H/2 DLC Protocol in Hard Real Time Systems

H. Wijaya\textsuperscript{23}, N. Esseling\textsuperscript{2}, O. Klein\textsuperscript{2}, W. Zirwas\textsuperscript{1}, Hui Li\textsuperscript{1}, J. M. Eichinger\textsuperscript{1}, A. Vidal\textsuperscript{2}

\textsuperscript{1} = Siemens AG, \textsuperscript{2} = ComNets, RWTH Aachen, \textsuperscript{3} = AixCom GmbH

GERMANY
Initiated by Siemens AG

To observe and to develop Multi-Hop networks

The testbed development is shared among TU Hamburg-Harburg, IAF and TU-Dresden, AixCom and RWTH Aachen (ComNets)

To analyse and to optimise new algorithms e.g. for channel estimation, Radio Resource Management, Vertical and Horizontal Handover procedures

To develop a Multi-Hop capability for H/2 using “Forwarder” concepts
Testbed Architecture

- The H/2 Data Link Control Protocol (DLC) is formally specified using Specification and Description Language (SDL)
- Modem Interface Software (MIS) is responsible to handle the IRQs and Memory Mapping
- Rosa Box is responsible for Tx and Rx timing / synchronization purpose
- Time critical processes
Timing in Mobile Station MAC

<table>
<thead>
<tr>
<th>BCH</th>
<th>FCH</th>
<th>ACH</th>
<th>DL phase</th>
<th>UL phase</th>
<th>RCH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Frame structure: receive/ transmit time + modulation)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Rosa Box**
  - Length info
  - IE Block
  - Received LCHs + SCHs

- **MIS**

- **DLC**
  - Generation of UL phase + RCHs
  - Evaluation of DL phase

 SIEMENS

 RWTH ComNets
Timing in Access Point MAC

MAC-Frame

BCH | FCH | ACH | DL phase | UL phase | RCH

Received LCHs + SCHs
Frame structure: transmit line + modulation
Feedback
Received RCHs

Rosa Box

MIS

BCCH + FCCH
LCHs + SCHs
Received LCHs, SCHs, RCHs

DLC

Generation BCCH, FCCH and DL phase
Evaluation of UL phase RCHs
Time Critical DLC Processing

PCI Interface Controller

TX FIFO

TIME CRITICAL DLC PROCESSING

Control Engine

Modem TX Multiplexer

Modem Control Engine

Modem Rx Demultiplexer

H2 Modem Board

RF Frontend
H/2 DLC Specification in SDL
Problems in DLC Development

SDL is easy and good to use for specifying communication protocols but ...

- The execution time of „standard“ SDL specification is very slow
  - SDL is not best suited for Hard Real Time system !!
- It is very difficult to program an interface that works well with both SDL data structures and hardware data structures (MIS)
  - SDL cannot provide a good Software Abstraction Level for Modem Interface Software (MIS) !!
- The „standard“ SDL executable from SDT Telelogic Tau is a user application
  - SDL introduces extremely high delay and the concept of device driver programming is violated !!
DLC Development in RT System

Therefore

- Introduce C pointer into the SDL specification
- Introduce C data structures which work perfectly in both SDL and hardware interface environments (MIS)
- Linux supports only „Round Robin“ scheduling
  - SDL protocol has the same priority as the mouse protocol
- Real Time Linux supports „Priority“ Scheduling
  - Development of some new SDL Kernels which enable the SDL specification to run as Real Time Linux module
  - Compile and link the SDL specification and the SDL Kernel together as RT Linux Module and assign a high priority to it
• MIS intercepts IRQs from H/2 modem board and writes a message into the IPC queue.

• The SDL Kernel (SCT World) wakes up periodically to poll the ENV.

• The ENVironment (ENV) checks the IPC queue if there is a message.

• Then the ENV will contact SDL (Sys) after processing the message from IPC queue.
• MIS intercepts IRQs and triggers the SDL Kernel on by sending „ON“ signal through IPC and writes a message into the IPC queue

• The ENV checks the IPC queue if there is a message

• Then the ENV will contact SDL (Sys) after processing the message from IPC queue

• The ENV, triggers the SDT Kernel off by sending „OFF“ signal through IPC
The objectives of COVERAGE project
To observe and to develop Multi-Hop networks
Multi-Hop for H/2 using “Forwarder” concepts
Development of H/2 Data Link Control (DLC) in SDL
Development of some new Soft and Hard Real Time SDL Kernels
Development of Hard Real Time H/2 device driver