Practical Use of TTCN-3 for Text Based protocol Testing

4th of May 2004

Stephan Schulz, Ph.D.

Senior Research Engineer
Mobile Networks Laboratory
Nokia Research Center, Helsinki

Overview

- Background on (Testing of) Text-Based Protocols
- Writing Test Cases for Text-Based Protocols with T3
- T3 Test Platform Design for Text-Based Protocols
  - ⇨ Derivation of TTCN-3 types for text message
  - ⇨ Implementation of a Generic Text-based Codec System
- Conclusions
- References

Background on Text-Based Protocols

- Message on the line follows textual encoding
- Originated from internet community
  - ⇨ “IP convergence” is slowly but surely bringing them into the telecom world, e.g., SIP
- 👍 Designed to be humanly understandable
  - 👎 Potentially “huge” message size
- General message structure:
  - ⇨ Collection of text headers/descriptors followed by some payload (which may be of binary format)
- Examples:
  - ⇨ SIP, SDP, HTTP, RTSP, MEGACO ABNF, ... telnet, ftp, smtp

A SIP Message Example

```
REGISTER sip:antipolis.net SIP/2.0
Via: SIP/2.0/UDP 172.23.243.90:5060
       ;branch=z9hG4bKA65E34CD46334B958A7E559C9C011230
From: Alice in Chains <sip:alice@antipolis.net>;tag=0g4E_UWw-J
To: Alice in Chains <sip:alice@antipolis.net>
Contact: "Alice in Chains" <sip:alice@172.23.243.90:5060>
Call-ID: 014B8E49AD6A40B58A94A4EE6A213BA6@antipolis.net
CSeq: 44 REGISTER
Expires: 500
Max-Forwards: 70
User-Agent: X-Lite build 1079
Content-Length: 0
```

Reality of Text-Based Protocols

- Myth:
  “I can specify/check text messages easily by hand”
- Possibility of white space
  - ⇨ Not well defined across different protocols
  - ⇨ Doesn’t affect information content (only appearance)
- Multiple ways to encode the same information
  - ⇨ Negatively affects message readability
  - ⇨ Examples:
    - Multiple header occurrence
    - Long versus compact form

↳ Handling of text messages in testing ***is*** a challenge!

A “not so nice” version of the SIP Message Example

```
REGISTER sip:antipolis.net SIP/2.0
f: Alice in Chains <sip:alice@antipolis.net> ;   tag=0g4E_UWw-J
max-FORWARDS: 70
t: Alice in Chains
<sip:alice@antipolis.net>
m: "Alice in Chains"            <sip:alice@172.23.243.90:5060>
i: 014B8E49AD6A40B58A94A4EE6A213BA6@antipolis.net
cseq: 44

REGISTER
EXpiRes: 500
Via: SIP   /   2.0   /UDP
   172.23.243.90:5060
      ;
branch=z9hG4bKA65E34CD46334B958A7E559C9C011230
User-Agent: X-Lite
                                                     build 1079
l: 0
```

short forms

white space

+ other funnies

Text-Based Protocol <u>Testing</u>

- Different testing philosophy at IETF
  - ⇨ Adherence to a specification is “verified” once two implementations interoperate (= no testing against spec)
  - ⇨ Concrete need for testing in implementation certification!
- State of text-based protocol conformance testing
  - ⇨ IETF
    - “torture tests” (=send messages with funny encoding)
    - Python/Perl scripts
  - ⇨ ETSI
    - “trial” TTCN-3 SIP test suite exists
  - ☹ Still no conformance test specifications for any text-based protocol required by any standardization body today

TTCN-3 for Text-Based Protocol Testing

- **The** hope to make conformance testing for (IETF) text-based protocols finally happen!
- Core notation is key
  - Similar feel as programming language
- Important new TTCN language features
  - Regular expressions for sophisticated checking of textual values (not possible with TTCN-2!)

    ```
    template charstring any_lcw := pattern “[a-z]#(1,)”;
    ```

  - New string handling operations drastically simplify working with textual values

