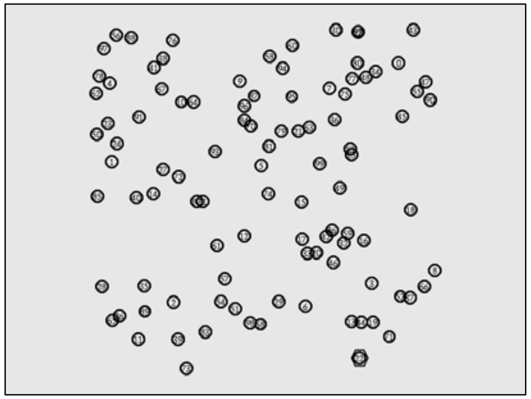
Experiences from Redeploying Simulation-based Code in the Real World

Georg Wittenburg

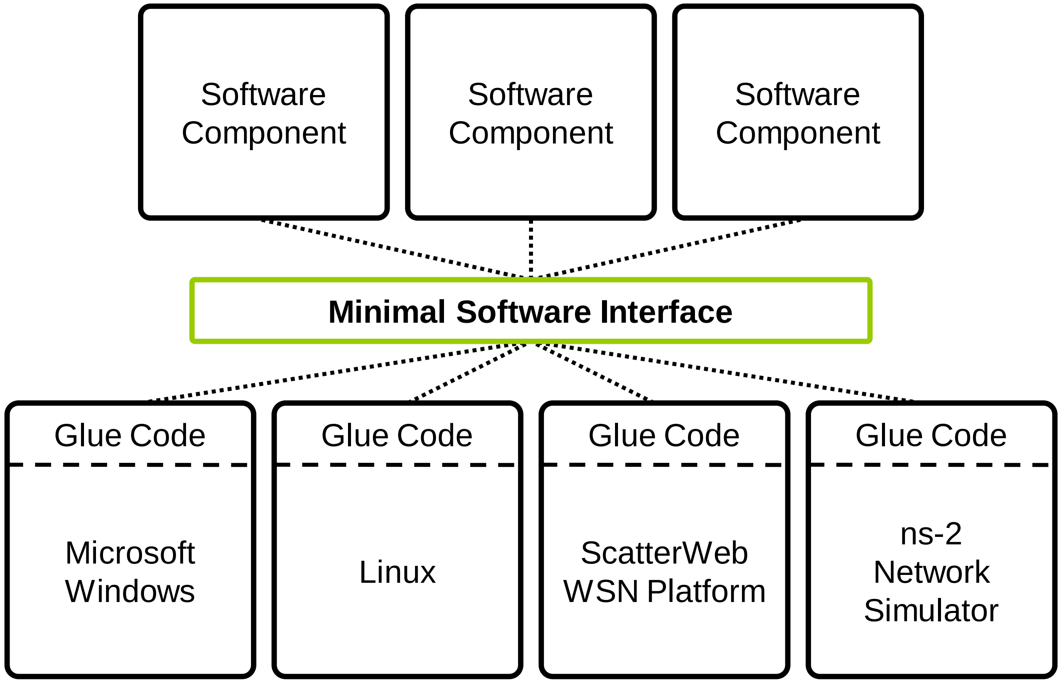
Institut National de Recherche en
Informatique et en Automatique (INRIA)

