HUGINN-TCP

Hannes Mehnert, *University of Cambridge*
Michael Norrish, *NICTA*

REMS workshop, 27 May 2016

MOTIVATION

- Transmission Control Protocol / Internet Protocol (TCP/IP)
- Widely deployed protocol
- "Specification" is distributed over several RFC
- No publicly available test suite
- Goal: turnkey solution (testing, validation of traces)

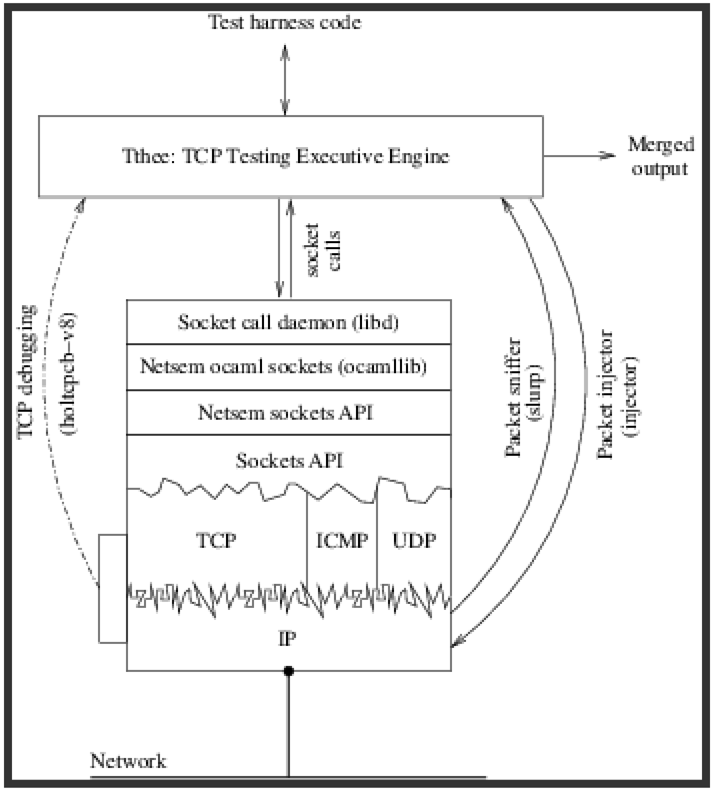
MODELING TCP/IP

- There is no formal specification
- There are a bunch of implementations out there
- Better be bug compatible
- Bugs usually result in more delay, less throughput
- Or invalid checksums and dropped packets

NETWORK SEMANTICS

- Research here in Cambridge 2000-2009
- TCP/IP model in HOL4 (~50000 lines)
- Test harness (~2000 lines) in OCaml
- Records traces
- FreeBSD-4.6, Linux-2.4.20, Windows XP

NETSEM STACK


Test harness code
Tthee: TCP Testing Executive Engine
Merged output
socket calls
TCP debugging (holtcpcb-v8)
Socket call daemon (libd)
Netsem ocaml sockets (ocamllib)
Netsem sockets API
Sockets API
TCP
ICMP
UDP
IP
Packet sniffer (slurp)
Packet injector (injector)
Network

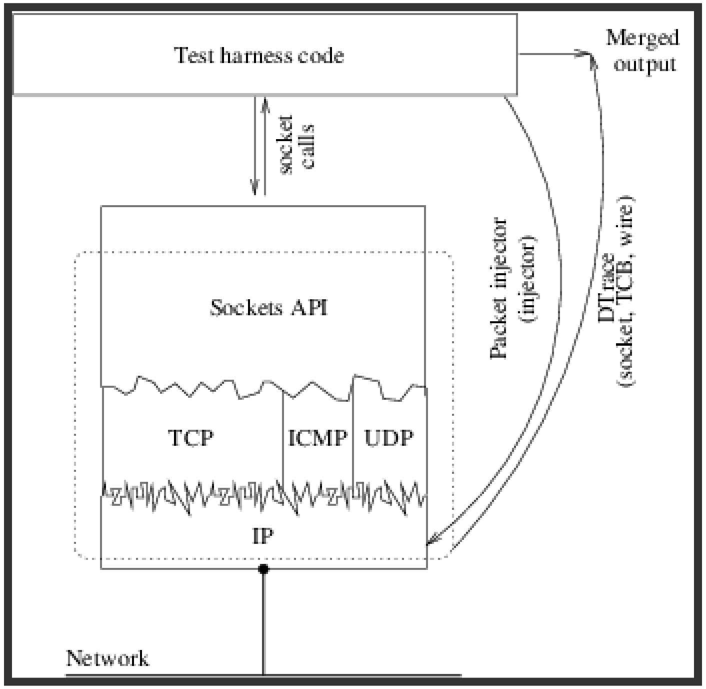
HUGINN (CHECK)

- Reanimated NetSem using a current HOL4
- PolyML instead of MoscowML
- On 2000 traces recorded 10 years ago:
- 20x faster
- Mostly same result
- https://www.cl.cam.ac.uk/~hm519/netsem/

HUGINN (TRACE)

- FreeBSD has DTrace support
- No need to wrap APIs
- Fewer code (10000loc instrumentation, now 450 lines D)
- Single trace source: no need for merge

HUGINN STACK



HUGINN (TEST)

- Single process, single thread
- Sets up control connection to remote
- Blocking semantics
- Instead of 25000loc now 600loc (not feature-complete)

FUTURE WORK

- Finish tests, new tests (congestion control)
- Model: TCP features (SACK, ...)
- Further speedup (finer instrumentation)
- Port tracing to Linux/MacOSX/Illuminos
- Code and model coverage for tests

CONCLUSION

- Model works with old traces!
- First 40 traces run and validate with FreeBSD-CURRENT
- 2-clause BSD licensed
- https://github.com/PeterSewell/netsem