System behaviour in initialisation phase

MOST control channel modelling

By means of this article should be shown, that it is possible to model the behaviour of MOST control channel to investigate the performance effect of key parameters with a view to their optimisation. On this basis it's possible to configure some parameters locally using diagnostics in many of the Jaguar and Land Rover MOST modules. Thereon a test rig is to be constructed early in 2010.

From Ross McMurran and John Leslie

For the robustness of MOST based infotainment systems the initialisation is a key phase which is influenced by the status of the individual MOST nodes and the interactions between them on the control channel. This paper will present the results of work to model the behaviour of a MOST system during the initialisation phase and in particular that of the control channel.

Initialisation is the busiest period of MOST control channel communication, and becoming increasingly challenging to optimise as the number of customer features, and therefore number of MOST function blocks, increases. There are key customer requirements to satisfy, not only that the system starts up in a robust manner, but that certain infotainment and comfort features are available as soon as the vehicle is started.

Types of issues which could occur and must be mitigated in the system design include; excessive delay before audio sources are available, missing customer features, and delayed system startup. Traditionally, these types of error cases are driven out during system practical testing and signoff. This can be a long resource consuming process, even with recent advancements in automated testing, and will only become more complex as feature numbers increase.

Analysis of MOST initialisations has shown a number of interesting behaviours. During high initialisation bus loading, cases have been observed where multiple slave modules are sending control messages to the master module simultaneously, leading to a high number of not-acknowledged messages. During this period of intense retries it was possible for one slave module to monopolise the master module where only its messages were being received and acknowledged. This prompted questions regarding why the network services cycle time and the low level retry timing were set to the same value, and why the low level retry timing was identical in all slave modules. During the notification setting phase, all modules are trying to set notifications simultaneously. This causes high MOST control channel bus loading, therefore high numbers of not-acknowledged messages, therefore delays. Measures of network performance during startup include:

- Time from light on until the system is fully initialised – all notifications set and audio sources set up.
- Time from light on until a particular Customer feature is available.
- Number of MOST message low level retries used.

Adjustable parameters which affect initialisation performance include:

- Network Services cycle time (the rate at which the network services software task is called by the module's operating system).
- Number of low level retries.
- Low level retry timing / spacing.
- Use of network services mid level retries, and associated timing parameters.
- Use of application high level retries, and associated timing parameters.
- Receive and transmit message buffering strategy.
- MOST ring order.
- Scheduling of notification setting, and therefore order of feature availability.

These parameters are not fixed in the MOST specifications and are therefore to be specified by the vehicle manufacturer or module suppliers. Some of these parameters such as number and timing of low level retries have remained constant since the very first MOST system designs, and have been re-used on subsequent systems. For other parameters such as the network services cycle time, only simple guidance might be given to the module suppliers, such as the master node should be faster than the slave nodes.

Motivation for MOST control channel modelling

Optimising these parameters could be achieved by practical experimentation.

Clearly this could become a long and expensive process, as it would require support from several module suppliers. With CAN networks, Jaguar and Land Rover have been modelling network behaviour for many years to optimise performance. Similar techniques can be used for the MOST control channel. This would enable a greater understanding of the effects of these parameters, and the links between them.

The application of the model is to examine the network initialisation strategy and predict the effects of changing parameters, for example, retry wait times, over a wide variety of initialisation scenarios. For example, although forbidden by the MOST standard, could varying the low level retry spacing across the network improve performance in a system where only the master node is able to send broadcast messages, and group cast messages are not used.

The objectives within Jaguar Land Rover are to determine and specify requirements for the MOST system to guarantee predictable initialisation behaviour, and understand the effects of increasing customer feature on the MOST system during model year updates, in advance of building prototypes.

Approach to modelling

The requirement for the model was to have representative levels of MOST control channel traffic in terms of source, destination and relative timing of messages during the initialisation phase and representative behaviour of the channel in terms of factors such as timing, bandwidth and arbitration. Based on previous work by one of the authors in modelling networked distributed systems, the MOST system is modelled as a series of interlinked state machines. The model was constructed using Matlab Simulink & Stateflow – a top level view is shown in **figure 1**.

A key issue with any modelling is the level of abstraction, i.e. which aspects of the system must be left in to give representative behaviour in areas under consideration and which can be left out to make construction and operation of the model tractable. An approach of only modelling at particular layers in the OSI reference model was not possible as the behaviour is a factor of all layers, so the model included all layers with appropriate abstractions at each, e.g. physical layer modelled as Boolean value of light out. Other areas abstraction were in modelling the arbitration process as a separate central function and in simplifying the interaction at application level to generic types where the message sent dictated particular response behaviour by the receiver.

