NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INFRASTRUCTURE SUITABILITY ASSESSMENT MODELING FOR CLOUD COMPUTING SOLUTIONS

by

Carsten Fehse

September 2011

Thesis Advisor:	Man-Tak Shing
Thesis Co-Advisor:	Albert Barreto III

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** 23 Sep 2011	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Infrastructure Suitability Assessment Modeling for Cloud Computing Solutions			**5. FUNDING NUMBERS**
6. AUTHOR(S) Fehse, Carsten			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approve for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Maturing virtualization in information technology systems has enabled increased implementations of the cloud computing paradigm, dissolving the need to co-locate user and computing power by providing desired services through the network. This thesis researches the support that current network modeling and simulation applications can provide to IT projects in planning, implementing and maintaining networks for cloud solutions. A problem-appropriate domain model and subsequent requirements are developed for the assessment of several network modeling and simulation tools, which leads to the identification of a capability gap precluding the use of such tools in early stages of cloud computing projects. Consequently, a practical, modular designed methodology is proposed to measure the essential properties necessary for developing appropriate cloud computing network traffic models. The conducted proof-of-concept experiment applied to a virtual desktop environment finds the proposed methodology suitable and problem-appropriate, and results in recommended steps to close the identified capability gap.

14. SUBJECT TERMS Cloud Computing, Modeling, Simulation, Virtualization, Virtual Desktop Infrastructure, Network Emulator			**15. NUMBER OF PAGES** 103
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INFRASTRUCTURE SUITABILITY ASSESSMENT MODELING FOR CLOUD COMPUTING SOLUTIONS

Fehse, Carsten
Commander, German Navy
E.E., Helmut Schmidt University, 1992

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
INFORMATION TECHNOLOGY MANAGEMENT**

**MASTER OF SCIENCE IN
SOFTWARE ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL
September 2011**

Author: Carsten Fehse

Approved by: Man-Tak Shing
Thesis Advisor

Albert Barreto III
Thesis Co-Advisor

Dan C. Boger
Chair, Department of Information Science

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Maturing virtualization in information technology systems has enabled increased implementations of the cloud computing paradigm, dissolving the need to co-locate user and computing power by providing desired services through the network. This thesis researches the support that current network modeling and simulation applications can provide to IT projects in planning, implementing and maintaining networks for cloud solutions. A problem-appropriate domain model and subsequent requirements are developed for the assessment of several network modeling and simulation tools, which leads to the identification of a capability gap precluding the use of such tools in early stages of cloud computing projects. Consequently, a practical, modular designed methodology is proposed to measure the essential properties necessary for developing appropriate cloud computing network traffic models. The conducted proof-of-concept experiment applied to a virtual desktop environment finds the proposed methodology suitable and problem-appropriate, and results in recommended steps to close the identified capability gap.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
- **A. METHODOLOGY 2**
- **B. STRUCTURE 2**

II. DEFINITIONS AND TERMINOLOGY 5
- **A. CLOUD COMPUTING 5**
  - **1. The Cloud 5**
  - **2. Cloud Service Models 6**
  - **3. Cloud Deployment Models 8**
- **B. VIRTUALIZATION 9**
  - **1. Types of Virtualization 9**
  - **2. XEN 10**
  - **3. Hyper-V 11**
  - **4. VMware vSphere 11**
- **C. NETWORK CHARACTERISTICS, PARAMETER 11**
  - **1. Bandwidth 11**
  - **2. Latency 12**
- **D. MODELING AND SIMULATION 13**
  - **1. Simulation 13**
  - **2. Modeling 13**
  - **3. Discrete-Event Simulation Models 14**
- **E. COMMON PROTOCOLS UTILIZED IN VDI IMPLEMENTATIONS 15**
  - **1. Remote Desktop Protocol - RDP 15**
  - **2. High Definition user eXperience – HDX™ 16**
  - **3. User eXtension Protocol – UXP 17**
  - **4. Personal Computer over Internet Protocol – PCoIP 19**
  - **5. Summary VDI protocols 19**