From Simulations to the Real World



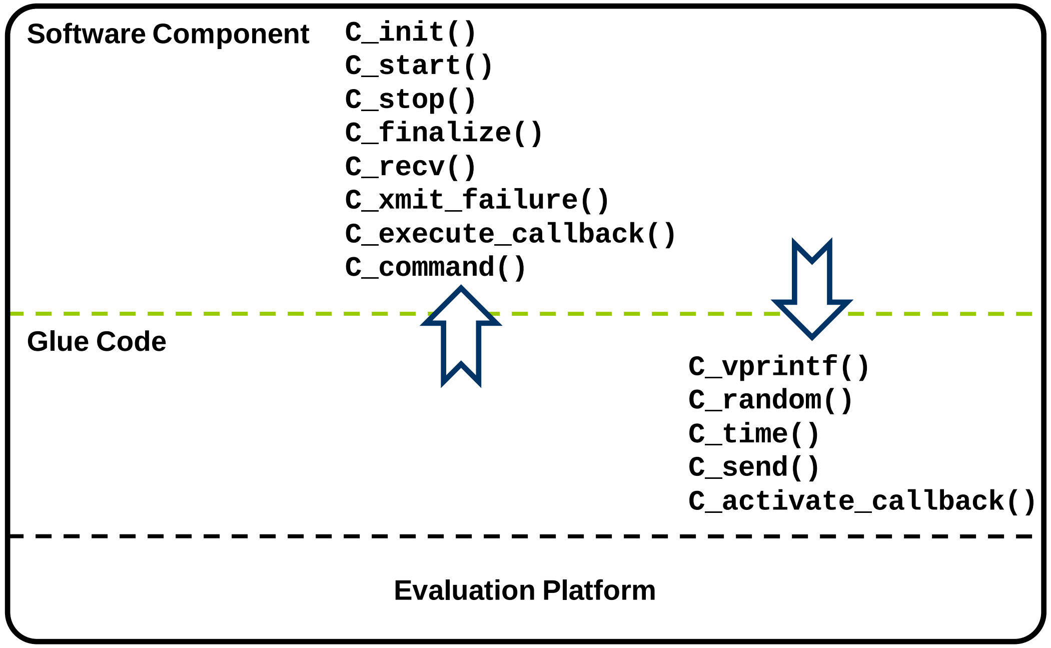
1. Simulations using the ns-2 network simulator
2. Intermediate deployment on IEEE 802.3 Ethernet
3. Real-world deployment on IEEE 802.11 wireless testbed

Evaluation Architecture



- Minimal software interface facilitates cross-platform evaluation
- Requires minimal platform-specific glue code

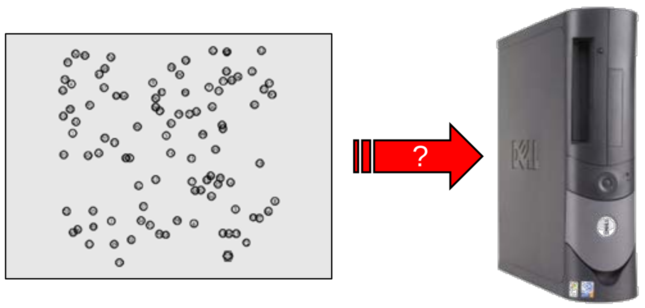
Software Interface



Functions for:

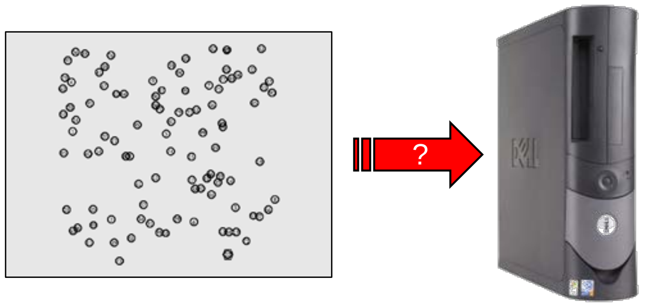
- Lifecycle management
- Output
- Packet-based communication
- Delayed execution
- Access to time and randomness sources

From Simulations to Real Hardware (1)



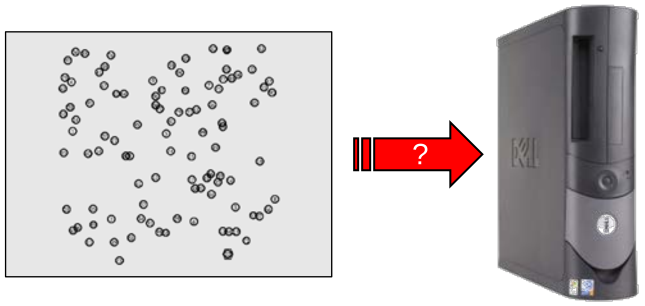
- Time passes while code is executed
  - Scalability of algorithms
    - Excessive CPU load causes packet loss
    - Complex updates of data structures per packet prohibitively expensive
    - ➢ ***Don’t assume that your data structures are up to date when handling individual packets.***
  - Comparisons of time-related values
    - Checks and branching fail due to false assumptions on execution speed
    - ➢ ***Checking for equality (or near equality) of time spells trouble!***

