Specification of Test Case Structure for Protocol Testing Automation

draft-cui-bmwg-testcase-spec

Yong Cui, **Yunze Wei**, Xiaohui Xie
Tsinghua University

Why We Need a Standardized Test Case Specification for Protocol Testing?

- **Challenges in Current Practice**
  - Test cases are often written in **ad hoc formats**, lacking consistency and reusability
  - Difficult to **share, reproduce, and automate** across vendors and labs
  - Automation frameworks require **structured input** — informal test descriptions don't scale

draft-cui-nmrg-auto-test

- **What We Aim to Solve**
  - Introduce a **machine-readable, protocol-agnostic test case schema**
  - Standardizing the **structure** of test cases
  - Support consistent test design, execution, and interpretation
  - Ensures **reproducible** test cases without ambiguity
  - Enable automated pipelines for executing and verifying test cases

A Structured Specification for Protocol Test Cases

- Define **6 key components** for protocol test cases
  - *Metadata*, *Topology*, *Setup*, *Parameters*, *Procedure*, *Evaluation Criteria*
- Classify test cases into **5 types**
  - Conformance, Functional, Performance, Interoperability, Robustness
- Provide **a machine-readable example** to enable automation

- **Test Case Structure Overview**

Component	Description
Metadata	Test ID, title, purpose, protocol, references
Topology	Node roles, links, addressing
Setup	Initial DUT/tester configurations
Parameters	Protocol knobs, traffic profiles
Procedure	Step-by-step actions and expected behavior
Evaluation	Metrics, pass/fail criteria, logs required

- **Application Scenarios**
  - Vendor internal testing
  - Pre-deployment acceptance testing in production networks

Thank You!

draft-cui-bmwg-testcase-spec

Yong Cui, **Yunze Wei**, Xiaohui Xie
Tsinghua University