ARTIST Workshop at DATE'06
W4: "Design Issues in Distributed, Communication-Centric Systems"

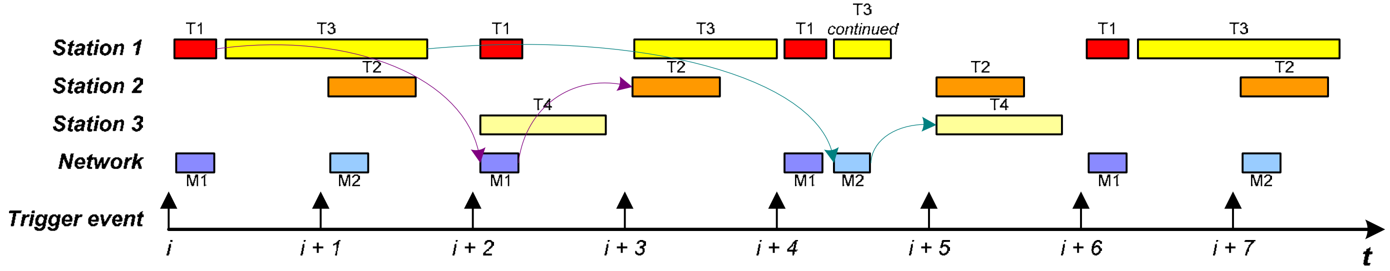
Safety-critical automotive systems:
New developments in CAN

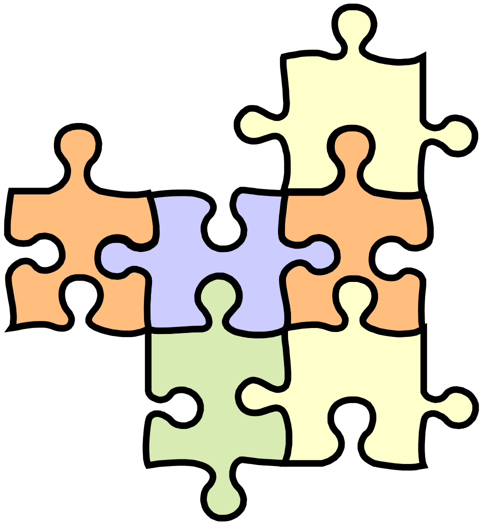
Luis Almeida
Electronic Systems Lab
http://www.ieeta.pt/lse
University of Aveiro
Portugal

Communication-centric design

❖**Integrated design of computations and communication**

- Communications establish **interdependencies** among tasks across the system
- Scheduling the whole system is a **multidimensional problem** that requires **joint scheduling** of tasks and communications
- Safety, reliability and consistency requirements further exacerbate the design problem
- Such integrated design **relies heavily on the network**
  - ***How long does communication take?***
  - ***When does communication take place?***
  - ***How reliable is the communication?***





Communication-centric design

- **A *good network* may provide properties that ease the integrated system design**
  - Bounded delays, isolated traffic classes, atomic broadcast...

- **What is a *good network*?**
  Application designers' perspective (speculative!)
  - A **simple and flexible** communication protocol that
    - ***provides basic communication services but allows building more complex services if required by the application***

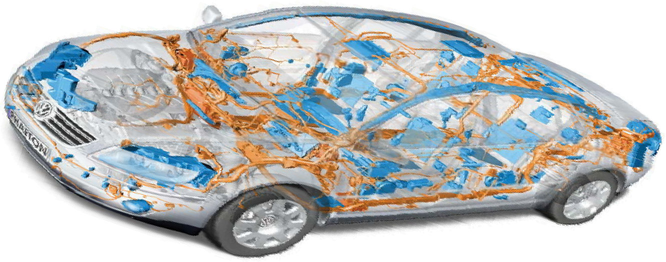
      But also

    - ***hides the idiossincracies of the low level communication while still meeting the time and reliability constraints***
    - ***and is cheap!***

