World Class Standards

From test design to validation

(with the example of the IPv6 test bed)

4th e-Infrastructure Concertation
Sophia Antipolis, 5/6 Dec 2007

Sebastian Müller

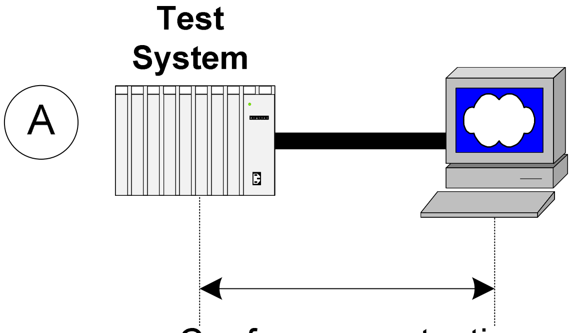
Centre for Testing and Interoperability

ETSI Standardized Testing

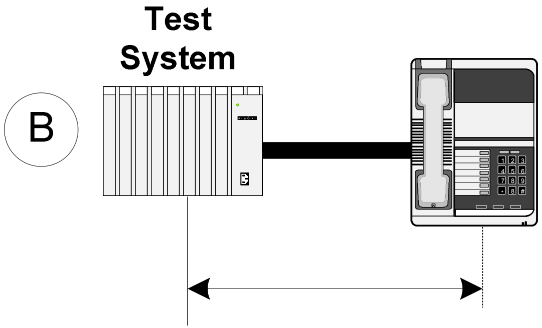
- ❑ Two complementary forms of testing
  - ➢ Conformance testing
  - ➢ Interoperability testing
- ❑ ETSI Test Specification Development Process
  - ➢ Requirement Catalogue
  - ➢ Test Purposes
  - ➢ Conformance tests/Interoperability tests
  - ➢ Test Validation
- ❑ ETSI techniques
  - ➢ TPLan
  - ➢ TTCN-3

Conformance and Interoperability Testing

Product testing against the standard

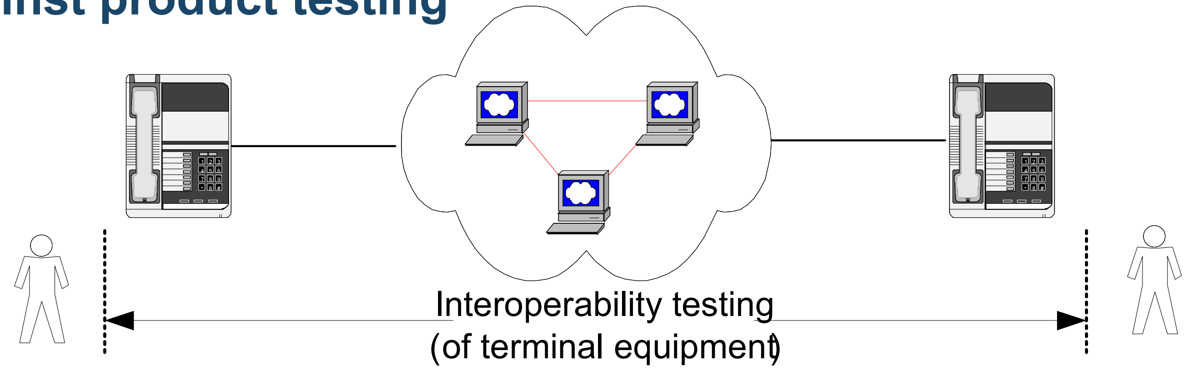


Conformance testing
(of a network element)



Conformance testing
(of terminal equipment)

Product against product testing



Interoperability testing
(of terminal equipment)

Characteristics of Conformance Testing

- ❑ Tests IUT against requirements specified either in a base specification or profile (standard)
  - Gives a high-level of confidence that key components of a device or system are working as they were specified and designed to do
  - Usually limited to one requirement per test
- ❑ High IUT control and observability
  - Can explicitly test error behaviour
  - Can provoke and test non-normal (but legitimate) scenarios
  - Can be extended to include robustness tests
- ❑ Test execution can be automated and repeated
  - Requires a test system development (and executable test cases)
  - Can be expensive (e.g., radio-based test system)
  - Tests under ideal conditions
- ❑ Tests are thorough and accurate but limited in scope
  - At level of detailed protocol messages, service primitives, or procedure calls

Limitations of Conformance Testing

- ❑ **Does not prove end-to-end functionality (interoperability) between communicating systems**
  - ➢ Conformance tested implementations may still not interoperate

    This is often a specification problem rather than a testing problem!
    Need for minimum requirements coverage or profiles
- ❑ **Tests individual system components**
  - ➢ But a system is often greater than the sum of its parts!
  - ➢ Does not test the user's 'perception' of the system
- ❑ **Standardised conformance tests do not include proprietary 'aspects'**
  - ➢ Difficult to test via proprietary interfaces, e.g., user APIs
  - ➢ Can however be done by a manufacturer with own conformance tests for proprietary requirements