From Simulations to Real Hardware (2)



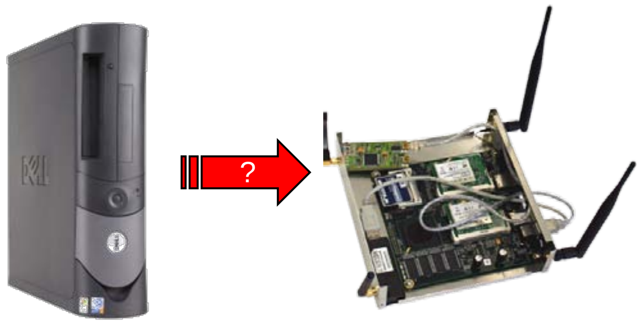
- Addressing issues
  - Local address
    - No unique local address (127.0.0.1, …)
    - No standard (i.e., POSIX-like) interface to establish local address
    - ➢ ***Have lots of sanity checks when using OS-specific heuristics to establish your address!***
  - Broadcast address / mask
    - Network equipment will drop packets with “incorrect” broadcast packets
    - ➢ ***Watch out for dropped broadcast packets! Know your broadcast domain!***

From Simulations to Real Hardware (3)



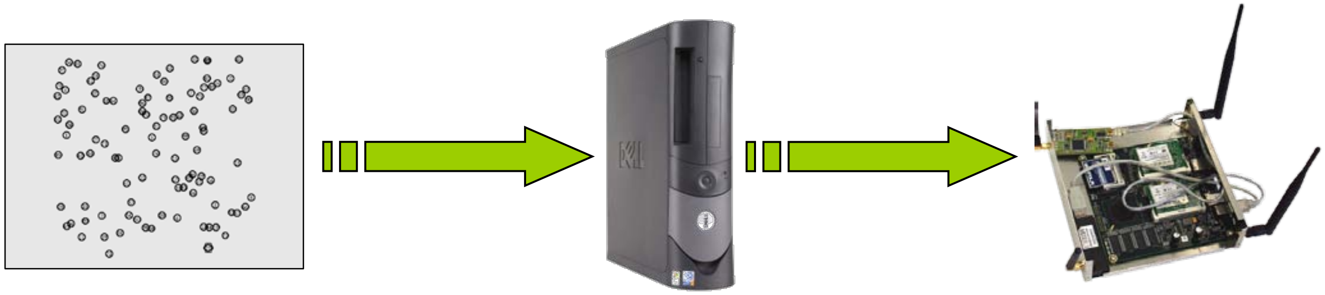
- Uninitialized variables / memory leaks
  - Sometimes masked by your glue code
  - Hint at implementation errors in your protocols
  - Hard to reproduce when running on real hardware
  - ➢ ***Run your code using debugging tools (e.g., gdb, valgrind, …) whenever possible!***

From Real Hardware to the Real World



- Wireless communication
  - Link quality
    - Algorithms / protocols with simplistic graph-like network model break
    - Single packets used as indication for existence of links
    - ➢ ***Don't rely on single packets; use link metrics!***
    - Some algorithms / protocols don't support link metrics / weighted graphs
    - ➢ ***Use threshold values to translate between continuous link metric and "Boolean" link, e.g., ETX ≤ 2.0!***
  - Link directionality (i.e., unidirectional links)
    - Many algorithms / protocols implicitly assume bidirectional links
    - Unidirectional links are common
    - ➢ ***Choose your link metric to satisfy assumptions higher-level components!***