Networks in the automotive domain

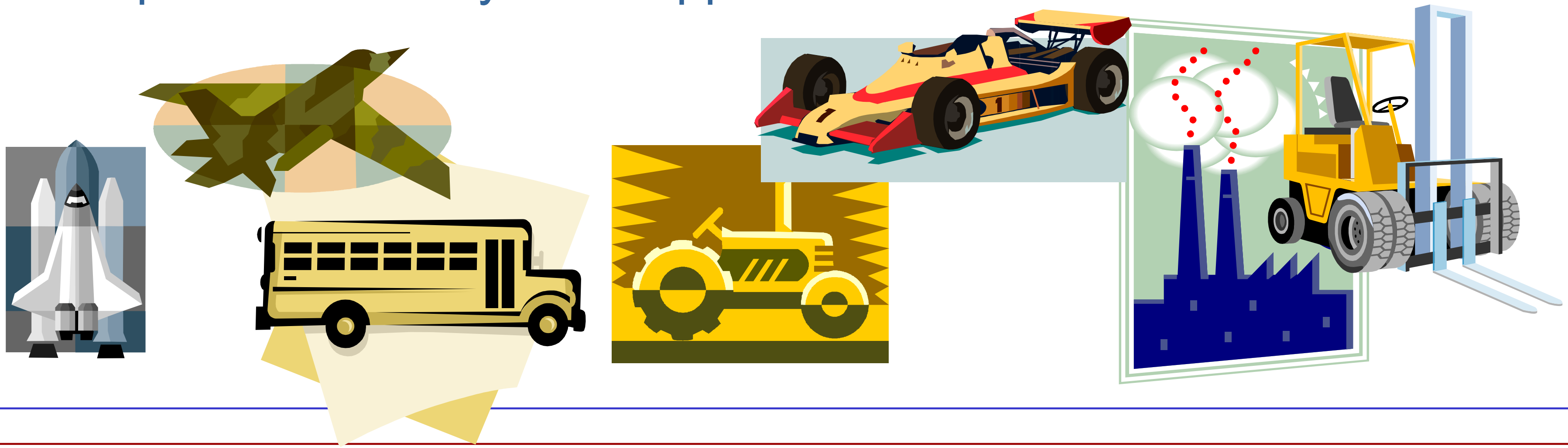
- **CAN**, **TTP/C, FlexRay, Byteflight**, LIN, MOST, Bluetooth...

Safety-critical subsystems



- **Among these protocols Controller Area Network (CAN) has particularly met those designer’s expectations up to a high degree**

and expanded to many other application domains!



Controller Area Network – a few facts

- **Pervasive use of CAN in many application domains**
  - Large installed base – over 10^9 controllers (2004)
  - Low failure rates
- **Very flexible protocol**
  - No constraints on the transmission instants, nor on the current set of exchanged messages
  - Uses only one global parameter (the message identifier)
  - Very easy to deploy
- **Good real-time behavior**
  - Establishes a global priority queue of messages
- **Robust physical layer**
- **Very good performance-cost ratio**

CAN – an on-going debate

❖But is CAN adequate for safety-critical applications?

- ➢ Already used in some safety-critical scopes...
  - ***Aerospace:*** *flap control by Hamilton Sundstrand (FAA certified)*
- ➢ **Many detractors:**
  - *CAN **inherent event-triggered** transmission mode does **not favor dependability***

  ☹

  - *It is **easier to detect errors** and **build fault-tolerant mechanisms** for **time-triggered** communication protocols (more a priori knowledge)*
- ➢ **And many supporters:**

  ☺

  - *CAN **inherent flexibility** may help **reacting to transient errors/overloads** while providing real-time behavior*