III. EVALUATION OF CURRENT M&S TOOLS 23
- **A. INITIAL SET OF GOALS / VISION 23**
- **B. USE CASES / EVALUATION METRICS 28**
- **C. EVALUATION OF EXEMPLARY TOOLS 35**
  - **1. Project 35**
  - **2. Scenarios 37**
  - **3. ns-3 ..... 40**
    - ***a. Description 40***
    - ***b. Assessment 43***
  - **4. NetSim (Academic / Standard) 44**
    - ***a. Description 44***
    - ***b. Assessment 46***
  - **5. OpNet ITGuru 48**
    - ***a. Description 48***
    - ***b. Assessment 49***

6. Summary 52
IV. EXPERIMENT 55
A. GOALS OF THE EXPERIMENT 55
B. METHODOLOGY 55
1. User Environment 56
2. Usability Metrics 57
C. USER LOAD SCENARIOS 59
1. Basic Task Set 59
2. Complex Task Set 60
3. Multimedia Task Set 60
D. EXPERIMENT SETUP 61
1. Virtual desktop infrastructure 61
2. Network Bridge – Bandwidth and Latency Emulator 63
3. Network Recording and Analysis 65
E. MEASUREMENTS 66
F. ANALYSIS 67
G. SUMMARY 70
V. CONCLUSION, FURTHER RESEARCH AND RECOMMENDATIONS 73
A. CONCLUSION 73
B. FURTHER RESEARCH AND RECOMMENDATIONS 73
1. Network Emulator Virtual Machine 73
2. User Experience Categories 74
3. Standardized Task Sets 75
4. Development of Models for Simulation 76
APPENDIX 77
A. EXPERIMENT RESULTS “LAPTOP” 77
B. EXPERIMENT RESULTS “ZERO CLIENT” 78
C. EXPERIMENT RESULTS “NETTOP” 79
D. EXPERIMENT RESULTS “MOBILE” 80
LIST OF REFERENCES 81
INITIAL DISTRIBUTION LIST 85

LIST OF FIGURES

Figure 1. Level of control and management responsibility for cloud service models 7
Figure 2. Cloud Deployment Models 8
Figure 3. Full Virtualization vs. Para-Virtualization 10
Figure 4. Modeling alternatives in relation to model fidelity and model size (number of nodes). After [33] 14
Figure 5. Citrix HDX components/architecture. After [40]. 17
Figure 6. NComputing UXP deployment architecture on virtualized server infrastructure. After [45]. 18
Figure 7. Domain model – VDI aspects 25
Figure 8. Domain model – network aspects 26
Figure 9. Domain model – complete 27
Figure 10. Naval Postgraduate School Campus – buildings colored according to their main functional role: academia, administration, general services, housing (From: NPS Intranet) 36
Figure 11. Typical sites from which NPS distant learning students participate 37
Figure 12. VDI deployment scenarios for hypothetical project 40
Figure 13. VDI infrastructure schematic 62
Figure 14. VMware vSphere client snapshot 63
Figure 15. "dummynet" pipe principle 64
Figure 16. Experiment setup schematic - network simulator, analysis components, VDI client/server connection 66
Figure 17. Generic graph test results 67
Figure 18. Virtualized experiment setup 71

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Supported environments for VDI implementations 21
Table 2. Metrics table, aspects under evaluation, value range 31
Table 3. Qualitative assessment results 52
Table 4. VDI client configurations for the experiment 57
Table 5. User experience categories 59

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ARM	Advanced RISC Machines, multinational semiconductor and software company, Cambridge, United Kingdom
CIO	Chief Information Officer
CONUS	Continental USA
DoD	Department of Defense
ERN	Education and Research Network
HDX	High Definition user eXperience, a proprietary, purpose built protocol utilized for VDI (Citrix®)
IaaS	Infrastructure as a Service (cloud service model)
IANA	Internet Assigned Numbers Authority
ICA	Intelligent Console Architecture
ICATS	Integrated Cloud Application Tool Suite
IP	Internet Protocol
LAN	Local Area Network
NIST	National Institute of Standards and Technology
NLB	Network Load Balancing
OCONUS	Outside Continental USA
PaaS	Platform as a Service (cloud service model)
PCoIP	Personal Computer over Internet Protocol, a proprietary, purpose built protocol utilized for VDI (VMware®)
RAID	Redundant Array of Independent (formerly: Inexpensive) Disks
RDP	Remote Desktop Protocol, a proprietary, purpose built protocol utilized for VDI (Microsoft®)
RISC	Reduced Instruction Set Computing
SaaS	Software as a Service (cloud service model)
SLA	Service Level Agreement
SoC	System on a Chip
TCP	Transfer Control Protocol
UDP	User Datagram Protocol
USEUCOM	U.S. European Command (area of operation)
USPACOM	U.S. Pacific Command (area of operation)