    ```
    var charstring v_str := “TTCN-” & int2str(3);
    ```

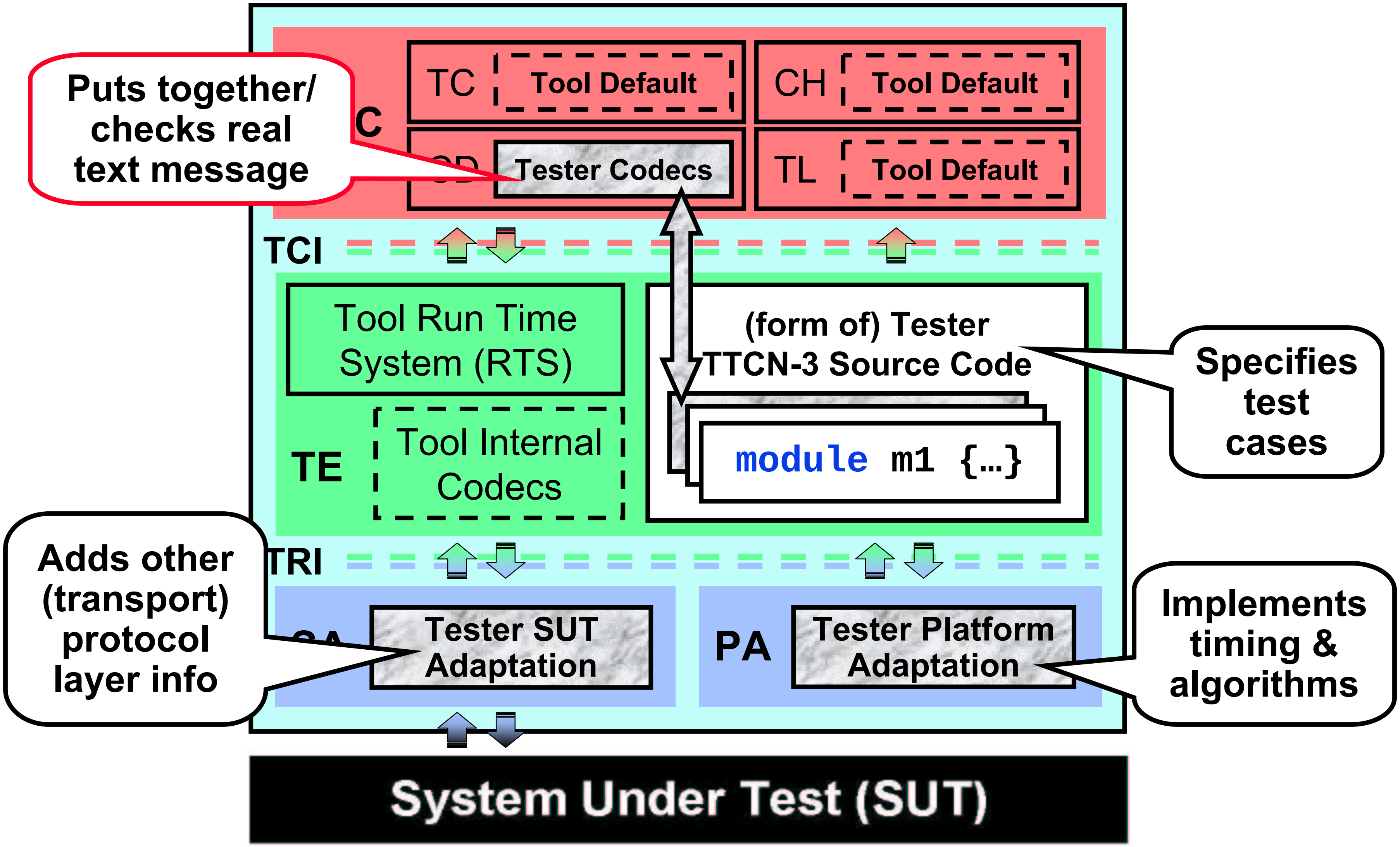
  - Encoding attributes down to level of TTCN-3 type fields

Test Systems for Text-Based Protocol Testing

Now appropriate support in TTCN-3 language but ...

- ... real challenge is design of <u>good</u> test platforms for text-based protocols!
  - ⇨ Lets test case implementers focus on the *information* contained in protocol messages
  - ⇨ Leaves syntax checking and assembly of text messages to test platform – *automates* it
  - ⇨ Shields test case writers from the quirks of text-based protocols (e.g., white space, short forms, etc)

TTCN-3 Test Systems for Text-Based Protocols



Puts together/ checks real text message

C

TC

Tool Default

CH

Tool Default

CD

Tester Codecs

TL

Tool Default

TCI

Tool Run Time System (RTS)

TE

Tool Internal Codecs

(form of) Tester TTCN-3 Source Code

`module m1 {…}`

Specifies test cases

TRI

Adds other (transport) protocol layer info

SA

Tester SUT Adaptation

PA

Tester Platform Adaptation

Implements timing & algorithms

System Under Test (SUT)

About TTCN-3 code and (text) codecs

- Codec implementation is *closely* coupled to the representation of protocol messages in TTCN-3
  - ⇨ Generally one or more modules define a (abstract) structure of every protocol message using TTCN-3 types
  - ⇨ Codecs (C/Java code) assemble real protocol message to be sent based on this structuring and vice versa

  … therefore choice in TTCN-3 message structuring *drives* a codec/testing platform implementation!