Characteristics of Interoperability Testing

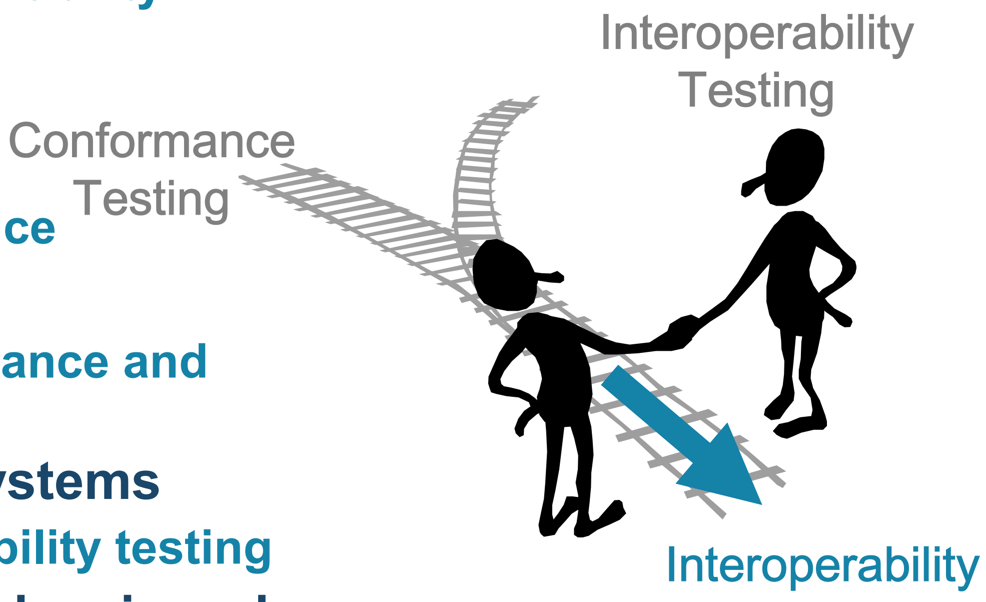
- ❑ It is system testing
  - ➢ Tests a complete device or a collection of devices
- ❑ Shows that (two or more) devices interoperate
  - ➢ But within a limited scenario
- ❑ Tests at a 'high' level (perception of users)
  - ➢ Tests the 'whole', not the parts or layers
    - E.g., protocol stacks integrated with applications
  - ➢ Tests functionality
    - Shows function is accomplished (but not how)
- ❑ Does not usually involve test systems
  - ➢ Uses existing interfaces (standard/proprietary)
- ❑ Interoperability testing looks at end-to end functionality
  - ➢ Less thorough but wide in scope
  - ➢ Gives a high-level of confidence that devices (or components in a system) will interoperate with other devices (components)

Limitations of Interoperability Testing

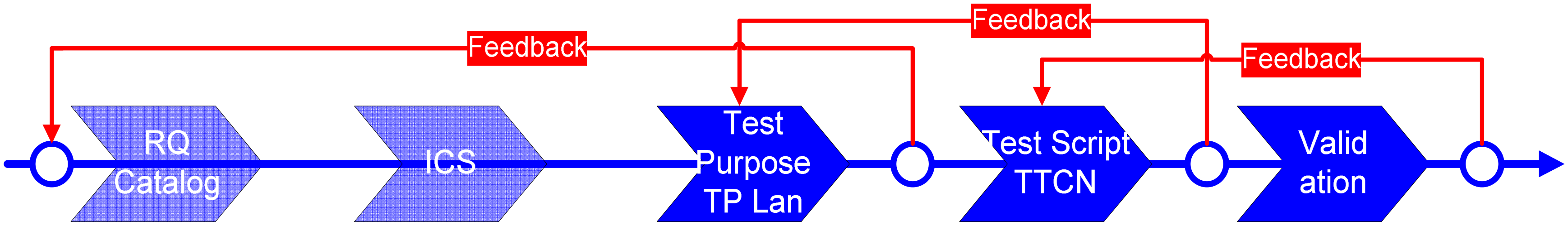
- ❑ Does not prove that a device is conformant
  - ➢ Interoperable devices may still interoperate even though they are non-conformant to a base specification or standard
- ❑ Cannot explicitly test error behaviour or unusual scenarios
  - ➢ Invalid conditions may need to be forced (lack of controllability)
  - ➢ Has limited coverage (does not fully exercise the device)
- ❑ Usually performed manually
  - ➢ Difficult to automate, tests use more time to be prepared than executed.
- ❑ Does not prove 100% interoperability with other implementations with which no testing has been done
  - ➢ ‘A’ may interoperate with ‘B‘ and ‘B’ may interoperate with ‘C’. But it doesn’t necessarily follow that ‘A’ will interoperate with ‘C’.

Conformance and Interoperability Testing are Complementary

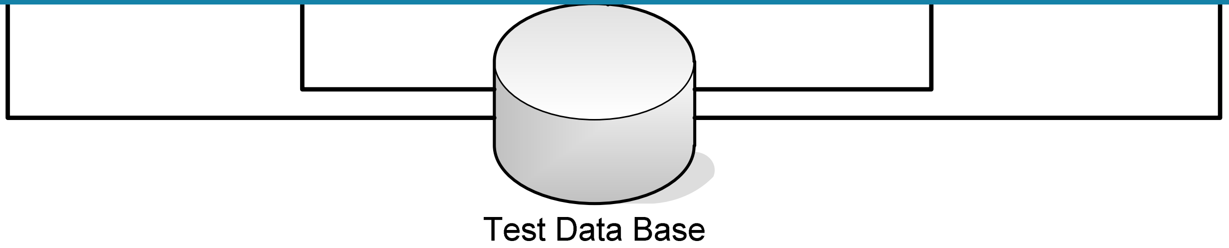
- ❑ ETSI experience
  - ➢ As you move up a system stack the emphasis should change from conformance to interoperability testing
- ❑ Lower layer protocols, infrastructure
  - ➢ Emphasis on conformance
- ❑ Middleware, enablers
  - ➢ Combination of conformance and interoperability testing
- ❑ Services, applications, systems
  - ➢ Emphasis on interoperability testing
- ❑ Conformance testing can be viewed as a pre-requisite to interoperability testing