CAN dependability aspects

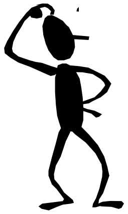
❖ **Faults in the channel**

- Many built-in mechanisms to **detect and signal errors**
- However an error in the last-but-one bit of a CAN frame may cause **inconsistent message duplicates** (IMD) or **omissions** (IMO).
  - *There are several solutions for this problem – **providing atomic broadcast / consensus** (Rufino, 1998; Kaiser, 1999; Proenza, 2000; Pinho, 2003; Lima, 2003).*
- **Experimental** data (Ferreira, 2004) indicates that the **probability of one IMO/h is less than 10^{-9}**
  - ***Possible use of CAN "as is" in safety-critical applications?***
  - ***Problems may arise when the automatic message retransmission upon error is time-limited** (TT protocols).*
- The **bus topology** presents several single points of failure
  - *Replicated bus? **Star topologies?***

CAN dependability aspects

- **Faults in the nodes**
  - CAN nodes may **fail uncontrollably**
    - *e.g., babbling idiot failure mode*
  - Using **bus guardians** grants fail-silence in the time domain, favoring the design of fault-tolerant mechanisms
    - ***No COTS*** *bus guardians but there are* ***several recent proposals*** *(Broster, 2003; Pimentel, 2005; Ferreira 2005)*
  - **Built-in error detection, masking and passivation** addresses **syntactic errors**, only, the latter being relatively **slow to act**
    - ***Fault-containment is essential*** *(substantial amount of work done)*
      - **Bus guardians, controlled retransmissions, star topology…**

So?



CAN – several complementary proposals

- **In recent years, several CAN-based protocols were presented to provide additional features**
  - **Better safety**
  - **Better fault-tolerance**
  - **Dependable flexibility**
  - **Better scheduling…**
- Generally, they provide **time-triggered transmission**
  - Facilitates error detection
- Some, require **fault-tolerant clock synchronization**
  - Many protocols available (Rodriguez-Navas, 2004)
  - Many COTS CAN controllers with HW support (*timestamps*)
  - Precision of 10μs is common
- **Dependability attributes** are taken into account