A model of a nine node system was constructed using an existing system as a reference for the communication patterns. At this stage no added delays at an application level are included, for

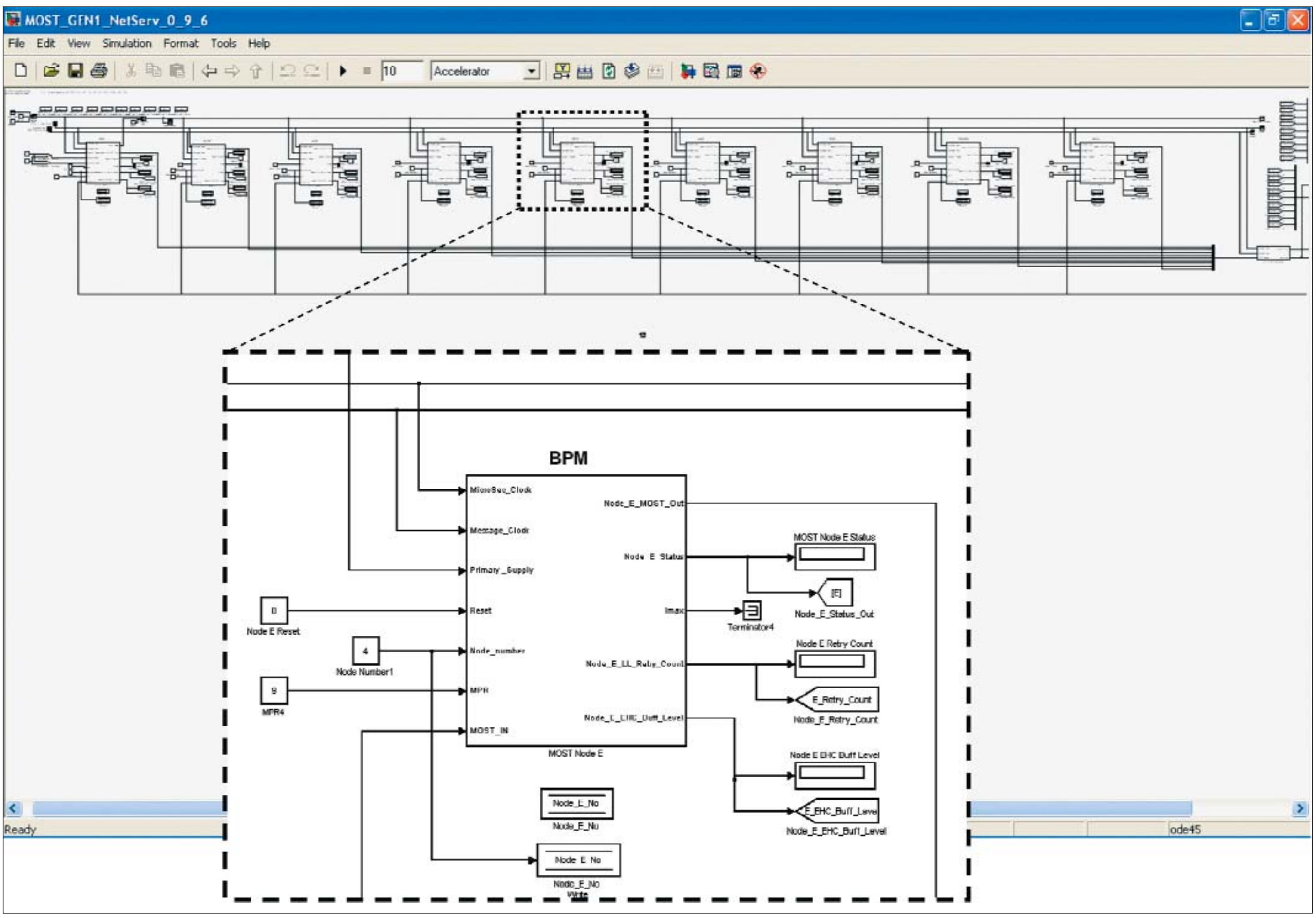


Figure 1. Top level view of model with enlarged view of three nodes.

