The following full text is a publisher’s version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/34748

Please be advised that this information was generated on 2017-12-30 and may be subject to change.
Revealed incorrectness upon receiving payload in 12-SENT or 12BIS-SENT.
Revealed possible deadlock with optional retransmitting I2 / I2bis messages.
Clarified confusion about responder nonce.
Revealed several other ambiguities, omissions and inconsistencies.
Acknowledged by SHIM6 draft authors. Will be incorporated in new IETF proposal.

UPPAAAL: Improve model to verify on scale.
Add failure detection and exploration. Extend UPPAAAL verifier language.
Indicate model state space.

Background

Multihoming
- A technique to increase the reliability of a network connection.
- Features redundancy, load sharing, performance and policy.
- Current multihoming practices (IPv4) impose a threat on address and routing scalability.
- SHIM6 is a proposal by the IETF to provide multihoming that solve these issues.
- No formal methods have been applied to the draft specification.

Aim: improve the quality of the specification by applying formal methods.

How SHIM6 works
IP roles SHIM6 splits the two semantics of an IP address (end point identifier and locator role).
Initial contact Normal data communication between end point identifiers, no SHIM6 needed.
Context Establishment Communication to exchange multihoming information. Data communication remains normal.
Failure detection Messages are transmitted to detect a link failure.
Locator pair exploration In case of a link failure, a new locator needs to be selected. Locators are mapped back at the host to the end point identifier. Transport session remains stable. Communication resumes with SHIM6 data packets that provide mapping information.

Formalization and Verification

UPPAAAL
An integrated tool environment for modelling, validation and verification of real-time systems modelled as networks of timed automata, extended with data types.

Properties:
- A[] not deadlock
- exists(h1:HostType) exists(h2:HostType) (h1 != h2 and heuristics[h1][h2]) --> forall(h3:HostType) forall(h4:HostType) (h3 != h4 imply Context(h3, h4).established)

Results

Future Work

Further Information

SHIM6:
- http://www.shim6.org

UPPAAAL:
- http://www.uppaal.com