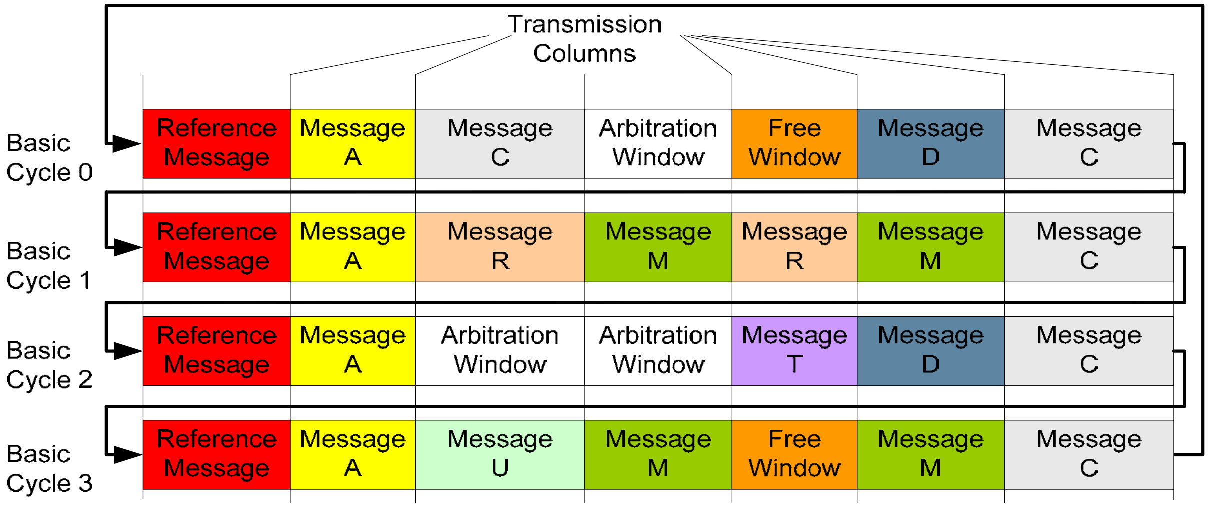
CAN – several complementary proposals

❖ **Some recent CAN-based protocols that provide additional safety features**

- **TTCAN - Time-Triggered CAN** (ISO11898-4, 2001)
  - *A few industrial applications (slow adoption...)*
- **FTT-CAN - Flexible Time-Triggered CAN** (Univ. Aveiro, 1999...)
  - *Applied to autonomous mobile robots and machine tools (Univ. Aveiro) as well as (on-going) steer-by-wire cars (Polyt. Coimbra,UFRGS Brazil)*
- **ServerCAN** (MRTC, 2002…)
- **TCAN - Timely CAN** (Univ. York, 2002...2004)
- **FlexCAN / SafeCAN** (Kettering Univ., 2004...)
  - *Applied to steer-by-wire car (Kettering Univ), steer-by-wire lift truck (Univ. Padova) and humanoid robot (Univ. Carlos III)*

Time-Triggered CAN – TTCAN

- **TDMA access** (requires specific controllers)
- **Prompt omission detection** (end of respective slot)
- **No automatic retransmissions** (single shot mode)
  - **Poor error recovery**
  - **High probability of IMO** (inconsistent omissions) (Broster, 2003) and poor safety support (Pimentel, 2006)
- **No bus-guardians considered**

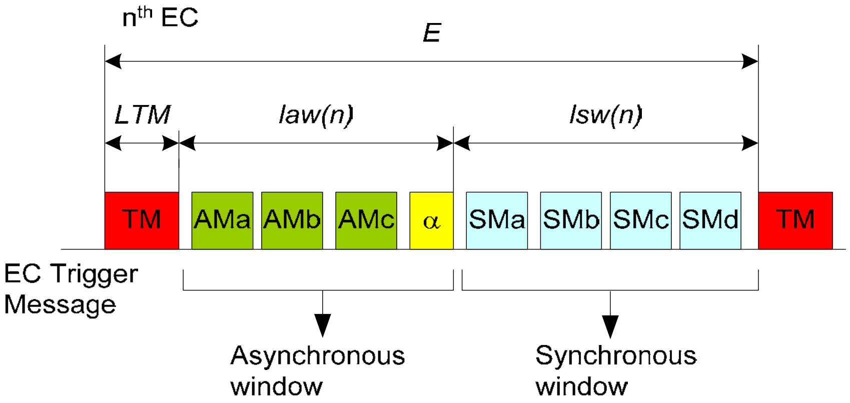


Transmission Columns

Basic Cycle 0	Reference Message	Message A	Message C	Arbitration Window	Free Window	Message D	Message C
Basic Cycle 1	Reference Message	Message A	Message R	Message M	Message R	Message M	Message C
Basic Cycle 2	Reference Message	Message A	Arbitration Window	Arbitration Window	Message T	Message D	Message C
Basic Cycle 3	Reference Message	Message A	Message U	Message M	Free Window	Message M	Message C

Flexible Time-Triggered CAN – FTT-CAN

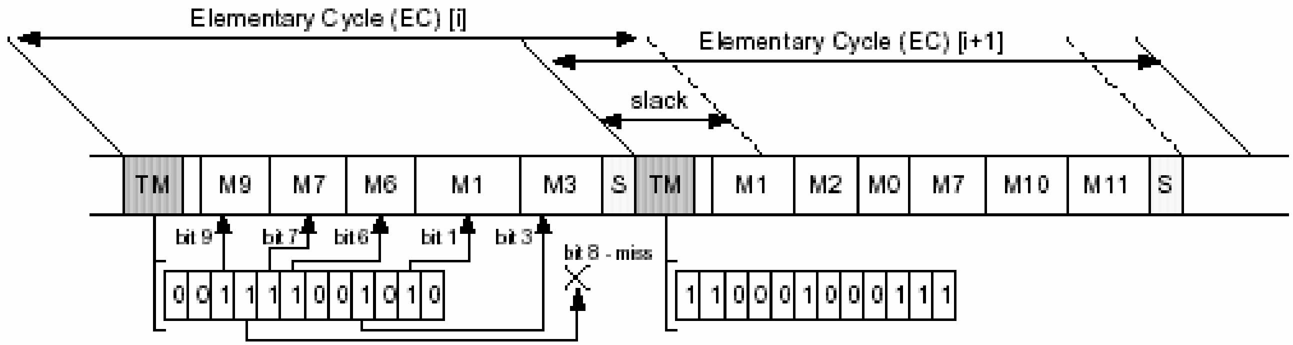
- **Master-slave** (optimized for low overhead)
  - Works with COTS controllers
- **Fast omission detection** (end of respective cycle)
- **Controlled retransmissions** (on-line rescheduling)
  - Medium probability of IMO
- **On-line scheduling, Rate adaptation, QoS management**
- **Specific bus-guardians designed**
- **Master replication:**
  - replacement, synchronization and consistent updates tested



