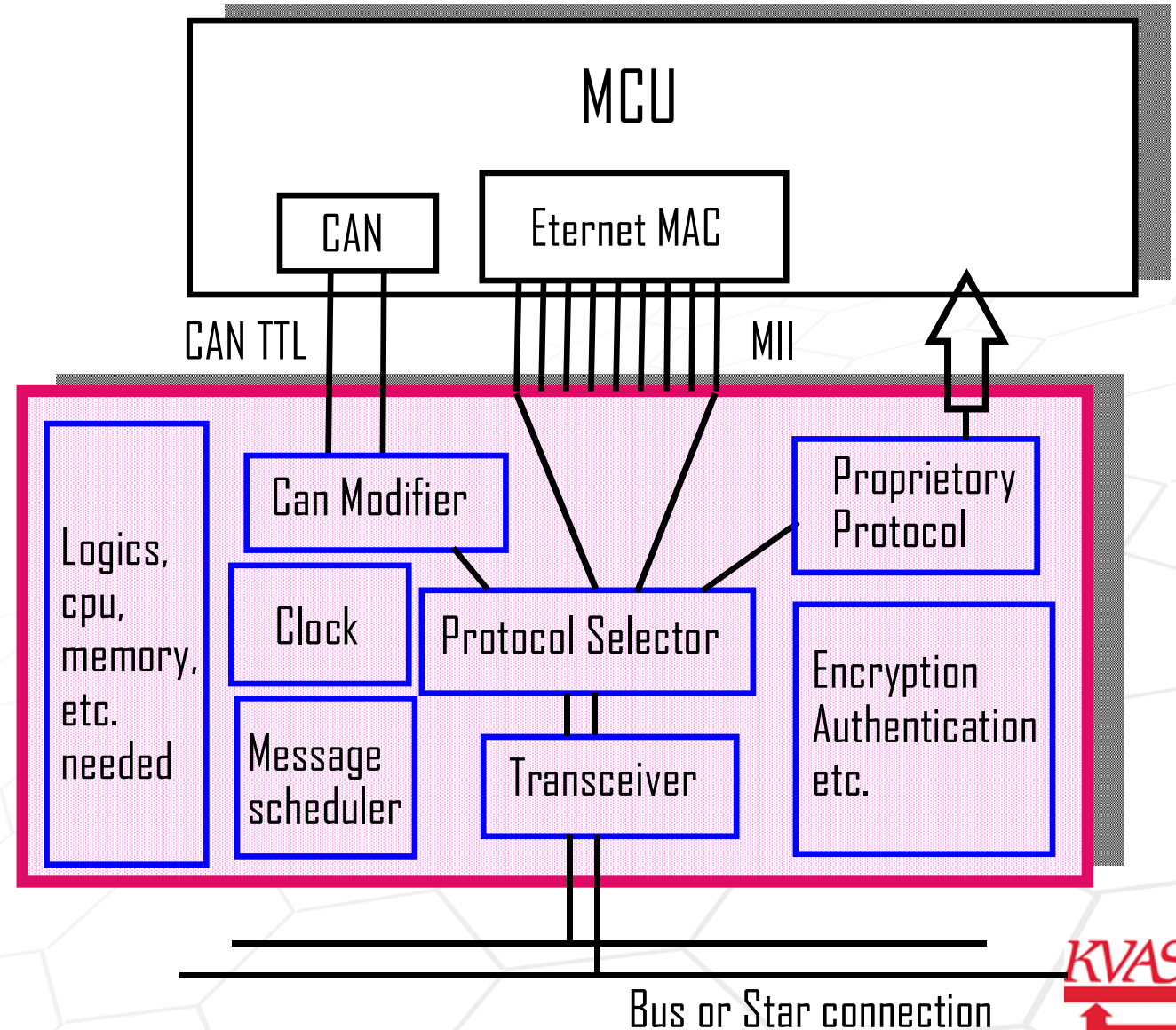


Classical CAN layer	For control purposes. Clear text. Classical CAN functionality
Embedded Protocol layer	Encryption, authentication, file transfers, etc.
CAN Check layer	Checking CAN frame by fix bits. Checking CAN bit quality by oversampling.
Bit coding layer	BTQ based. Splitting CAN and Embedded protocol
Modulation layer	Selection of appropriate modulation

Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

An intelligent transceiver

- Leave Classical CAN as is
 - Good enough for control purposes
 - Lots of tools
 - Trained engineers
- Separate systems design and MCU design
 - Message scheduling, encryption, etc. are system tasks
- Bring in new technology at a lower level
 - Fast changing
 - Only one component has to be updated



Some ideas for how we can enhance CAN by using the hidden bandwidth that modern technology releases

- Continuously check the signal quality
- Increasing the bit value accuracy
- Enhancing the clock synchronization within the system
- Encoding specific arbitration phase bits
 - Encoding stuff bits
 - Bit embedded signatures
 - Embedded encryption key
- A complete embedded protocol
- A star topology for CAN
- Combine Ethernet PHY and CAN in the same chip
- Dynamically changing the network between CAN, 10BASE T1, 100BASE T1 and other protocols