UXP	User eXtension Protocol, a proprietary, purpose built protocol utilized for VDI (Ncomputing®)
VDI	Virtual Desktop Infrastructure
VPN	Virtual Private Network
VTC	Video Tele-Conference
WAN	Wide Area Network

ACKNOWLEDGMENTS

I would like to thank everyone who helped my family and me during our time here in Monterey; it was a great pleasure to experience such amazing hospitality, support, and friendship – I hope that sometime in the future I will have the opportunity to return such kindness when visited in Germany.

My sincere thanks and appreciation go to my advisors, Dr. Man-Tak Shing and Albert “Buddy” Barreto III, whose patience, trust, and faith allowed me to overcome all obstacles and finish this thesis. It was an extraordinary privilege working with you.

Finally, I thank my family for their steadfast support; Freia, Jan, and Tobias – you are my love and life.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Since its introduction a few years ago, the paradigm of “cloud computing” is gaining momentum, in industry as well as the private sector. The underlying concepts of virtualization, abstracting physical computing resources into multiple logical resources, now find support in the complete range of information technology, from specific computer hardware extensions to management applications for virtualized operating systems and applications. The promise of cost-effectiveness, flexibility in provisioning, ease in management, and energy savings, to name just a few, have encouraged the effort “to go cloud.”

The Federal Chief Information Officer mandated the shift to a “cloud first” policy in his 25-point implementation plan to reform federal IT [1]. Within the Department of Defense (DoD), the “804-Report” to Congress [2] and the corresponding industry perspective [3] underline the desire to derive IT solutions that leverage commercial state-of-the-art products and services to satisfy future information technology needs within the existing and increasing budgetary constraints. The industry-proposed and federally supported move to IT portfolios that managers can use to derive IT solutions to new problems by combining modular capabilities is intriguing. However, it requires building a versatile portfolio based on best-practices, industry available IT components, and open standards.

Currently, numerous vendors offer services and solutions for cloud computing that allow the consolidation of data centers, centralized administration, provisioning, and so forth. Projects attempting to implement cloud based solutions can choose between a variety of solutions, often combined packages of software and hardware that have been optimized for the specific purpose. However, all of these solutions require certain performance parameters to enable networks that will connect the user and the cloud service in a manner that realizes the service’s full potential. For projects that include the design and implementation of such networks, best practices and support are available to ensure proper functionality and performance of the entire implementation. On the other hand, projects that have to leverage from existing network infrastructure or only have limited

influence on the connecting network (e.g., because the network is managed by another entity or the budget does not allow extension of the project to it), face risks that should be mitigated as early as possible.

One suitable option to assess risks and design options prior to managerial decisions in the early phases of a project is modeling and simulation (M&S). Using this general tool can help avoid unnecessary vendor dependence or "architectural lock-in." This thesis explores how current network modeling and simulation tools can support the early phases of projects seeking cloud computing implementations.

A. METHODOLOGY

Both main parts of this thesis (the structured capability assessment in Chapter III and the experimental proof-of-concept in Chapter IV) follow well-structured methodologies.

For the assessment, selected activities from the initial phases of the Unified Process are utilized [4], [5]. First, a problem-appropriate domain model for virtual desktop infrastructures is developed. Then, based on a set of problem-oriented use cases, a list of functional and non-functional requirements is derived and then used to assess the selected network modeling and simulation tools.

The experimental part identifies the dependent and independent variables for the problem at hand. It also designs a methodology based on practical experiments to gain information on the influence of network bandwidth and latency on virtual desktop infrastructure implementations under controlled conditions. The executed proof-of-concept experiment confirms the suitability of the proposed methodology and identifies areas for further research and refinement.