Lessons Learnt

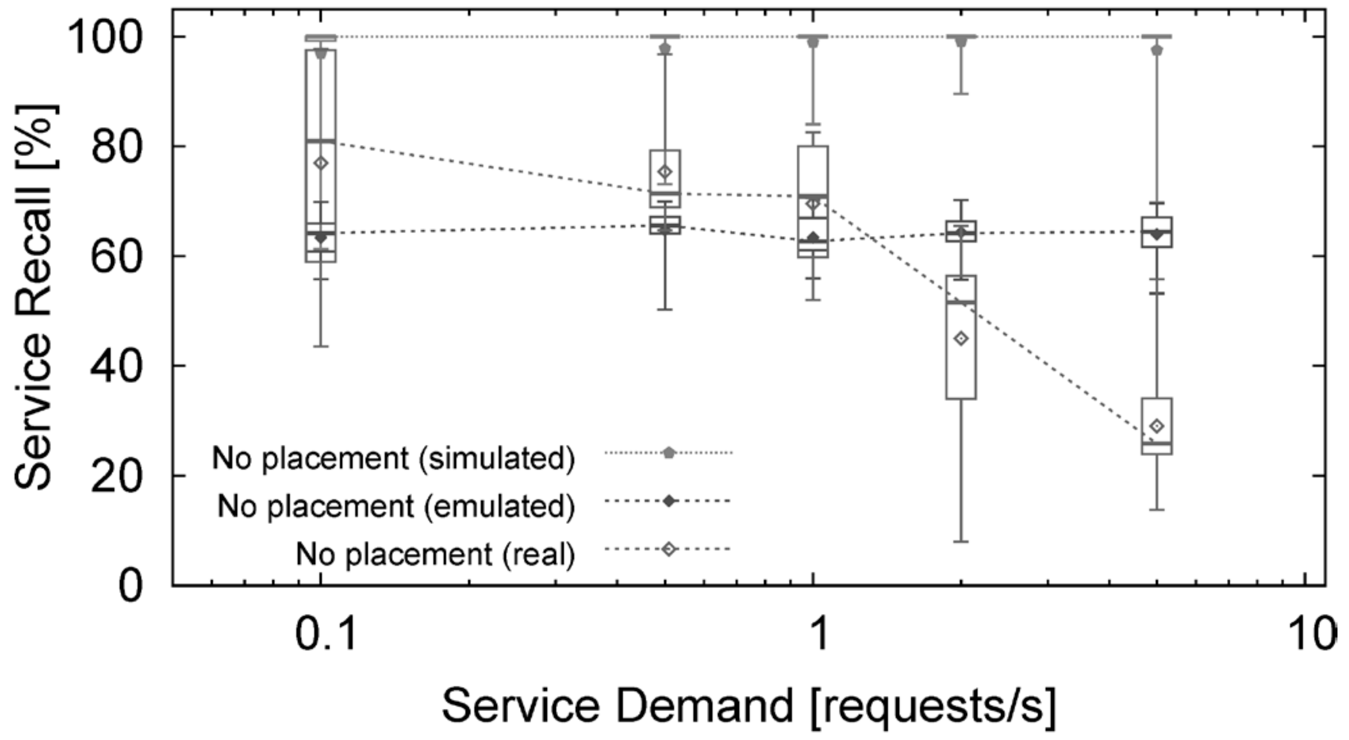


- *Don't assume that your data structures are up to date when handling individual packets.*
- *Checking for equality (or near equality) of time spells trouble!*
- *Have lots of sanity checks when using OS-specific heuristics to establish your address!*
- *Watch out for dropped broadcast packets! Know your broadcast domain!*
- *Run your code using debugging tools (e.g., gdb, valgrind, …) whenever possible!*
- *Don't rely on single packets; use link metrics!*
- *Use threshold values to translate between continuous link metric and "Boolean" link, e.g., ETX ≤ 2.0!*
- *Choose your link metric to satisfy assumptions higher-level components!*