ETSI Test Specification Development Process



Testing provides vital feedback into the base standard

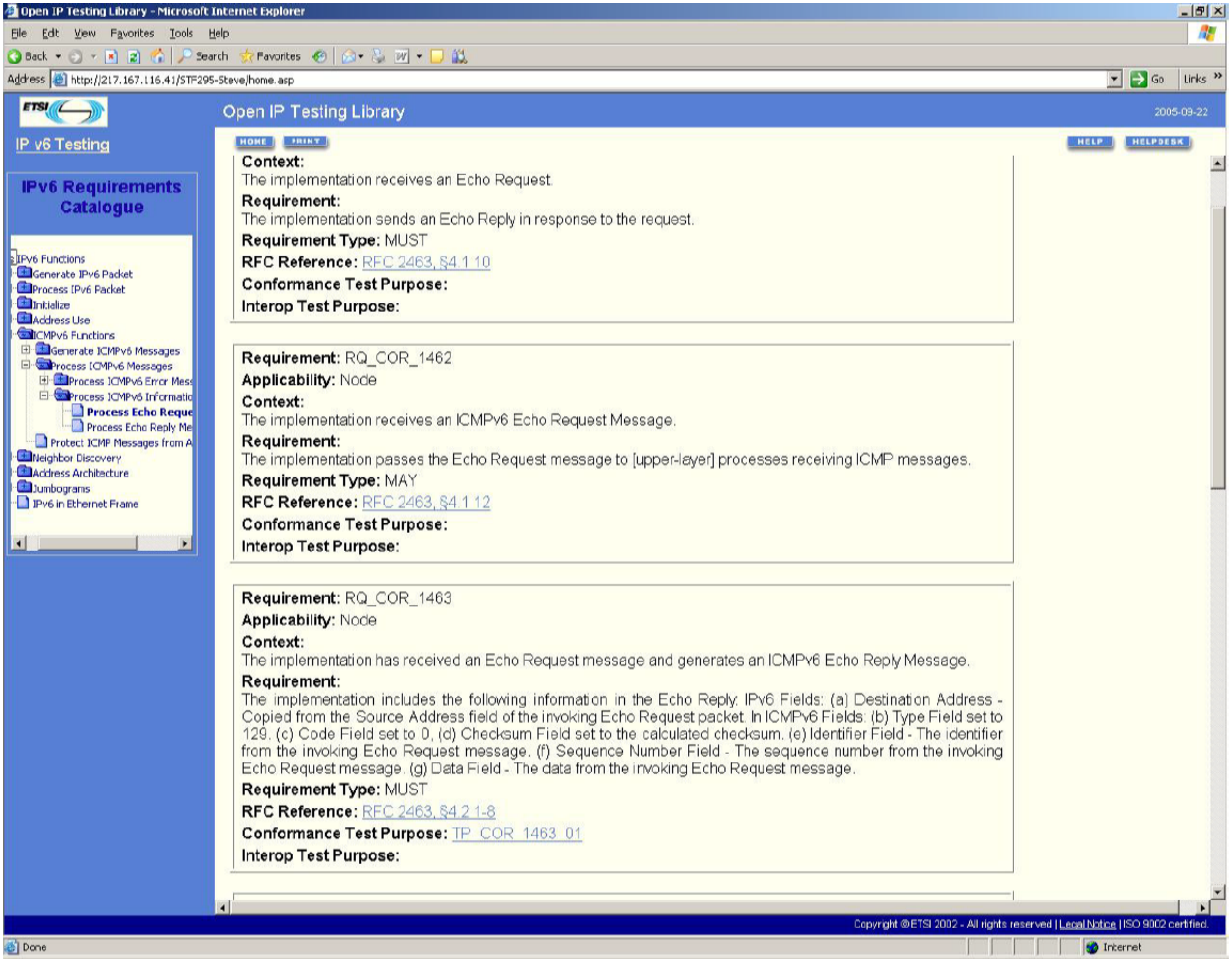


Web interface, browsing by function, user-defined search and filter, traceability, document generation, …!

Requirements Catalogue

- Each requirement is identified, extracted, and catalogued as follows (example from IPv6):
  - Requirement type:
    - Mandatory (MUST, MUST NOT, SHALL, SHALL NOT)
    - Recommended (SHOULD, SHOULD NOT)
    - Optional (MAY, MAY NOT, COULD)
  - Requirement role (tested entity):
    - E.g., Host, Router, Node (Host or Router)
  - Requirement context
  - Requirement text
  - Exact base document citation and reference
  - Functional grouping
    - E.g., Process Fragmented packet, Generate ICMPv6 Error Type etc.

The Requirements Catalog Online



Context:
The implementation receives an Echo Request.

Requirement:
The implementation sends an Echo Reply in response to the request.

Requirement Type: MUST

RFC Reference: RFC 2463, §4.1.10

Conformance Test Purpose:

Interop Test Purpose:

Requirement: RQ_COR_1462

Applicability: Node

Context:
The implementation receives an ICMPv6 Echo Request Message.

Requirement:
The implementation passes the Echo Request message to [upper-layer] processes receiving ICMP messages.

Requirement Type: MAY

RFC Reference: RFC 2463, §4.1.12

Conformance Test Purpose:

Interop Test Purpose:

Requirement: RQ_COR_1463

Applicability: Node

Context:
The implementation has received an Echo Request message and generates an ICMPv6 Echo Reply Message.