ServerCAN

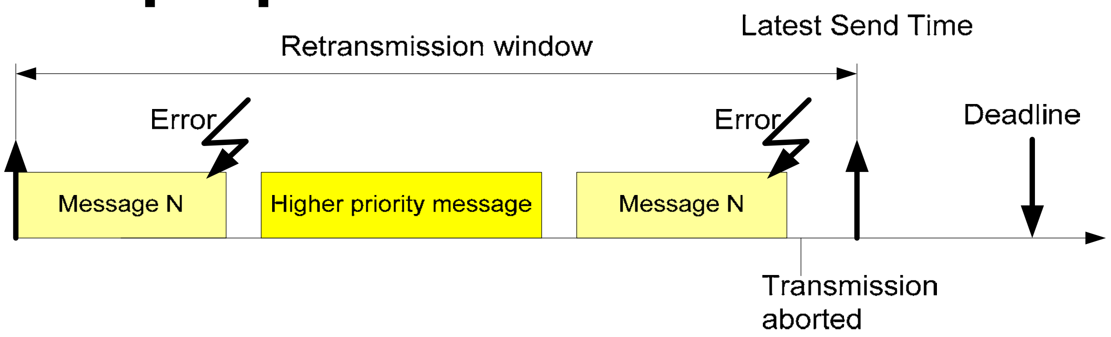
- **Master-slave** (optimized for low overhead)
  - **Works with COTS controllers**
- **Designed to improve scheduling (server-based)**
  - **Sporadic server, Constant Bandwidth Server, ...**
- **Omissions are part of scheduling**

  (i.e., no requests to be processed by the server)
- **On-line scheduling, improved isolation among flows**
- **Servers replication proposed**



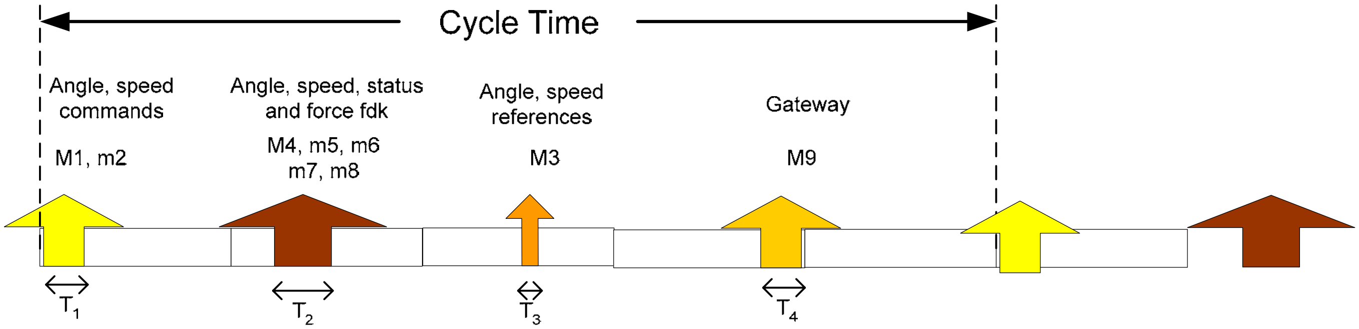
Timely CAN – TCAN

- **Predetermined Tx instants** (requires explicit clock sync.)
  - Effective tx can be delayed (e.g.,errors) until the **Latest Send Time**
  - LST = deadline – transmission time – clock uncertanties
  - Predetermined Tx and LST are known by all nodes
- **Slower omission detection** (by the respective deadline)
- **Bounded automatic retransmissions** (until the LST)
  - **Low probability of IMO**
  - **Best combination of reliability and timeliness**
- **Several bus-guardians proposed**