B. STRUCTURE

Following the current chapter's primer on the subject, Chapter II will introduce some important terms and concepts from the fields of cloud computing and computer networking. The different service types and deployment options for clouds are briefly described and compared. Then, the concept of virtualization is introduced and different

dominant implementations of the concept are concisely contrasted. Focusing on the subject of this thesis, essential modeling and simulation concepts are presented before the chapter closes out with a portrayal of four dominant virtual desktop infrastructure network protocols.

Chapter III will develop a problem-appropriate domain model for virtual desktop infrastructures and derive a set of requirements, which are then used to assess the suitability of selected network modeling and simulation tools for a hypothetical but realistic project. The requirements are used to assess the support of planning, designing, implementing, and maintaining virtual desktop infrastructures for the project.

As a result of the above assessment, Chapter IV proposes a practical, modular methodology to collect the essential properties of virtual desktop infrastructure network protocols to allow development of appropriate traffic models for simulations. The chapter continues by describing the proof-of-concept experiment conducted with a fully functional virtualization environment created for this thesis. The results of the experiment are analyzed and respective conclusions are derived.

Chapter V summarizes the findings of this thesis and suggests several practical areas for further research that will allow closing the identified capability gap, better enabling modeling and simulation tools to support the early stages of cloud computing projects.

THIS PAGE INTENTIONALLY LEFT BLANK

II. DEFINITIONS AND TERMINOLOGY

This chapter presents terms and definitions in the field of cloud computing and computer networks that will be used throughout this thesis.

The chapter is organized as follows. First, general terminology and concepts with respect to cloud computing and virtualization are presented, briefly covering the various options currently offered by vendors. Second, “bandwidth” and “latency” are introduced as characteristics of cloud-enabling computer networks. Third, the cloud-enabling concept of virtualization is sketched. And finally, major protocols used in virtual desktop infrastructure solutions are described in their main characteristics with respect to the subject of this thesis.

A. CLOUD COMPUTING

The term “cloud computing” carries a wide range of interpretations, depending on the user and the context. It is sometimes discredited as “nothing new” (Larry Ellison, CEO of Oracle, in [6]) and a “marketing hype campaign” (Richard Stallman, Free Software Foundation, in [7]). It is also sometimes offered by vendors as the ‘silver bullet” to all problems facing IT, according to the informational material available for their products.

These rather biased opinions and interpretations of cloud computing do not allow for a structured approach to the emerging field. A more suitable approach is offered by The National Institute of Standards and Technology (NIST) in its “NIST Cloud Computing Program” resources. This work was compiled from collaborative activities with contributors from the federal, industrial, and academic sectors, in an attempt to issue a special publication series for the subject: “Cloud Computing Reference Architecture” [8].

1. The Cloud

In its draft document regarding the definition of cloud computing, NIST states:

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing re-

> sources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models. [9]

Along the lines of this paradigm, the cloud itself can be viewed as a pool of computing resources and services that can be easily accessed by the end-user, who is not required to have information on the configuration or physical location of the resources. This information can remain invisible, "in the cloud."

NIST's Cloud Computing Reference Architecture, as well as other reference models [10] [11] [12], provide an abstraction of cloud computing concepts and their relationships that can be used to create standards and guidelines for their application. The models also support informed decisions for the adoption of cloud technologies. For the scope of this thesis, the architectural models for cloud service and cloud deployments will be introduced briefly in the following sections.

2. Cloud Service Models

The services provided by cloud computing implementations can be categorized into three models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). These service models are distinguished by the services that are provided by the cloud service provider to the cloud service consumer, and to which degree the cloud service consumer has management control over components of the cloud infrastructure.

SaaS allows the cloud consumer to use the provider's applications, which are executed on the cloud infrastructure through various technologies including thin clients and web browsers. The consumer does not manage or control any of the underlying cloud infrastructure components. Current examples for SaaS implementations are Google Docs [13] and Microsoft Office Web Apps [14] (part of Microsoft's Office 365 software plus service offerings), both providing office applications through web browsers.

