Conformance tests for Multipath TCP

draft-coene-mptcp-conformance-00

Yvan Coene

87th IETF
Berlin, Germany

Why conformance testing ?

- Important, somewhat complex extension to TCP
  - Changes many semantics of TCP
  - Assessing conformance is *not* trivial
- Needed to guarantee interoperability

Test Objectives

- Reference: RFC 6824
  - Prescriptive statements (`MUST`, `SHOULD`, ...) (~100)
  - Behaviour descriptions

Example Test Objective	
Name	Send SYN on new connection
Reference	`p. 14: Connection Initiation begins with a SYN, SYN/ACK, ACK exchange on a single path.`
Description	Establish a new connection to the TS on the SUT through the socket interface, i.e. by calling `connect()`. FAIL if no SYN received.
Req. Level	`MUST`

Test Cases

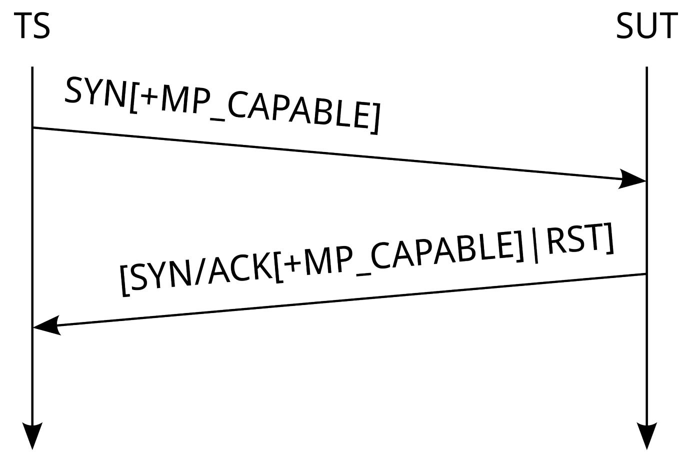
- Derived manually from the specification
- Maximize coverage of test objectives
- Parametric pseudo-code
  - Input parameters
  - Parameters domain
  - covers a class of similar protocol executions

Passive connection opening

```
INPUT PARAMETERS
  "SYN with MP_CAPABLE" in {true, false}
  "SYN MP_CAPABLE MPTCP version" in {0, 1}
  "SYN MP_CAPABLE flags" in {A|H, A, A|B|H, B|H}

PARAMETERS DOMAIN
  {false} x {0} x {A|H} u
  {true} x {0, 1} x {A|H, A, A|B|H, B|H}

DESCRIPTION
  send SYN that corresponds to input parameters

  if SYN flag B set
    assert no response received ("Flag B ignored")
    exit
  else
    assert response received ("send SYN/ACK upon SYN reception")
  ...
```

Passive connection opening

- Normal case
- Input parameters:
  - “SYN with MP_CAPABLE”
    = *true*
  - “SYN MP_CAPABLE MPTCP VERSION”
    = *0*
  - “SYN MP_CAPABLE Flags”
    = *A|H*



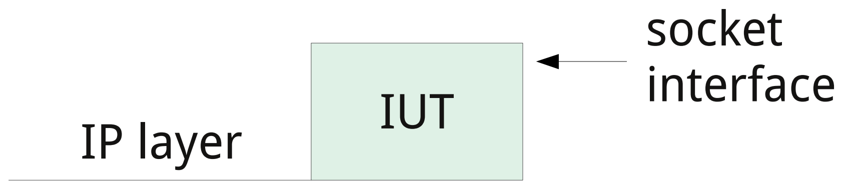
- Expected output:
  - SYN/ACK
  - RST
  - Nothing

Test Architecture

- Implementation under test
  - Hosted by the system under test
- Lower and upper testers
  - Lower tester hosted by the test system

