

Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

Universitätsstraße 38
D-70569 Stuttgart

NET - An Integrated Emulation Environment for Mobile Network Research

Andreas Grau, Klaus Herrmann, Kurt Rothermel

July 14th, 2011

NET Project

- **Network Emulation Testbed**
 - Founded by German Science Foundation
- **Goal**
 - Provide **flexible network infrastructure** for **comparative performance measurement** of **distributed applications** and **communication protocols**
 - e.g. internet-scale distributed peer-to-peer networks
 - e.g. mobile ad-hoc networks (MANETs)
- **Advantage**
 - **Flexibility** (real implementations)
 - **Reproducibility**
 - **Controllability** (arbitrary network conditions and topologies)
 - **Scalability**



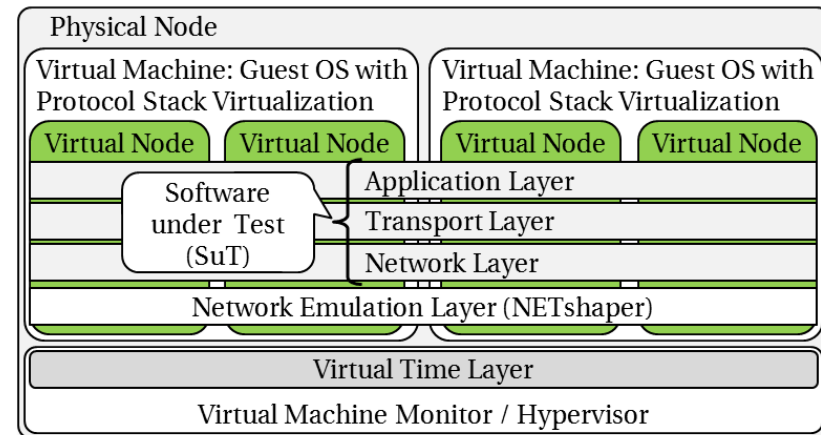
Outline

- Node and Time virtualization
- *NETshaper*: Emulation of Link Characteristics
- *NETplace*: Virtual Node Placement
- *NETbalance*: Dynamic Reconfiguration
- *NETcaptian*: Integrated Experiment Platform
- Summary



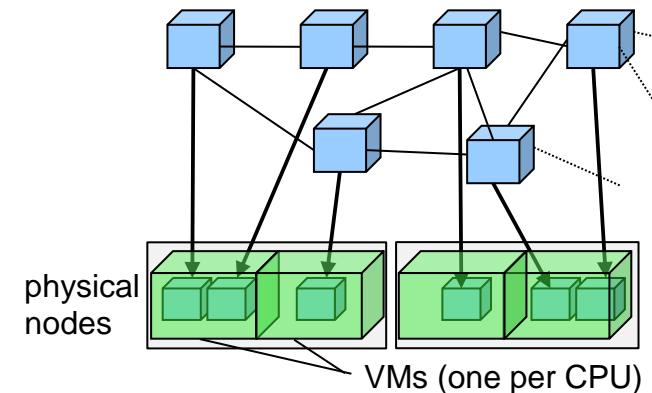
Node Virtualization

- Node virtualization (*PADS'08/SPECTS'10*)
 - Multiplex virtual nodes to physical nodes
- Lightweight virtualization using OpenVZ
 - Multicore support
 - Virtual protocol stacks
 - Sockets, routing table, network devices
 - Efficient communication, no context switches
 - Memory-efficient: one OS kernel, single caching system, library-sharing
 - Namespace partitioning
 - Virtualization of processes and file systems
 - Real node behavior: individual file systems, daemons
 - Support for common tools: ps, ping, traceroute, ...