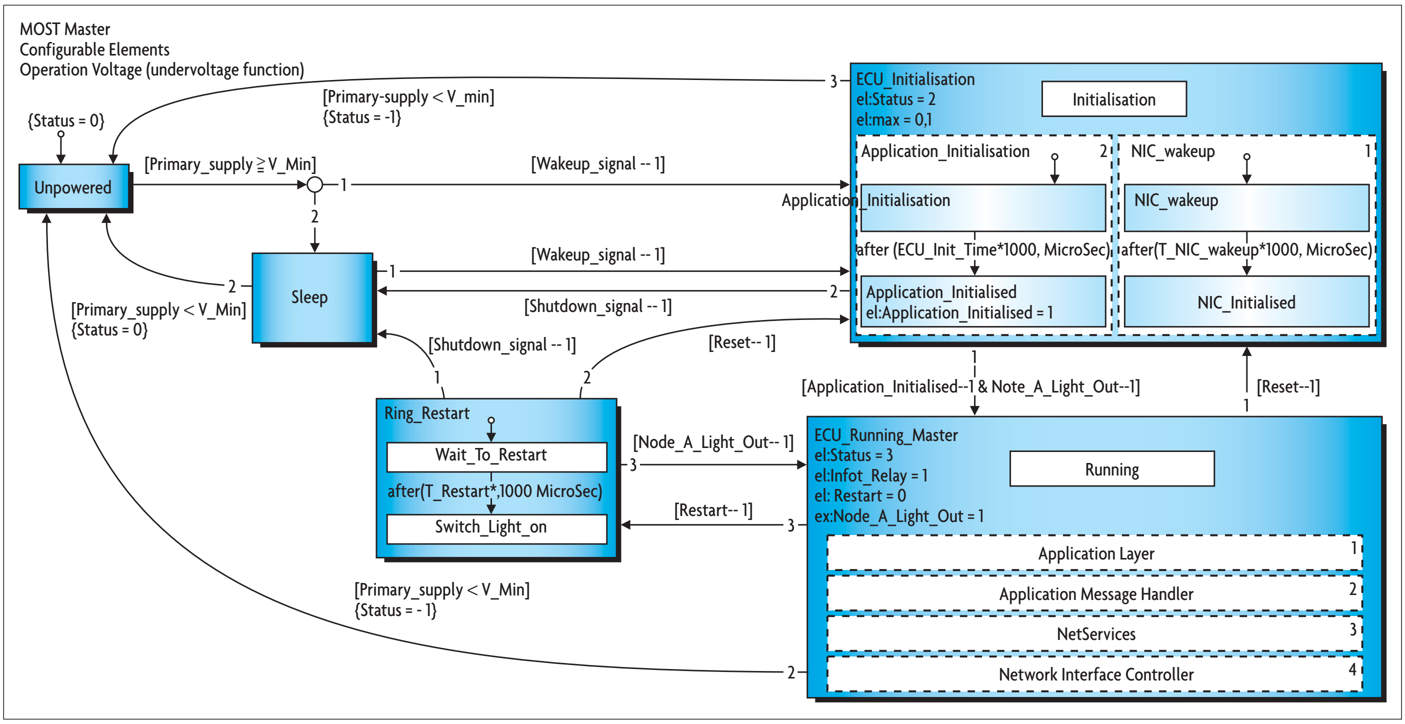


Figure 2. Overall structure of a node in the model.

example to space out notifications or response processing delay, to raise the level of intensity in the communications. The overall structure of a node is illustrated in **figure 2** and contains states for unpowered, sleep, initialisation and running. The top level of the running state shows the underlying layers which are the NIC (Network Interface Controller), Net Services, Application Message Handler, e.g. buffering in External Host Controller (EHC), and Application layers.

The individual nodes can be configured by parameters for factors such as ring order, MOST transceiver and application initialisation time, Low Level retry timing and number, NetServices cycle time and level of buffering in the EHC. **Figure 3** illustrates the output of the initialisation status from the model.

