Automated Verification of Network Function Binaries

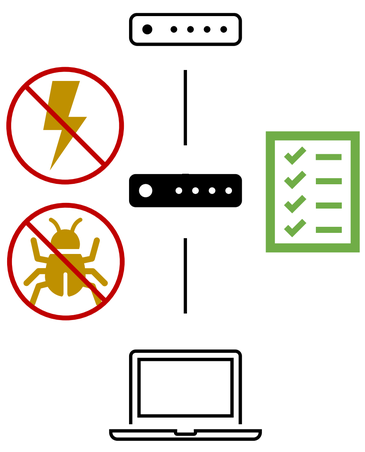
EPFL

Solal Pirelli, Akvilė Valentukonytė,
Katerina Argyraki, George Candea

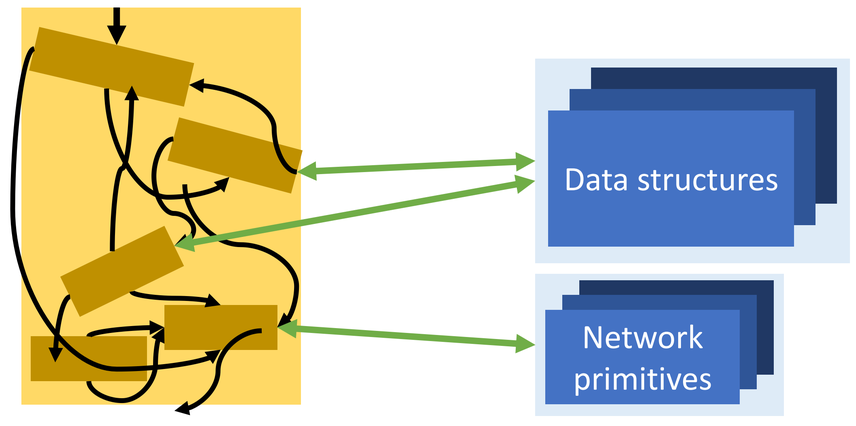
Describing data structures with maps
enables the automated verification
of network function binaries

Previous automated network function verification efforts:
- Require operators to have access to source code
- Require developers to use specific data structures
- Require experts to write invariants for the known data structures

We remove these requirements,
and only require map-based contracts to use any data structure



Goals:
- Crash freedom,
- memory safety,
- spec compliance (e.g., RFC)



Key idea to verify binaries:
- Observe interactions (= calls) with the environment, i.e., data structures + network

State: map M (value → age)

Precondition:
$\text{length}(M) > 0$

Postcondition:
$\text{contains}(M, \text{result}) \wedge$
$M' = \text{remove}(M, \text{result}) \wedge$
$\forall\ (v,a) \in M: a \leq \text{get}(M, \text{result})$

Example contract for
a least-recently-used cache
“evict” operation

Individual network functions
verify in <2min on a laptop

Prototyping is now easy,
our performance beats Click

Paper and code: dslab.epfl.ch/research/klint