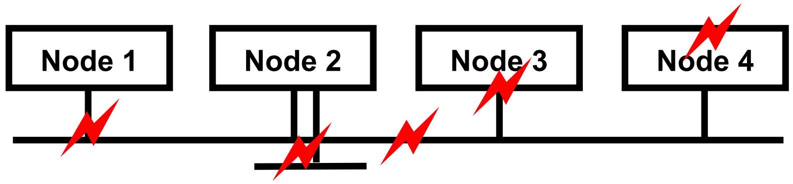
FlexCAN / SafeCAN

- **TT at the application level / ET in the network**
  - Cycle composed by sequence of windows
  - CAN native distributed medium access
- **Node and bus replication** (optional)
  - All nodes transmit on all channels they are connected to
- **Fast omission detection** (end of respective cycle)
- **Bounded automatic retransmissions**
  - Within each window (**low probability of IMO**)
- **Bus-guardians designed and tested**



CAN topology

- **BUT topology is also an issue!**
  - Original bus topology has several single points of failure
    - *grounded wires, loose connectors, faulty transceivers,...*
    - *errors propagate through the bus affecting the whole system*
  - Even replicated buses may suffer common-mode failures
    - *Both replicas must come together in the neighbourhood of each node*

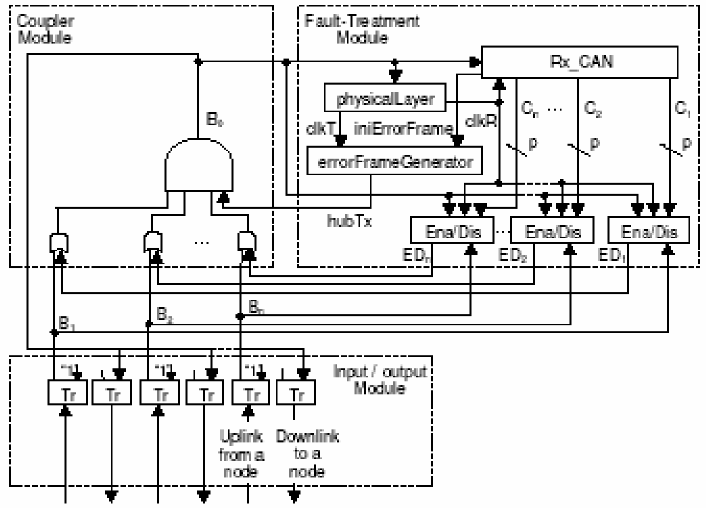


- **Solution!**
  - Follow the same trend has Ethernet, TTP/C and FlexRay
  - **Use a STAR topology with an active HUB**
    - ***CANcentrate*** *(Univ. Illes Baleares, Univ. Aveiro, 2004)*
    - ***ReCANcentrate*** *(Univ. Illes Baleares, Univ. Aveiro, 2005)*

CANcentrate

- First CAN-hub designed for error-confinement
  - Wired-AND of CAN bus replaced by logical AND
  - Uplinks separated from downlinks
  - Allows fast detection of several types of errors
    - *Link isolation when error threshold crossed*
      (latency to isolate stuck-at or bit-flipping faults: 73μs, 150...600μs)
    - *Automatic reintegration after error-free period*
    - (latency to reintegrate isolated links: 5.2ms)