PaaS allows the consumer to use self-developed or acquired applications, which are executed on the provider's cloud infrastructure. The consumer manages and controls

the applications and selects hosting environment configurations, but has no control over the rest of the underlying cloud infrastructure components. Typical representatives of such service offers are Amazon's Elastic Compute Cloud (EC2) [15] and Google's App Engine (GAE) [16], delivering platforms in the cloud for the cloud service consumer's applications.

IaaS allows the consumer to provision fundamental computing resources of the cloud service provider to deploy and manage applications, operating systems, and selected platform configurations. Amazon Web Services – AWS [17] and Flexiant Ltd's FlexiScale [18] are examples of IaaS offers.

Figure 1 depicts the three cloud service models and the respective control and management responsibility for the different cloud infrastructure components.

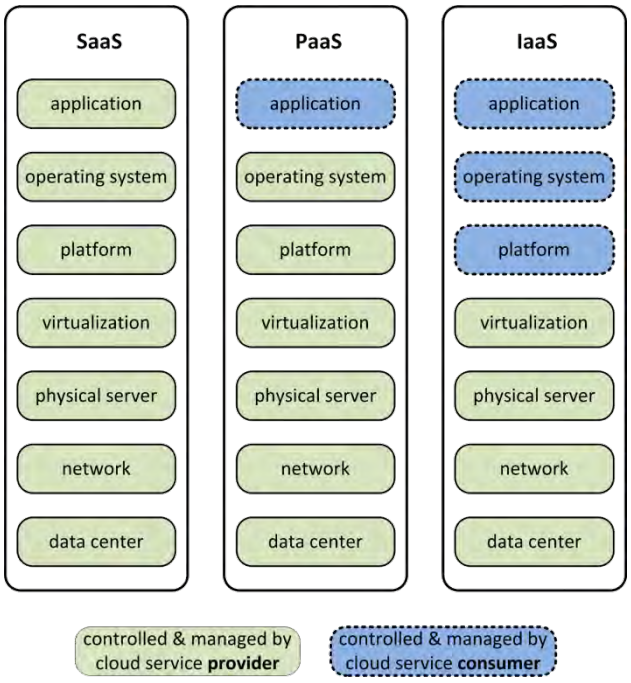


Figure 1. Level of control and management responsibility for cloud service models

3. Cloud Deployment Models

The four cloud deployment models capture the various options for service access and operation mode. NIST distinguishes between Private Cloud, Public Cloud, Community Cloud, and Hybrid Cloud.

Cloud infrastructures operated exclusively for one organization are considered Private Clouds, regardless of: whether the organization itself or a third party is responsible for the operation, and where the infrastructure is located (on premise or off premise). In comparison, Public Clouds are accessible by the general public or several organizations; the infrastructure for this deployment model is usually operated by cloud service providers on their own premises (data-centers). In cases where the cloud infrastructure is available for a specific community of interest consisting of separate organizations and limited to this community, the services are considered as Community Cloud. Finally, in cases where the cloud infrastructure is a composition of two or more clouds that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability, it is considered a Hybrid Cloud. Figure 2 depicts the four cloud deployment models.

