Verification of Mascara-Control

Natalia Sidorova       Martin Steffen

Dept. of Electrical Engineering
TU Eindhoven

Inst. für Informatik u. Prakt. Mathematik
CAU Kiel

June 2000
Vires-Review, Autrans
Overview

- Mascara-control
- Approach and tools
- Results
- Conclusion
Mascara

- **medium-access** for wireless ATM \(^1\)

- **Vires-Spec:** \(\geq 300\) pages (graphical)
  SDL-92 \([1]\)

- **large sub-entity:** Mascara-control
  - **task:** signalling/control
  - **interfaces** will all other entities

\(^1\)“**M**obile **A**ccess **S**cheme based on **C**ontention and **R**eservation for **A**TM”.
main/largest entities:

- **dynamic** control
  - set-up/tear down associations and connections$^2$
  - address management (conn./assoc.)
  - process creation

- **steady-state** control
  - monitor current association
  - monitor radio environment
  - collect info. about alternative associations

$^2$Hand-over is no MCL-service by itself, but composed of association and deassociation.
Main protocols involved

- association handling (set-up, tear-down)
- connection handling (set-up, tear-down)
- incommunicado: scanning the radio environment regularly
- I’m alive: sending invitations to overdue MTs
- alarm handling: fast reaction, if association breaks down
Approach

- **bottom-up** approach:
  1. MTC/steady-state control
  2. dynamic-control
  3. one-sided configuration for MCL
  4. 2-sided configuration for MCL

- **closing** manuall each configuration by enviroment

- **debugging** entities step-by-step
Abstractions

- underlying physical layer + data pump + WDLCs + CSR: lossy buffer\(^3\)

- abstracting away from generic Mascara control (= global initialization)

- abstracting largely radio control: non-determinism to represent decisions depending on physical environment

- general types of abstractions:
  - data-abstractions:
    * keeping the control-structure + reducing/removing the data-part
    * e.g.: two addresses
  - simple control-abstractions

\(^3\)for 2-sided configurations.
Model checking environment

- **ObjectGeode**: design of the SDL-specification, syntax checking, debugging using the ObjectGeode simulator facilities

- **sdl2if**: automatic translation of the SDL-specification into the Intermediate format (IF)

- **LIVE**: automatic transformation of the IF specification aimed at the reducing of the state space of the model

- **if2pml**: automatic translation of the obtained IF-specification into DT Promela/Promela

- **Spin/DTSpin**: model checking the model
Reachability checks

- easily done, by assertion violations

⇒ routine check-pointing crucial steps

- additional value: asserting desiredly reachable points as false

⇒ illustration of desired behaviour

⇒ comparison with Wand-MSCs [4]
Most common errors

- plain programming bugs
- values out of range
- race conditions (especially in the initialisation phases)
- ambiguous receiver
- unspecified reception
Time-dependent property: unique MAC addresses

- AP dynamic control: administers MT-addresses

Requirement:

"never the access point relinquishes an association before the mobile terminal does"

\[ \square (\varphi_{mt-\text{lost}} \rightarrow \varphi_{ap-\text{lost}}) \] (1)

- satisfaction depends on 4 timers + 4 program constants

<table>
<thead>
<tr>
<th>side</th>
<th>timer</th>
<th>program constant</th>
<th>process entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-side</td>
<td>( T_{iaa_poll} )</td>
<td>( \text{Max_Time_Periods} )</td>
<td>AIA</td>
</tr>
<tr>
<td></td>
<td>( T_{frame_start} )</td>
<td>( \text{IAA_Max} )</td>
<td>AIA</td>
</tr>
<tr>
<td>MT-side</td>
<td>( T_{GDP_period} )</td>
<td>( \text{Max_Cellerrors} )</td>
<td>GDP</td>
</tr>
<tr>
<td></td>
<td>( T_{rcm} )</td>
<td>( \text{Max_AP_Index} )</td>
<td>MTC</td>
</tr>
</tbody>
</table>
Unique MAC addresses (2)

“minimal time for AP to give up must exceed maximal time for MT to give up” \( \min(\tau_{AP}) > \max(\tau_{MT}) \),

- **Bounds of times**

\[
\begin{align*}
\tau_{AP} &\geq (\text{Max\_Time\_Periods} + 1) \times T_{iaa\_poll} + (\text{IAA\_Max} - 1) \times T_{frame\_start} \\
\tau_{MT} &\leq (\text{Max\_Cellerrors}) \times T_{GDP\_period} + (\text{Max\_AP\_Index} + 1) \times T_{rcm}
\end{align*}
\]

- extremal case (for \( \tau_{AP} \)): 1 MT

\( \Rightarrow \) configuration 1 AP, 1 MT
### Results (overview)

<table>
<thead>
<tr>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MT can go incommunicado</td>
</tr>
<tr>
<td>2. forward hand-over possible</td>
</tr>
<tr>
<td>3. incommunicado scenario</td>
</tr>
<tr>
<td>4. backward hand-over at MT</td>
</tr>
</tbody>
</table>

### Errors found

<table>
<thead>
<tr>
<th>Errors found</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. negative number of associated MT’s</td>
</tr>
<tr>
<td>2. twice “start-of-tip” without end-of-tip in between</td>
</tr>
<tr>
<td>3. twice “end-of-tip” without start-of-tip in between → illegal termination</td>
</tr>
<tr>
<td>4. incommunicado becomes impossible</td>
</tr>
<tr>
<td>5. illegal termination</td>
</tr>
<tr>
<td>6. infinite undetected loop in backward hand-over</td>
</tr>
</tbody>
</table>

### Verified properties

<table>
<thead>
<tr>
<th>Verified properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. no illegal termination</td>
</tr>
<tr>
<td>2. toggle-array chooses correct branches in AT1 (start-tip)</td>
</tr>
<tr>
<td>3. toggle-array chooses correct branches in AT1 (end-of-tip)</td>
</tr>
</tbody>
</table>

### Timed properties (positive and negative results)

<table>
<thead>
<tr>
<th>Timed properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. permanent backward hand-over (im)possible</td>
</tr>
<tr>
<td>2. unique/ambiguous MAC-addresses</td>
</tr>
</tbody>
</table>
Conclusions

- **Debugging** of one large part of Mascara using **model-checking**

- good **touch-stone** of the **Vires** tool-set

- experiments provided valuable **feed-back** to the tool-development

- **simple** abstractions already get you far
Lessons learned

- more effort on specification

- tool integration

- make easy things easy

⇒ support for routine task, like
  - diagnostics
  - automatic closing of the model
  - whole-sale chaotic abstraction of complete entities + necessery (small, but many) interface adaptations
Literatur