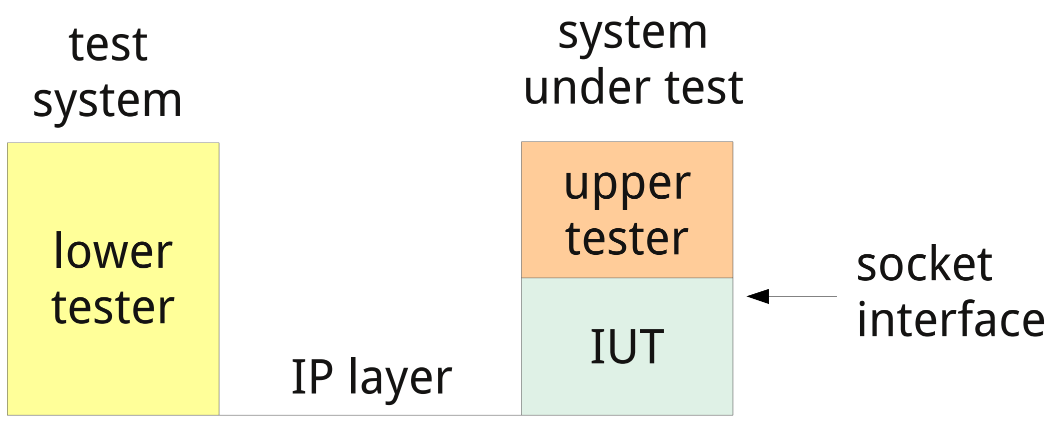
Test Architecture

- Implementation under test
  - Hosted by the system under test
- Lower and upper testers
  - Lower tester hosted by the test system



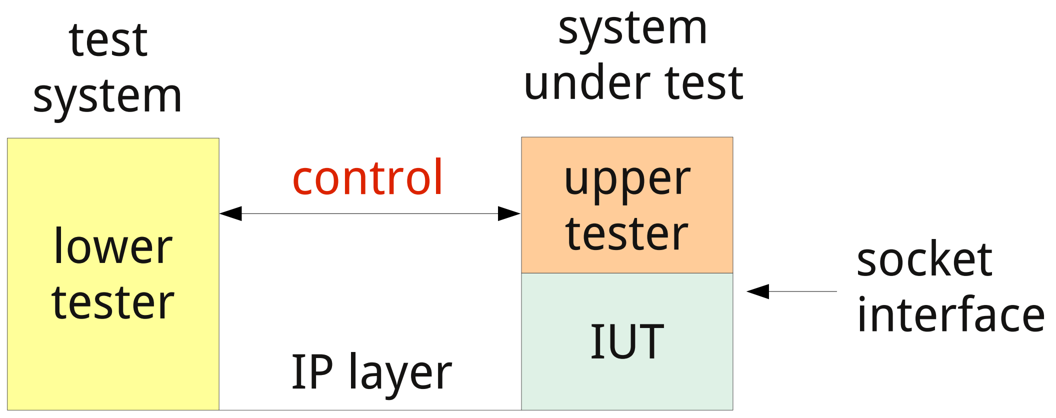
Test Architecture

- Implementation under test
  - Hosted by the system under test
- Lower and upper testers
  - Lower tester hosted by the test system



Test Architecture

- Implementation under test
  - Hosted by the system under test
- Lower and upper testers
  - Lower tester hosted by the test system



Implementation

- Based on *libnet* and *libpcap*
- Around 3500 lines of C source code
- Open-source, available at `bitbucket.org/ycoene`
- Divided in two modules
  - Master (upper tester) (GNU/Linux)
  - Slave (lower tester) (hopefully portable)
- Covers draft and
  - addition of subflows
  - connection termination
  - a few robustness aspects

Results

- Linux kernel MPTCP v0.86
- Minor specification violations
  - Flag B, 64-bit DSN, segments without DATA_ACK
  - Some of them already known
- Instabilities
  - SUT crashed when receiving wrong DSS

Conclusion

- Current state
  - Partially documented testing tool
- Any interest to further document tests ?