Requirement:
The implementation includes the following information in the Echo Reply: IPv6 Fields: (a) Destination Address - Copied from the Source Address field of the invoking Echo Request packet. In ICMPv6 Fields: (b) Type Field set to 129. (c) Code Field set to 0, (d) Checksum Field set to the calculated checksum. (e) Identifier Field - The identifier from the invoking Echo Request message. (f) Sequence Number Field - The sequence number from the invoking Echo Request message. (g) Data Field - The data from the invoking Echo Request message.

Requirement Type: MUST

RFC Reference: RFC 2463, §4.2.1-8

Conformance Test Purpose: TP_COR_1463_01

Interop Test Purpose:

Test Purposes

- ❑ Precise descriptions of the purpose of a test in relation to a particular (base standard) requirement
  - ➢ Meaningful to a larger audience than technical experts
- ❑ Define the functionality being tested, i.e., WHAT?
  - ➢ Should be formulated using a similar level of information content and wording as the corresponding requirement description
  - ➢ Should only mention directly relevant aspects of interactions but not describe them in detail
  - ➢ Reference a test configuration (i.e., architecture)
  - ➢ Identify one or more test purposes per requirement
- ❑ Specified at ETSI in
  - ➢ Natural language, or
  - ➢ ETSI's Test Purpose notation (TPLan): www.tplan.info
  - ➢ But definitely not coded!

TPLan Example from Digital Public Mobile Radio

```
TP id      : TP_PMR_0406_01
summary    : 'Header frame acknowledges connect request'
TP type    : conformance
RQ ref     : RQ_001_0406 -- Catalogue Identifier
IUT Role   : CSF   -- Configured Service Function (CSF)
config ref: CF_dPMR_CSF_01 -- CSF Implementation Under
                           -- Test (IUT) and TESTER

with   { IUT in standby }

ensure that {
  when { IUT receives a Connection_Request }
  then { IUT sends an Acknowledgement_Frame }
}
```

Conformance Test Suite

- ❑ A collection of detailed test cases or scripts that implement test purposes
  - ➢ Can be compiled and executed, e.g., when using TTCN-3
- ❑ Specifies HOW to test
  - ➢ Implements also handling of, e.g., unexpected or background behaviour, or returning the SUT to the initial state after a test
- ❑ Composition of test cases
  - ➢ Each individual test has a preamble, test body (i.e., implementation of the Test Purpose), and postamble
  - ➢ Test components may be used clearly separate interactions with different logical IUT interfaces during a test
- ❑ Assigns test verdicts
- ❑ Should be parameterizable at run-time, e.g., with SUT address
- ❑ A test suite is not a test system
  - ➢ Test focus on IUT behaviour
  - ➢ Test system handles also message encoding and transport

What is TTCN-3?

www.ttcn-3.org

- Testing and Test Control Notation Version 3
- Internationally standardized language developed specifically for executable test specification
- Look and feel of a regular programming language
- Good tool support (today 6 commercial tools available)

Advantages of Using TTCN-3

- Access to standardized test suites
  - UMTS by 3GPP
  - WiMAX, SIP, IPv6 and more by ETSI
- One language enables reusage of test elements
- Cost reduction in testing process and higher quality of products
- Is independent of a test execution environment
  - Standardized runtime interfaces (TRI/TCI)
- Build-in test automation and system specification features
- Further development and maintenance of TTCN-3

Interoperability Test Description

- ❑ Specifies detailed steps to be followed to achieve stated test purpose, i.e., HOW?
- ❑ State steps clearly and unambiguously without unreasonable restrictions on actual method:
  - ➢ Example:
    - Answer incoming call

      NOT
    - Pick up telephone handset
- ❑ Written in a structured and tabulated natural language so tests can be performed manually
- ❑ At least in theory it is possible to increase automation with test scripts, e.g., written in TTCN-3
  - ➢ Proprietary nature of user APIs makes this in practice difficult
  - ➢ Requires additional adapter development from equipment vendors