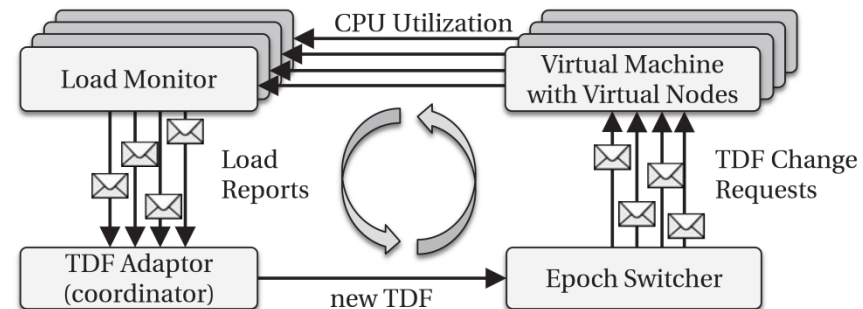
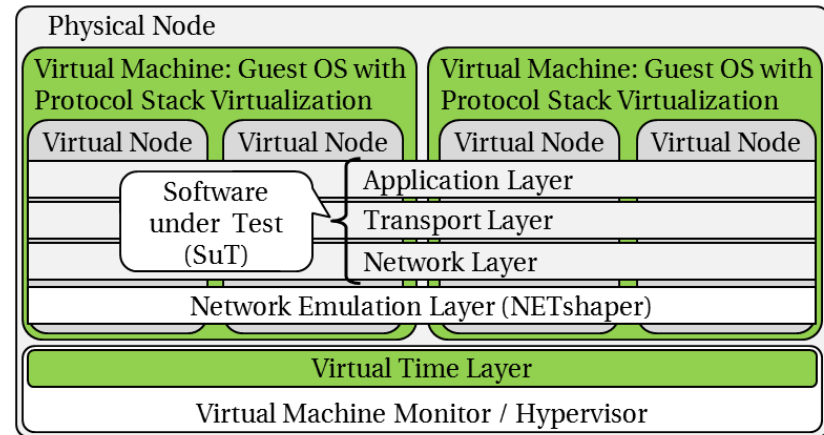
virtual nodes:

router, hosts running software under test



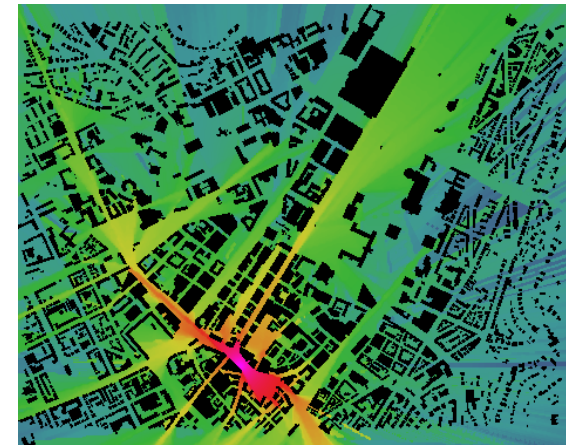
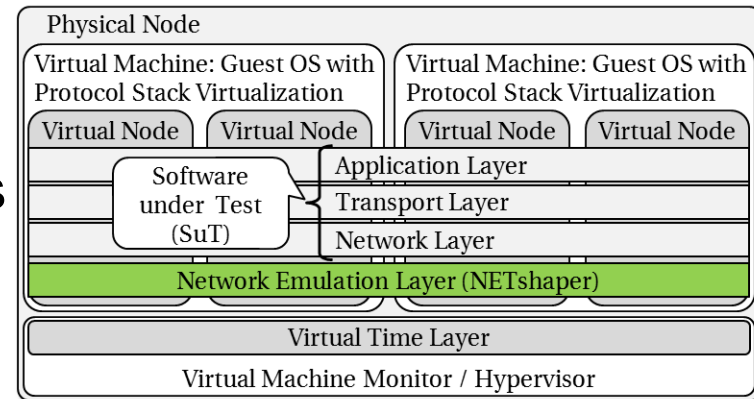
Time Virtualization

- Goal:
 - Avoid resource overload/underload
- Approach (*ICCCN'09*)
 - Virtual machine (XEN) provides virtual time transparently to the SuT
 - dynamically adapt virtual clock rates to current system load
 - ➔ Optimize experiment runtime while avoiding biased results
- Time virtualization example:
 - 1 Physical node (realtime)
 - 10 Nodes with 10MBit NICs
 - 1 Physical node (100 times slower)
 - 100 Nodes with 100MBit NICs