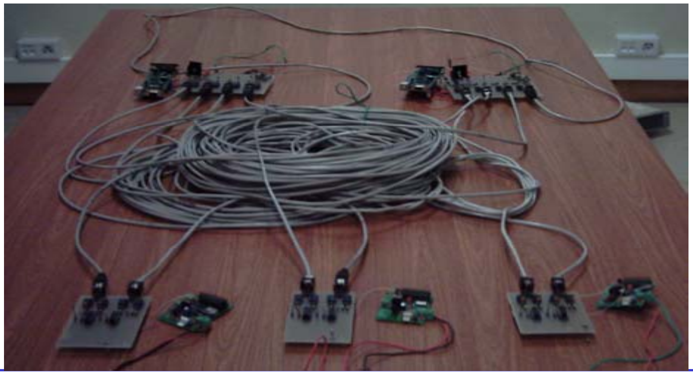
- Works with COTS CAN controllers and any existing application
  - It is just a replacement of the wiring

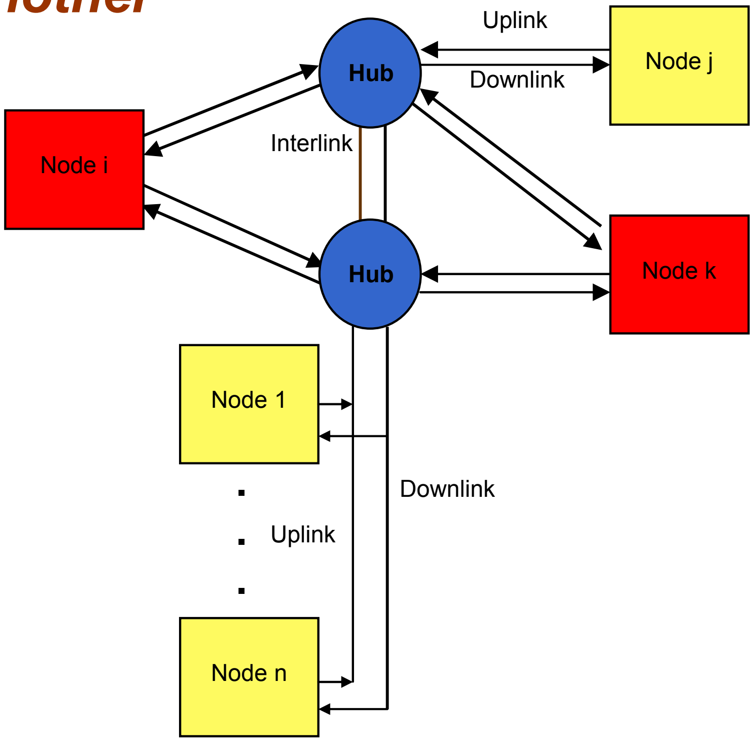


ReCANcentrate

- **First replicated CAN-hub architecture**
  - Targets very demanding safety requirements
  - Replicated hubs are synchronized bit-by-bit
  - Made by two interconnected CANcentrate hubs
    - *Hubs can isolate / reintegrate one another*

- **Supports mixed architectures with critical / non-critical nodes as well as bus segments**





(Re)CANcentrate

- Both **CANcentrate** and **ReCANcentrate** are **more expensive than a CAN** bus (due to wiring plus hubs)

but

- Still **potentially** **less expensive** than TTP/C or FlexRay while **potentially** **as dependable** (with ReCANcentrate)

and

- They can be **readily used** with **COTS CAN controllers** and in **current applications**

- **(Re)CANcentrate hub failure rate:** **~3...6 x10^{-7}**
  (similar to a CAN controller)

Further CAN limitations

- However, at least **one limitation remains**
  - **The limited bandwidth of CAN** (max. 1Mbit/s)
- **But how strong is this limitation?**
  - Most likely, the **car architecture** will continue being **multisegmented**
  - Typical requirements of the most demanding subsystems go up to a **few bytes** exchanged every **1 to 10ms**
    - *typical shared variables: temperature, speed, pressure, position...*

Conclusion

- ❖ CAN has been successfully used for **about 15 years** in many **different application domains**
- ❖ It is a **mature**, **well known**, **cheap** and **robust** technology
- ❖ It uses probably the **most bandwidth efficient technique** for **non-controlled bus access** with **small PDUs**
- ❖ It is very **flexible** and **simple** to use