Validation of Tests

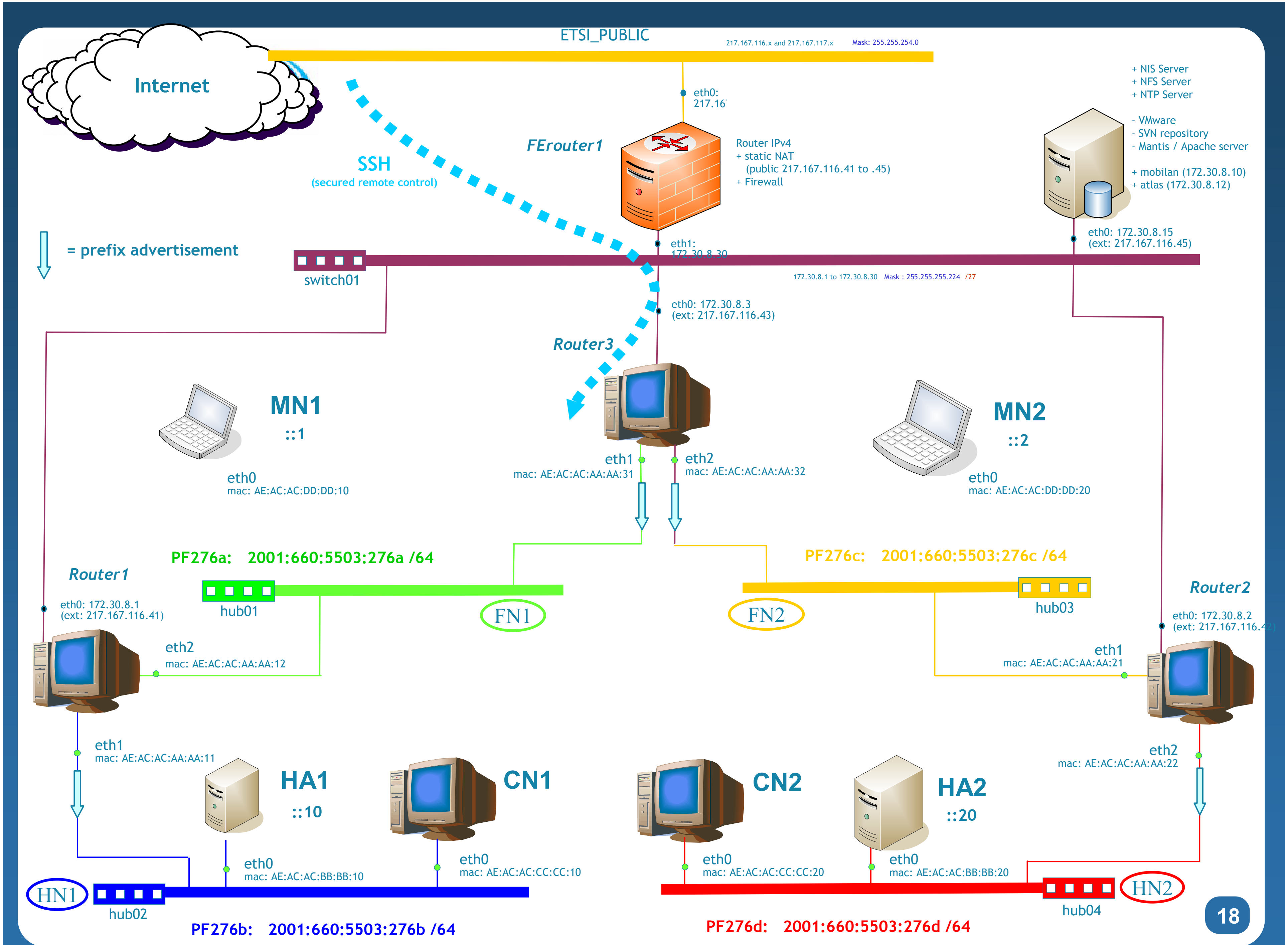
- Typical test suite validation requires that
  - Tests compile with several TTCN-3 tools
  - Tests are executed on one or more platforms against various implementations of the standard (usually by ETSI members)
- Requires very close co-operation with test tool suppliers, test platform providers, equipment manufacturers etc.

Test Tool Supplier: Testing Tech

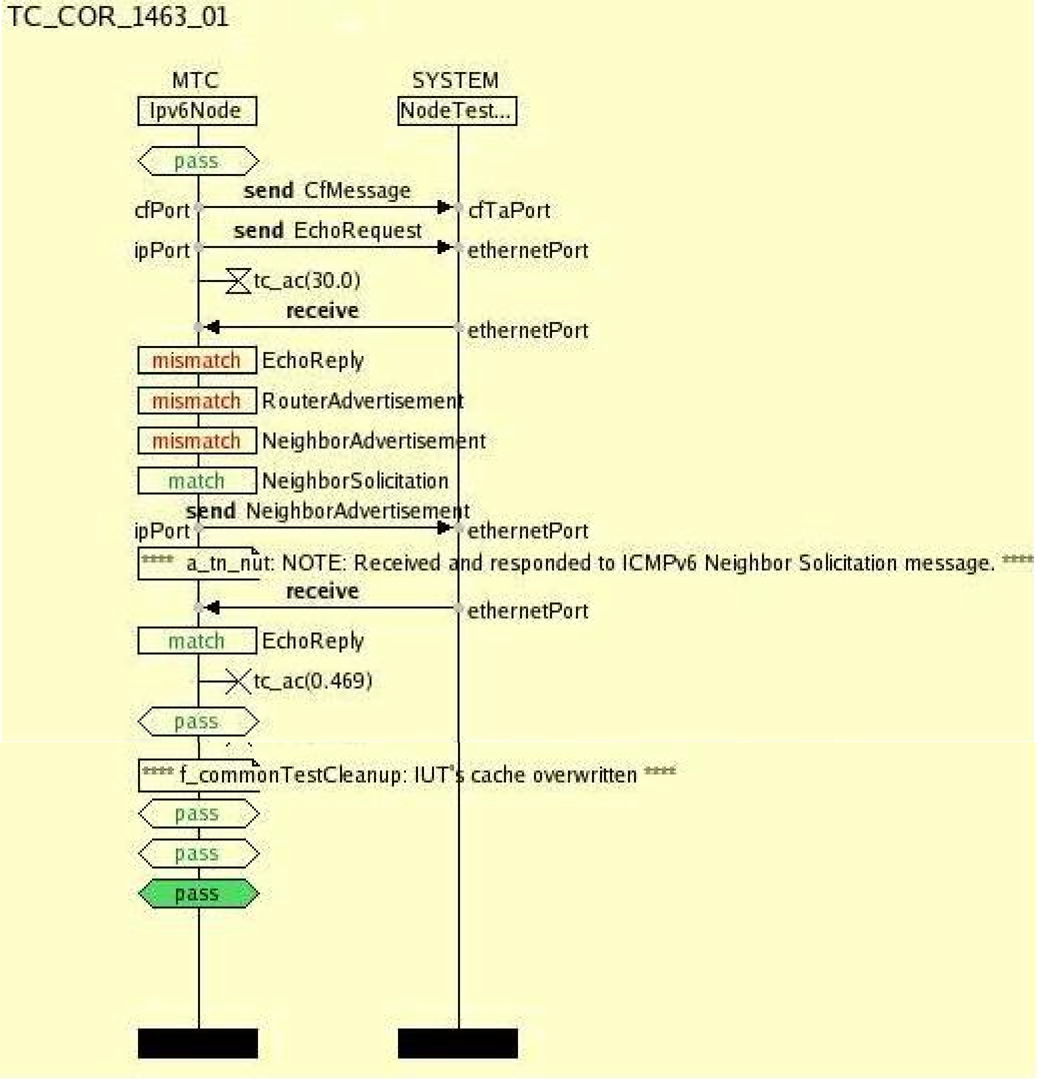
Test Platform Provider: ETSI testbed

Equipment Manufacturers:

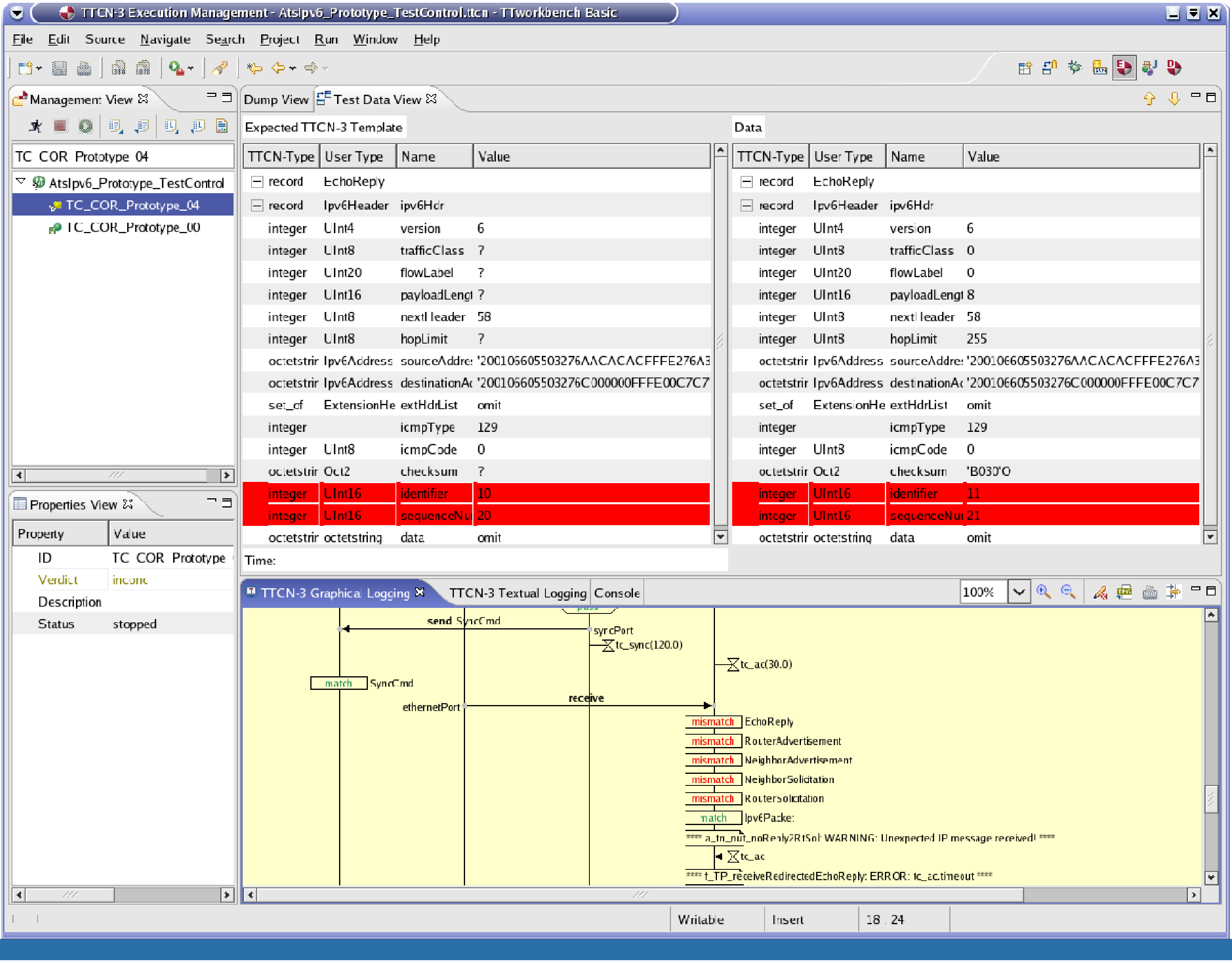
- Linux: USAGI, MIPL, OpenIKEv2
- FreeBSD: KAME, SHISA, OpenIKEv2