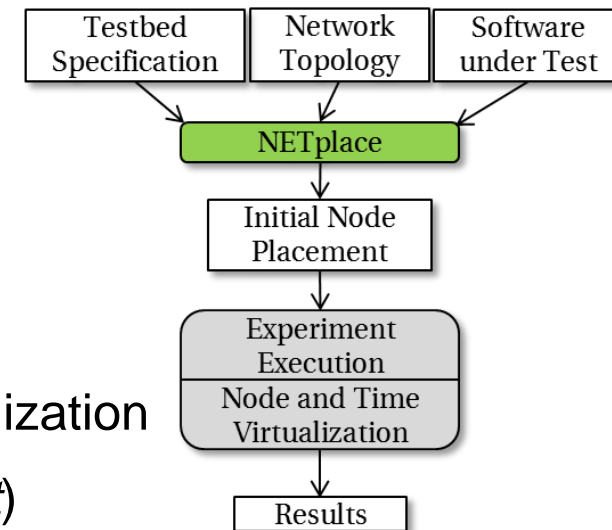
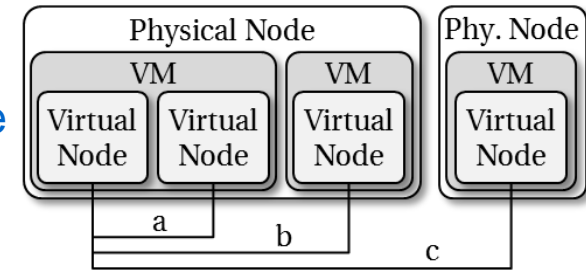
Network Emulation (NETshaper)

- Distributed emulation (*Comp. Com.'07*)
- Emulated network between virtual nodes
 - Arbitrary topologies
 - Add/remove links during runtime
 - Configurable link properties (bandwidth, delay, loss)
 - Dynamically adjust properties
 - Emulation of point-to-point links and shared media
 - Ray-tracing based radio propagation models
 - Adjust link properties based on node positions
 - Tracefile-based node mobility



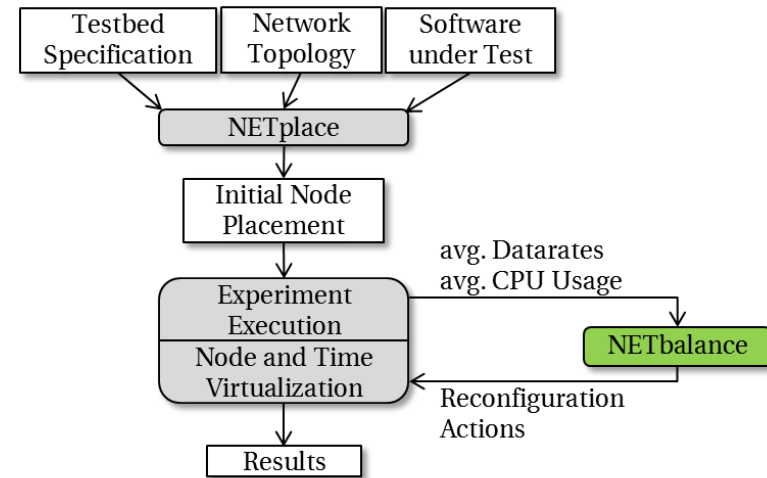
Initial Node Placement (NETplace)

- Goal: minimize experiment runtime
 - Minimize the load of the **maximum loaded phy. node**
 - Load of a physical node depends
 - Load generated by software under test
 - Emulation of virtual links
 - Different emulation costs for virtual links:
 - a) Intra-VM, b) Inter-VM, c) Inter-PNode (1:10:20)
- Approach (*SPECTS'10*)
 - Calculate **initial placement** of virtual nodes
 - Assumptions: known and constant load
 - Hierarchical graph partitioning + greedy optimization
 - Up to 60% runtime reduction (*k-way edge-cut*)



Dynamic Reconfiguration (NETbalance)