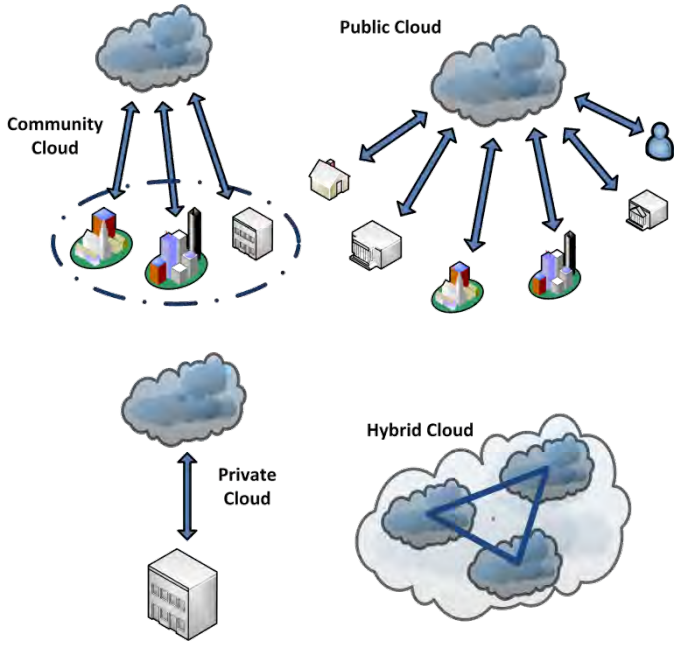


Figure 2. Cloud Deployment Models

It is important to note that the different cloud deployment models do not necessarily define the location where the respective infrastructure is installed. The deployment model is part of the respective service level agreement between cloud service provider and cloud service consumer.

B. VIRTUALIZATION

The cloud computing paradigm is based on several enabling key technologies, such as the widespread availability of fast computer networks, inexpensive computing power provided by small-form factor servers and, most importantly, high-performance virtualization for commodity hardware.

"Virtualization technologies encompass a variety of mechanisms and techniques used to decouple the architecture and user-perceived behavior of hardware and software resources from their physical implementation" [19]. Prominent technologies include virtual local area networks (VLAN), establishing a separation in local network architecture on top of the physical structure, and virtual file systems, providing unified view and access to storage, which is in fact physically distributed. However, most important for cloud computing are virtual machine monitors (VMM) or hypervisors, providing a layer between software environments and physical hardware, and thereby allowing single instances of hardware to execute multiple instances of software environments simultaneously. Formerly limited to mainframe systems, current commodity processor hardware [20], [21], [22] includes support for virtualization, increasing the efficiency and performance of virtualization technology implementations.

1. Types of Virtualization

This research focuses on virtual desktop infrastructures, and consequently the major implementations of virtual machine monitors / hypervisors are briefly introduced. The two major types of virtualization are full virtualization and para-virtualization, depending on how VMMs / hypervisors are executed on "host machines" and provide virtualized "guest machines" on top of the real hardware environment. In full virtualization, the hypervisor provides the complete simulation of the actual hardware, thus allowing execution of unmodified guest operating systems. This can be achieved by either on-the-fly bi-

nary translation of the non-virtualizable code of the guest operating system or by hardware support[1] that allows the hypervisor to be executed in a privilege level below the guest OS.

In the case of partial virtualization, which abstracts some but not all aspects of the host environment, modifications to the software being executed on the guest machine may be required. In the case of para-virtualization, the software on the guest machine is executed in an isolated manner, as if it is being executed on separate systems; however, it also needs to be modified to be executed in the para-virtualized environment. A more detailed discussion and comparison of virtualization techniques for x86 processor architectures, including memory, device and I/O virtualization can be found in [23]. Figure 3 shows the conceptual difference between full virtualization and para-virtualization.

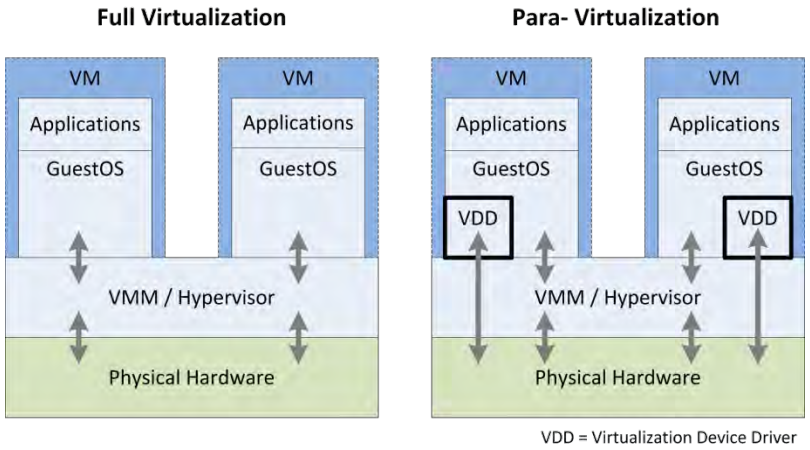


Figure 3. Full Virtualization vs. Para-Virtualization

2. XEN

The open source project Xen, strongly supported by Citrix and other IT enterprises, maintains the Xen hypervisor, targeted at the enterprise level server virtualization [24]. Xen provides support for a wide range of operating systems as guests, including Microsoft Windows, Linux, Solaris, and various versions of BSD. The Xen hypervisor provides para-virtualization for modified operating systems (Xen-specific kernel modules for Linux, Solaris, FreeBSD, and other UNIX operating systems), as well as full virtual-