Internet
ETSI_PUBLIC
217.167.116.x and 217.167.117.x
Mask: 255.255.254.0
eth0:
217.16
FErouter1
Router IPv4
+ static NAT
(public 217.167.116.41 to .45)
+ Firewall
SSH
(secured remote control)
+ NIS Server
+ NFS Server
+ NTP Server
- VMware
- SVN repository
- Mantis / Apache server
+ mobilan (172.30.8.10)
+ atlas (172.30.8.12)
eth0: 172.30.8.15
(ext: 217.167.116.45)
= prefix advertisement
switch01
eth1:
172.30.8.30
172.30.8.1 to 172.30.8.30
Mask : 255.255.255.224
/27
eth0: 172.30.8.3
(ext: 217.167.116.43)
Router3
MN1
::1
eth0
mac: AE:AC:AC:DD:DD:10
eth1
mac: AE:AC:AC:AA:AA:31
eth2
mac: AE:AC:AC:AA:AA:32
MN2
::2
eth0
mac: AE:AC:AC:DD:DD:20
PF276a: 2001:660:5503:276a /64
PF276c: 2001:660:5503:276c /64
Router1
eth0: 172.30.8.1
(ext: 217.167.116.41)
hub01
FN1
FN2
hub03
Router2
eth0: 172.30.8.2
(ext: 217.167.116.42)
eth2
mac: AE:AC:AC:AA:AA:12
eth1
mac: AE:AC:AC:AA:AA:21
eth1
mac: AE:AC:AC:AA:AA:11
HA1
::10
CN1
CN2
HA2
::20
eth2
mac: AE:AC:AC:AA:AA:22
eth0
mac: AE:AC:AC:BB:BB:10
eth0
mac: AE:AC:AC:CC:CC:10
eth0
mac: AE:AC:AC:CC:CC:20
eth0
mac: AE:AC:AC:BB:BB:20
HN1
hub02
hub04
HN2
PF276b: 2001:660:5503:276b /64
PF276d: 2001:660:5503:276d /64

TTCN-3 Graphical Logging


TC_COR_1463_01
MTC
Ipv6Node
SYSTEM
NodeTest...
pass
send CfMessage
cfPort
cfTaPort
send EchoRequest
ipPort
ethernetPort
tc_ac(30.0)
receive
ethernetPort
mismatch EchoReply
mismatch RouterAdvertisement
mismatch NeighborAdvertisement
match NeighborSolicitation
send NeighborAdvertisement
ipPort
ethernetPort
**** a_tn_nut: NOTE: Received and responded to ICMPv6 Neighbor Solicitation message. ****
receive
ethernetPort
match EchoReply
tc_ac(0.469)
pass
**** f_commonTestCleanup: IUT's cache overwritten ****
pass
pass
pass

TTCN-3 Result Visualisation



Property	Value
ID	TC COR Prototype
Verdict	inconc
Description	
Status	stopped

Expected TTCN-3 Template

TTCN-Type	User Type	Name	Value
record	EchoReply		
record	Ipv6Header	ipv6Hdr	
integer	UInt4	version	6
integer	UInt8	trafficClass	?
integer	UInt20	flowLabel	?
integer	UInt16	payloadLengt	?
integer	UInt8	nextHeader	58
integer	UInt8	hopLimit	?
octetstrir	Ipv6Address	sourceAddre:	'200106605503276AACACACFFFE276A3
octetstrir	Ipv6Address	destinationA	'200106605503276C000000FFFE00C7C7
set_of	ExtensionHe	extHdrList	omit
integer		icmpType	129
integer	UInt8	icmpCode	0
octetstrir	Oct2	checksum	?
integer	UInt16	identifier	10
integer	UInt16	sequenceNu	20
octetstrir	octetstring	data	omit

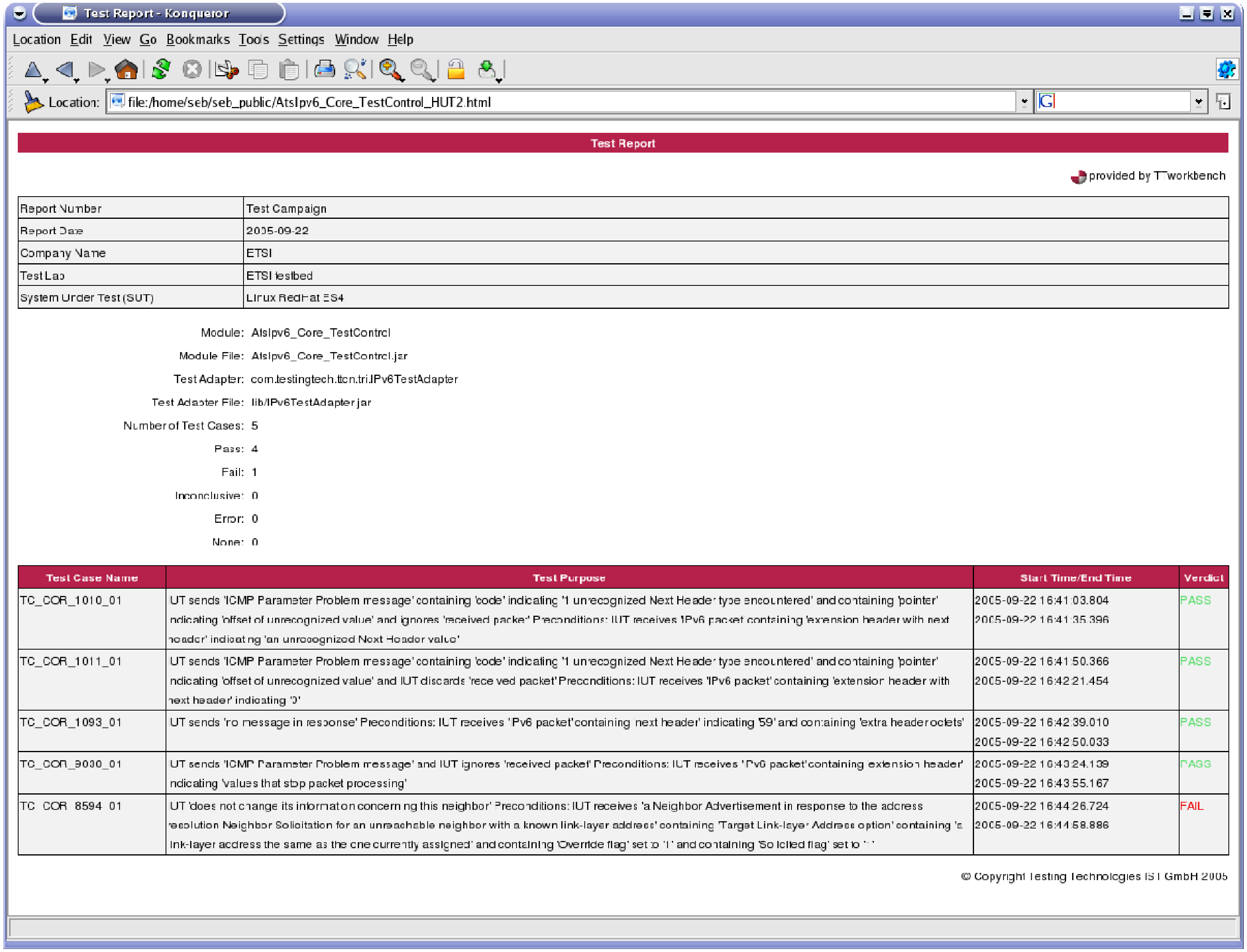
Time:

Data

TTCN-Type	User Type	Name	Value
record	EchoReply		
record	Ipv6Header	ipv6Hdr	
integer	UInt4	version	6
integer	UInt8	trafficClass	0
integer	UInt20	flowLabel	0
integer	UInt16	payloadLengt	8
integer	UInt8	nextHeader	58
integer	UInt8	hopLimit	255
octetstrir	Ipv6Address	sourceAddre:	'200106605503276AACACACFFFE276A3
octetstrir	Ipv6Address	destinationA	'200106605503276C000000FFFE00C7C7
set_of	ExtensionHe	extHdrList	omit
integer		icmpType	129
integer	UInt8	icmpCode	0
octetstrir	Oct2	checksum	'B030'O
integer	UInt16	identifier	11
integer	UInt16	sequenceNu	21
octetstrir	octetstring	data	omit

```
send SyncCmd
syncPort
tc_sync(120.0)
tc_ac(30.0)
match SyncCmd
receive
ethernetPort
mismatch EchoReply
mismatch RouterAdvertisement
mismatch NeighborAdvertisement
mismatch NeighborSolicitation
mismatch RouterSolicitation
match Ipv6Packet
**** a_tn_nut_noReply2RtSol: WARNING: Unexpected IP message received! ****
tc_ac
**** f_TP_receiveRedirectedEchoReply: ERROR: tc_ac.timeout ****
```

Test Report Generation



Test Report - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: file:/home/seb/seb_public/AtsIpv6_Core_TestControl_HUT2.html

Test Report

provided by TTworkbench

Report Number	Test Campaign
Report Date	2005-09-22
Company Name	ETSI
Test Lab	ETSI testbed
System Under Test (SUT)	Linux RedHat ES4

Module: AtsIpv6_Core_TestControl

Module File: AtsIpv6_Core_TestControl.jar

Test Adapter: com.testingtech.ttcn.tri.IPv6TestAdapter

Test Adapter File: lib/IPv6TestAdapter.jar

Number of Test Cases: 5

Pass: 4

Fail: 1

Inconclusive: 0

Error: 0

None: 0

Test Case Name	Test Purpose	Start Time/End Time	Verdict
TC_COR_1010_01	IUT sends 'ICMP Parameter Problem message' containing 'code' indicating '1 unrecognized Next Header type encountered' and containing 'pointer' indicating 'offset of unrecognized value' and ignores 'received packet' Preconditions: IUT receives 'IPv6 packet containing 'extension header with next header' indicating 'an unrecognized Next Header value'	2005-09-22 16:41:03.804 2005-09-22 16:41:35.396	PASS
TC_COR_1011_01	IUT sends 'ICMP Parameter Problem message' containing 'code' indicating '1 unrecognized Next Header type encountered' and containing 'pointer' indicating 'offset of unrecognized value' and IUT discards 'received packet' Preconditions: IUT receives 'IPv6 packet' containing 'extension header with next header' indicating '0'	2005-09-22 16:41:50.366 2005-09-22 16:42:21.454	PASS
TC_COR_1093_01	IUT sends 'no message in response' Preconditions: IUT receives 'IPv6 packet' containing 'next header' indicating '59' and containing 'extra header octets'	2005-09-22 16:42:39.010 2005-09-22 16:42:50.033	PASS
TC_COR_9030_01	IUT sends 'ICMP Parameter Problem message' and IUT ignores 'received packet' Preconditions: IUT receives 'IPv6 packet' containing 'extension header' indicating 'values that stop packet processing'	2005-09-22 16:43:24.139 2005-09-22 16:43:55.167	PASS
TC_COR_8594_01	IUT 'does not change its information concerning this neighbor' Preconditions: IUT receives 'a Neighbor Advertisement in response to the address resolution Neighbor Solicitation for an unreachable neighbor with a known link-layer address' containing 'Target Link-layer Address option' containing 'a link-layer address the same as the one currently assigned' and containing 'Override flag' set to '1' and containing 'Solicited flag' set to '1'	2005-09-22 16:44:26.724 2005-09-22 16:44:58.886	FAIL

© Copyright Testing Technologies IST GmbH 2005

Conclusions

- ❑ Standards and Test Specification development should be systematic
- ❑ Standards should be designed for interoperability and testability
- ❑ Synchronize testing with standards and product development
- ❑ Perform Interoperability events !
- ❑ Validate Standards and Products through testing
- ❑ ETSI can help you achieve the above!

Thank you for your attention

Contact:

Sebastian Müller, Senior Technical Expert

sebastian.mueller@etsi.org