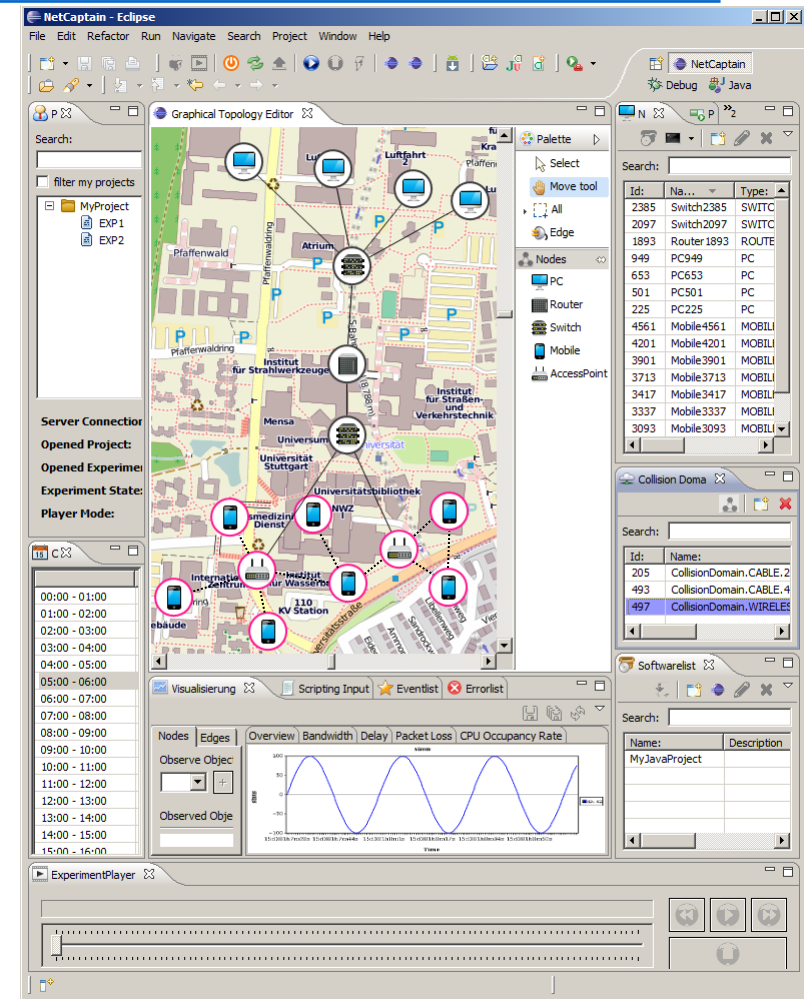
- Goal: minimize experiment runtime for scenarios with unknown or/and varying load
- Approach (ICCCN'11)
 - Monitor load and datarates
 - Calculate optimized placement p'
 - Iff $runtime(p') + T_{p \rightarrow p'} < runtime(p)$
 - Reconfigure emulation environment
 - Stop virtual time and migrate virtual nodes
- Transparent to software under test
- ➔ Up to 70% runtime reduction



	VM _{src}	VM _{dst}
Application Layer	SuT	→
Transport Layer	Sockets	→
Network Layer	Routing tables	→
Data Link Layer	NETshaper	→
Virtual Topology	SW-Bridges	→
	VLans	→

Integrated Experiment Platform (NETcaptian)

- Based on Eclipse framework
- Project/Experiment organization
- Graphical scenario visualization
 - Live topology view
 - Console access
 - Visualization (user defined data)
- Mobility support (NS2 format)
- Radio propagation models
 - Ray-tracing and distance-based
- Scripting engine (Javascript)
- Experiment player (replay experiment)



Summary

- Scalable infrastructure for comparative performance evaluation
 - Flexible, controllable and reproducible network experiments
 - Wired and wireless networks
 - Large network topologies
 - Node virtualization
 - Time virtualization
- Runtime minimization of experiments
 - Adaptive virtual time
 - Initial and dynamic node placement
- Efficient experiment execution
 - Integrated experiment platform



Thanks for your attention!



- Further information:
 - NET project homepage: <http://net.informatik.uni-stuttgart.de>



IPVS

Research Group
“Distributed Systems”

11

Universität Stuttgart
IPVS