However

- ❖ It presents **limitations** concerning
  - ***Safety aspects***
  - ***Bandwidth***

Conclusion

- ❖ Several protocols have been recently proposed that reduce the safety limitations
  - ***TTCAN, FTT-CAN, TCAN, FlexCAN***
- ❖ A new star topology has been proposed that eliminates the limitations of buses with respect to error confinement
  - ***CANcentrate*** *(simplex) and* ***ReCANcentrate*** *(replicated)*
- ❖ These solutions **provide CAN** with the required **safety level** for **critical automotive** applications
  - ➢ **With the potential for lower costs than other alternatives!**
- ❖ **Finally, there are many real-time analysis available for CAN to facilitate communication-centric designs**

References

J. Rufino, P. Verissimo, G. Arroz, C. Almeida, and L. Rodigues. Fault-tolerant broacast in CAN. FTCS 1998.

J. Kaiser, M. Livani. Achieving Fault-Tolerant Ordered Broadcasts in CAN. EDCC 1999.

J. Proenza, J. Miro-Julia. MajorCAN: A modification to the Controller Area Network to achieve Atomic Broadcast. IEEE Int. Workshop on Group Communication and Computations, 2000.

L. M. Pinho, F. Vasques. Reliable real-time communication in

can networks. IEEE Trans. Comput., 52(12):1594-1607, 2003.

G. M. A. Lima, A. Burns. A consensus protocol for CAN-based systems. RTSS 2003

J. Ferreira, A. Oliveira, P. Fonseca, J. Fonseca. An experiment to Assess Bit Error Rate in CAN. RTN 2004.

I. Broster, A. Burns. An Analyzable Bus-Guardian for Event-Triggered communication. RTSS 2003.

G.Buja, J.R.Pimentel and A.Zuccollo. Overcoming Babbling-Idiot Failures in the FlexCAN Architecture: A Simple Bus-Guardian. ETFA 2005

J.Ferreira, L. Almeida, J.Fonseca. Bus Guardians for CAN: a Taxonomy and a Comparative Study. WDAS 2005.

G. Rodriguez-Navas, J. Proenza. Clock Synchronization in CAN Distributed Embedded Systems. RTN 2004

I. Broster, A. Burns, G. R.-Navas. Comparing real-time communication under electromagnetic interference. ECRTS 2004.

J. R. Pimentel. Verification, Validation, and Certification of Safety-Critical Communication Systems. Kettering University Technical Report ECE-2006-01, 2006.

TTCAN:

ISO11898-4. Road vehicles - controller area network (CAN) - part 4: Time triggered communication. 2001.

FTT-CAN:

L. Almeida, P. Pedreiras, J. A. Fonseca. The FTT-CAN Protocol: Why and How. IEEE Tr. Industrial Electronics 49(6), 2002.

R. Marau, L. Almeida, J. A. Fonseca, J. Ferreira, V. Silva. Assessment of FTT-CAN master replication mechanisms for safety-critical applications. SAE World Congress 2006.

ServerCAN:

T. Nolte, M. Nolin, H. Hansson. Real-Time Server-Based Communication for CAN, IEEE Trans. Industrial Informatics 1(3):192-201, IEEE Industrial Electronics Society, 2005.

TCAN:

I Broster. Flexibility in Dependable Communication. PhD thesis, University of York, UK, 2003.

FlexCAN/SafeCAN:

J. R. Pimentel, J. A. Fonseca. FlexCAN: A Flexible Architecture for Highly Dependable Embedded Applications. RTN 2004.

(Re)CANcentrate:

M. Barranco, G. R.-Navas, J. Proenza, L. Almeida. CANcentrate: An active star topology for CAN networks. WFCS 2004.

J. Proenza, M. Barranco, L. Almeida. ReCANcentrate: A replicated star topology for CAN networks. ETFA 2005.

J. Proenza, M. Barranco, L. Almeida. Experimental assessment of ReCANcentrate, a replicated star topology for CAN. SAE World Congress 2006