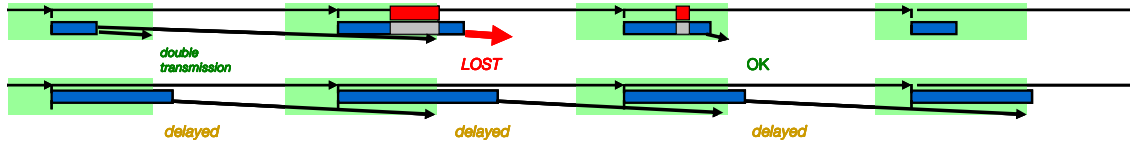


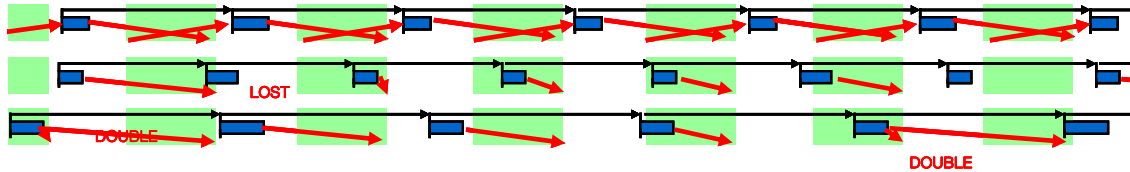
Ideal but challenging: Perfectly Synchronized FlexRay Nodes:



Synchronization Problems:



Asynchronous FlexRay:



Problem

FlexRay is often proposed as the simple way to eliminate timing problems by using deterministic frame timing, and the migration from CAN to FlexRay has started already.

However, FlexRay does not automatically solve all timing problems. Frames are only containers for the signals that establish the actual communication. Interactions of ECU functions with basic software lead to complex end-to-end communications exhibiting over- and under-sampling and signal jitter, thereby taking away much of FlexRay's determinism.

Automotive OEMs are faced with a variety of questions: Which slot assignment (and multiplexing strategy) shall be used? Which ECUs require synchronization while others can run asynchronously (and on cheaper hardware)? How to optimize the synchronization and how to communicate with the suppliers?

Answering these questions requires understanding the timing and performance impact of each decision. This is currently challenging the effective deployment of FlexRay.

Solution

SymTA/S determines - besides the scheduling of individual frames - also the interaction timing along signal paths incl. under- and over-sampling effects, data loss and dynamic task chaining between the sending and receiving ECUs and functions.

SymTA/S calculates maximum end-to-end signal delays and visualizes critical situations. The ECU timing can be included through detailed task scheduling models or in terms of abstract black-box ECU models. SymTA/S directly produces the corresponding requirements for the ECU suppliers in terms of function timing constraints.

SymTA/S lets network architects understand and explore different FlexRay configuration options and compares these to previous CAN-based solutions. The outcome is an optimized FlexRay configuration that meets the needs of the project.

SymTA/S ideally supports FlexRay network design, both from scratch as well as the migration from an existing CAN-based solution.