- ***Try to look at platform issues and wireless issues separately!***

Results – Recall

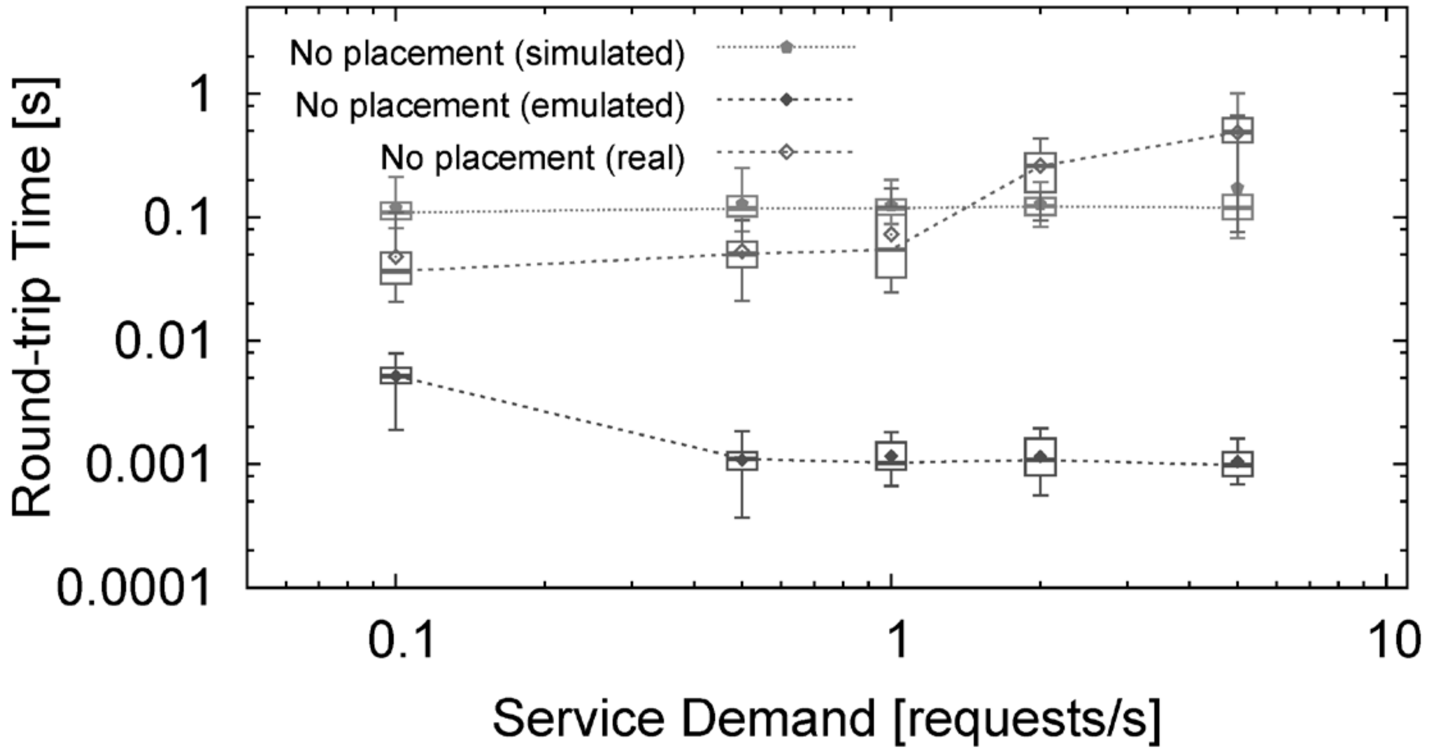
INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



- Simulation ► ***Model*** of wireless channel; no packet loss
- Emulation ► Per-packet ***model*** of transmission probability
- Real-world ► Subject to channel loss and interference

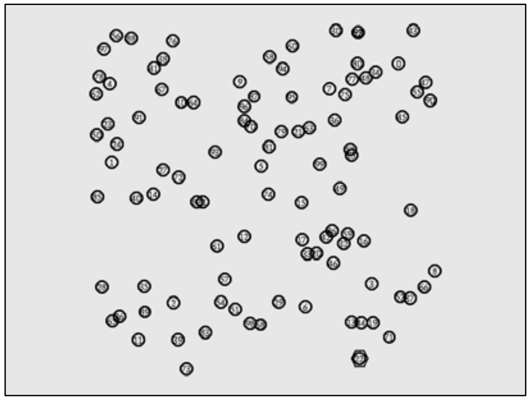
Results –Timing

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE INRIA



- Simulation ► ***Model*** of transmission (and processing) time
- Emulation ► Subject to wired transmission and processing
- Real-world ► Subject to wireless transmission and processing

Conclusion



- Separating platform and wireless issues is advantageous.
- Properties of employed models directly affect quantitative results.
- Simulation-based code can be successfully used in real deployments.