MITTEN

Scenario-Based Consensus Protocols Testing

Functori

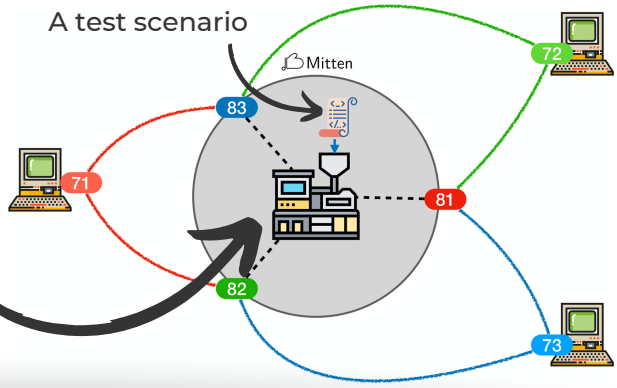
Parameterizable
Man-In-The-Middle
proxy to run test scenarios
on consensus protocols
implementations

DETECT DISCREPANCIES BETWEEN DESIGN AND CODE

A test scenario

node whose port is 71 sees node whose port is 73 through mitten proxy port 83

Matching exchanged messages against the scenario and forwarding desired ones



FEATURES

- ✔ Declarative scenario description language
- ✔ Combinators for scenario steps (sequence, disjunctions, loops, parallel composition)
- ✔ Assertions written with the full power of OCaml
- ✔ Modular implementation (engine for forwarding, engine for matching, etc.)
- ✔ Man-In-The-Middle proxy to monitor and act on messages on P2P network
- ✔ Messages reorg, delay or loss, clocks drift
- ✔ No instrumentation required to run code (nodes)

EXAMPLE

```
let scenario = seq [
    wait (propose a (level 3) (round 0) p1 [a;b]);
    (preendorse a (level 3) (round 0) __ [a;b] &&
    preendorse b (level 3) (round 0) __ [a;b]) ;
    endorse b (level 3) (round 0) __ [a;b];
    propose b (level 3) (round 1) p1 [a;b]
]
```

At level 3, round 1, node b should re-propose payload p1 it already endorsed at (3,0)

Functori

université PARIS-SACLAY

nomadic labs

contact@functori.com
https://functori.com/mitten
https://www.youtube.com/watch?v=KLasrzyF5Rw

Ç Bozman - M Iguernlala - M Laporte - M Levillain - A Mebsout - S Conchon