- Text protocol standards only specify message encoding – not their *abstract* definition!
- Strong need for such a definition to really unleash the power of TTCN-3!

So the “real” Challenge: Text Messages in TTCN-3

⇨ One approach to extract an abstract representation for text messages!

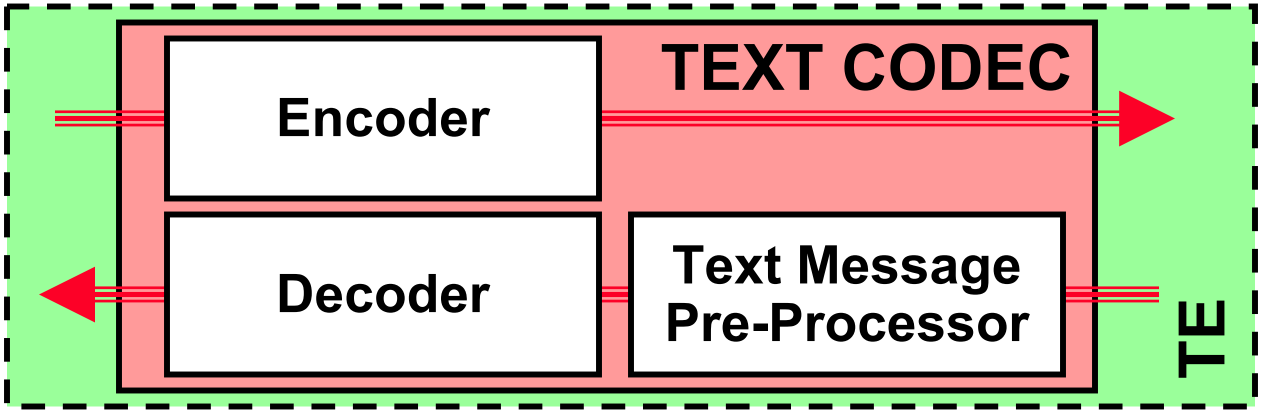
- Benefits:
  - ⇨ More readable, robust, consistent and high quality test suite implementations
  - ⇨ Creates a textual encoding scheme
  - ⇨ Enables creation of one single codec implementation for many text-based protocols
  - ⇨ Reduces test system development time for new or variations of text protocols
  - ⇨ [Possibility to change encoding of message content]
- Proposal for ABNF defined textual encoding exists
  - ⇨ Nokia TestCom 2004 paper (see References)

Text Codec Implementations

- Current approaches
  - ⇨ Open source parsers
    - Not independent of protocol
    - Quality of encoding error report?
  - ⇨ Generated parsers (yacc etc)
    - Poor quality of codec error report: “ok” or “not ok”
    - Needs more extra work than may be expected …
  - ⇨ Not really designed for testing!
- A generic text codec implementation
  - ⇨ Relies on a generic TTCN-3 type derivation approach
  - ⇨ Protocol independent
  - ⇨ Designed for testing (e.g., useful error reports)

A Generic Text Codec Implementation

- Preprocessor normalizes incoming messages
- En/decoder driven by encoding information specified as part of TTCN-3 message types
- Technology successfully applied in Nokia product testing of SIP, SDP, MEGACO ABNF, etc



NOKIA

Conclusions

- Text-based protocol conformance testing finally possible with TTCN-3 and a reality!
  - New language features have proven to simplify TTCN code
  - Also key factor: easy-to-learn TTCN-3 core notation
- Real challenge however lies in representation of text message content with TTCN-3 types
  - Proposal for TTCN-3 type derivation from ABNF exists
  - Single generic text codec implementation possible
- TTCN-3 community is ready but (currently) requests from standardization organizations are lagging
  - Expected breakthrough with 3GPP Rel5 testing
  - Now possibility to positively influence IETF!

References

- TTCN-3
  - ETSI ES 201 873-1: The TTCN-3 Core Language
  - ETSI ES 201 873-5: The TTCN-3 Runtime Interface
  - ETSI ES 201 873-6: The TTCN-3 Control Interface
- Test System Design for Text-Based Protocols
  - S.Schulz, “Derivation of Abstract Protocol Type Definitions for the Conformance Testing of Text-Based Protocols”, *Proceedings of TestCom Conference*, 177-92, Oxford, UK, 2004.