

Contents

1	Introduction	1
1.1	Requirements for Evaluation Techniques	1
1.2	Performance Evaluation Techniques	2
1.2.1	Network Testbeds / Real-World Measurements	2
1.2.2	Network Simulators	3
1.2.3	Analytic Models	3
1.3	Hybrid Evaluation Methodologies	3
1.4	Challenges in Hybrid Evaluation	4
1.4.1	Synchronization and Integration of Timing Domains	4
1.4.2	Interoperability of Evaluation Techniques and Tools	5
1.4.3	Realistic and Contextual Integration for Hybrid Evaluation Frameworks	5
1.5	Research Questions	6
1.6	Contributions	6
1.7	Thesis Organization	8
2	Fundamentals and Background	9
2.1	An Introduction to the Performance Evaluation of Computer Networks	9
2.1.1	Performance Metrics	11
2.1.2	Performance Evaluation Methodologies	13
2.1.3	Hybrid Performance Evaluation Methodologies	25
2.1.4	A BitTorrent Model for Network Emulation	28
2.1.5	The Synchronization Problem of Network Emulation	32
2.2	Virtual Machines	34
2.2.1	Implementation Aspects of System Virtualization	35
2.2.2	The Xen Hypervisor	37

2.3	Hardware Simulation and Full-System Simulators	39
2.3.1	The Simics Full-System Simulator	40
2.3.2	SystemC	41
2.4	Interim Summary	42
3	Synchronized Hybrid Evaluation	43
3.1	Time Synchronization	45
3.1.1	Execution of System Representations: Assumptions	45
3.1.2	Synchronization Requirements	46
3.1.3	The Synchronization Scheme	47
3.1.4	The Synchronization Protocol	51
3.2	System Representation Components	53
3.2.1	Characteristics of System Representations	53
3.2.2	Virtual Machines (VMs)	54
3.2.3	Discrete Event-based Network Simulation	56
3.2.4	Full-System Simulators	58
3.2.5	Other System Representation Types	59
3.2.6	Hybrid System Representations	59
3.3	Packet Exchange between System Representations	60
3.3.1	Non-Synchronized Packet Exchange	60
3.3.2	Integration at the Medium Access Control (MAC) layer	61
3.4	Types of Synchronized Hybrid Evaluation setups	62
3.4.1	Synchronized Network Emulation	62
3.4.2	Synchronized Hardware/Network Co-Simulation	63
3.4.3	Other Types of Compositions	64
3.5	Interim Summary	66
4	Synchronized Emulation Frameworks	67
4.1	SliceTime: A Platform for Scalable and Accurate Network Emulation	68
4.1.1	Conceptual Design	68
4.1.2	Implementation of SliceTime	69
4.1.3	Evaluation of SliceTime	75
4.1.4	Interim Summary	80

4.2	Device Driver-enabled Wireless Network Emulation	80
4.2.1	DDWNE Architecture	81
4.2.2	Implementation of DDWNE for 802.11	85
4.2.3	Evaluation	89
4.2.4	Compatibility with unmodified Wireless Software	92
4.2.5	Interim Summary	94
4.3	Application Studies using SliceTime and DDWNE	94
4.3.1	HTTP Use Case	94
4.3.2	AODV routing daemon study	96
4.3.3	A Hybrid Evaluation Platform for BitTorrent systems	98
4.4	Related Work	104
4.4.1	Link Emulation Engines	105
4.4.2	Virtualization-based Emulation Engines	108
4.4.3	Network Simulator-based Emulation Engines	114
4.4.4	Hardware-based Network Emulation	122
4.5	Interim Conclusion	127
5	A hybrid evaluation framework for networked embedded systems	129
5.1	Motivation	129
5.2	Conceptual Design	130
5.2.1	Synchronization Component	130
5.2.2	Network Simulation	131
5.2.3	Virtual Platform (VP)	131
5.3	Implementation	132
5.3.1	The SyncVP Component	133
5.3.2	Virtual Network Chip	133
5.4	Evaluation	134
5.4.1	Virtual Platform Performance	135
5.4.2	Influence of Synchronization Accuracy	135
5.4.3	Influences of Network Traffic on the VP Bus Load	137
5.4.4	Influence of Payload Processing on Round Trip Time	137
5.5	Related Work	138
5.5.1	Hardware Simulator-based Approaches	139

5.5.2	Approaches based on Co-Simulation	142
5.6	Limitations	144
5.6.1	Non-Optimal Integration of Timing	144
5.6.2	Dependency on Proprietary Software	145
5.7	Interim Summary and Conclusion	145
6	Monitoring and Debugging Communication Software using Virtual Time	147
6.1	Motivation	147
6.2	A Distributed Analysis Framework	148
6.2.1	Challenges and Solutions	148
6.2.2	Architecture	149
6.3	Prototype Implementation	151
6.3.1	Scripting Environment	152
6.3.2	Back-end Server	152
6.4	Functional Tests	153
6.4.1	Time Isolation	153
6.4.2	Monitoring State Changes	154
6.5	Related Work	154
6.5.1	Virtual Machine-based Debugging	154
6.5.2	Distributed Debugging	155
6.5.3	Online System Monitoring	156
6.6	Limitations and Potential Future Work	157
6.7	Interim Summary and Conclusion	158
7	Conclusion	159
7.1	Key Results	160
7.2	Conceptual Limitations	162
7.3	Future Research and Possible Extensions	163
7.4	Final Remarks	166

A	Appendix	185
A.1	SliceTime: Additional Timing Measurements	186
A.2	BitTorrent Client Model Implementation	187
A.2.1	Component Interaction	187
A.3	Setup and Application of DDWNE	188
A.3.1	Setting up DDWNE for ns-3	188
A.3.2	Setup of the DDWNE Driver	188
A.4	List of Abbreviations	189