Initial results

For the initial testing of the model key parameters modified were NetServices timing (test 1), Low Level retry timing (test 2), Low Level retry number (test 3), Ring order (test 4) and a finally a

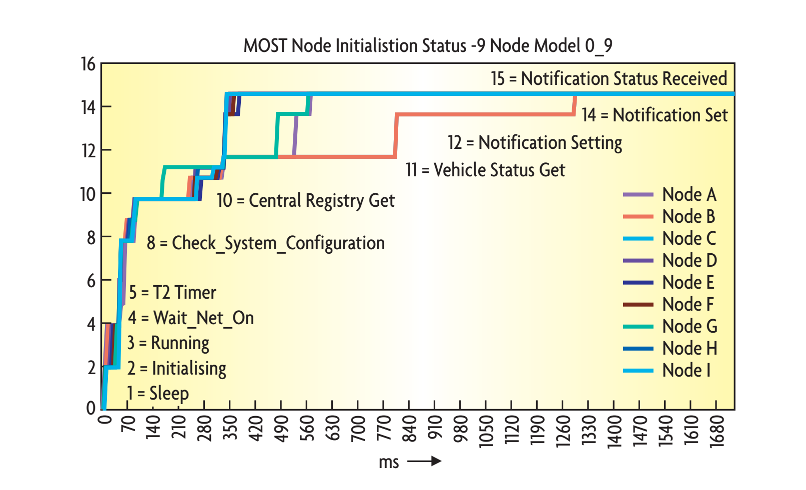


Figure 3. Output of node initialisation status from model.

utilisation of the control channel particularly in terms of received messages was moved from directly after the master to last in ring order (test 4). This was enough to prevent a high level retry which resulted in a reduction in initialisation time. While the HL retry distorted the impact of the effect it shows that ring order can have an effect.

Finally the results of the two hybrid scenarios whereby NetServices timing is reduced and retry timing differentiated indicate that if the NetServices times are sufficiently reduced the benefits from differentiating the retry spacing are negated. *bg*

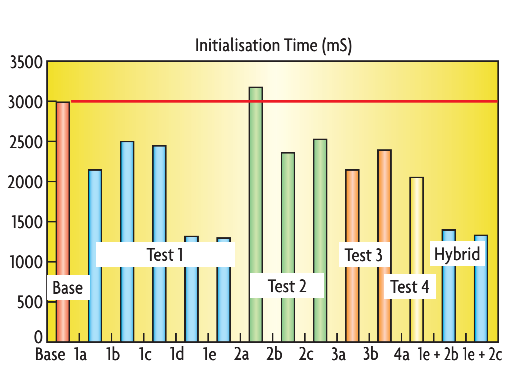
Hybrid of test 1 and 2. The results are shown in **figure 4**. Note the level of High Level retries reflects the deliberately greater level of intensity in the communications due to lack of delays in the model at an application level.

Unsurprisingly the highest impact parameter is NetServices timing (test 1). Within this there are effects in terms of the relative levels of NetServices timing between the nodes. For example reducing the NetServices cycle time of a single node did not produce level of benefit anticipated until the cycle time of key communication partners was similarly reduced. An area for further investigation is the whether it is best to synchronise times NetServices time or to have designed levels of variation.

The results also indicated there to be benefits in bending MOST rules by deliberately introducing a variation in low level retry between nodes, although this would only be possible if no other nodes than the master were allowed to send broadcast messages. While this was not successful in reducing initialisation timing when applied only to the reducing retry timing of master (test 2a), introducing a variation across all nodes with either a small or large variation did provide benefits through spacing out and desynchronising retries (test 2b and 2c).

Slightly increasing the retry number of the nodes (test 3) also had a beneficial effect but this may not be seen against a baseline scenario with no high-level retries. However it does suggest a role for model based experimentation for optimising the number of retries.

If all other factors are equal in the control channel message arbitration then according to the authors' observations, ring order is the deciding criteria. In the test a node with a heavy



	Initial'n Time	% NAKs	> 3 Retries	HL Retries
Base	2981	45 %	32	14
1a	2143	47 %	35	15
1b	2503	45 %	34	12
1c	2455	46 %	41	7
1d	1309	41 %	40	4
1e	1295	47 %	68	1
2a	3182	49 %	56	14
2b	2366	50 %	50	7
2c	2533	56 %	83	11
3a	2153	49 %	37	10
3b	2398	50 %	50	10
4a	2061	45 %	36	13
1e + 2b	1399	42 %	22	0
1e + 2c	1331	48 %	62	2

Figure 4. Initial test results from nine node MOST system.

Ross McMurran

leads an automotive electronics research group at the University of Warwick with interests in the robustness, validation, diagnostics, modelling and simulation of complex systems. He was formally senior manager for infotainment at Jaguar and Land Rover before moving to the university five years ago.

John Leslie

is the Technical Specialist for communication systems, and manager of the In-vehicle networks and software platform group within Jaguar and Land Rover. He has over nine years experience developing automotive networked systems, and is responsible for delivering all current and future Jaguar and Land Rover vehicle network architectures.