[1] For example in the case of x86 architectures AMD's 'AMD-V' or Intel's 'VT-x'.

ization for all other, unchanged operating systems like Microsoft Windows. However, only operating systems running as para-virtualized guests achieve full performance. The ports of Windows operating systems are the subject of research, however, due to intellectual property limitations, they are not available for productive environments.

3. Hyper-V

Microsoft's Hyper-V Server 2008 is a dedicated stand-alone product, tailored to support Windows operating systems at optimal performance [25]. Windows guests systems of the Hyper-V host can leverage full driver support and provide seamless integration in Microsoft's product portfolio. However, para-virtualization support for operating systems other than Windows is limited to the SUSE Linux Enterprise Server; all other operating systems require full virtualization and therefore achieve only limited performance.

4. VMware vSphere

VMware's vSphere (formerly ESXi) supports a broad range of operating systems, both open source and commercial, by providing driver support for para-virtualization, thus leveraging increased performance in comparison to full virtualization [26]. In addition to the broad range of provided para-virtual drivers, VMware handles the main memory assignments to the virtual machines dynamically during run-time, further increasing flexibility in the provisioning of resources.

C. NETWORK CHARACTERISTICS, PARAMETER

1. Bandwidth

In telecommunication engineering, the term "bandwidth" is defined as the "difference between the limiting (upper and lower) frequencies of a continuous frequency spectrum" [27]. This original meaning was characteristic of data channels, measured in hertz [Hz], and mainly used for (wireless, digital, or analog) communication channels when referring to "signal bandwidth" [28]. However, in computer networking, the term is commonly used to refer to "channel capacity," "data rate," or "throughput rate," and is measured in bits per second [bit/s] to express the amount of data that can be transferred in

a given time through a network connection [29]. Physical and logical implementation details define the relation between the available frequency bandwidth and the resulting data bandwidth for any given communication channel.

In the computer network context of this thesis, the term “bandwidth” will be used synonymously with “channel capacity,” unless noted otherwise.

Closely related to bandwidth in computer networking are the terms “throughput” or “goodput,” expressing the average amount of (net) data transferred per time unit. It must be noted that the “goodput” in a network is always lower than the theoretical (gross) bandwidth, since protocol overheads, error correction schemata, connection management information, and so forth, add to the amount of data that has to be communicated over the respective channel to transfer the usable data itself properly.

2. Latency

“Latency” in computer networking refers to any of the delays that occur when data is sent from source to destination. In packet switched networks, this travel time consists of two categories: transfer latency (the duration of the data package being transferred over a specific medium[2]) and processing latency (the time consumed by active network components[3] to process the data package before forwarding it).

The latency of a network can be either measured one-way, from source to destination, or two-way (round-trip), from source to destination and back to source (usually excluding the processing time at the destination to generate the response). From the perspective of a user in a virtual desktop implementation, the round-trip latency is of more importance, as the user interacts with the host of the virtualized desktop through the network and expects this interaction not to be hindered by unacceptable delays between their own action and the reaction of the system that would render the implementation unusable.

Latency in non-trivial computer networks is by no means a constant throughout operation of the network. The configuration of active components, changes in configura-

[2] Copper cable, fiber optics, wire-less (including satellite connections).

[3] Active network components in packet switched, non-trivial networks include routers, gateways, crypto-devices, firewalls, intrusion detection systems, network interface cards, etc.

tion, rule-based load-balancing, and so forth, can have a strong influence on the delay that packets experience on their way from sender to receiver beyond the level of statistical variance ("jitter"). Within local networks, the network administration authority inside an organization usually has access to and influence on the many variables; but once the connection includes inter-network connections at the wide-area level, connecting sites through telecommunication service providers (commercial as well as governmental), the details of such connections often become inaccessible beyond the contracted parameters of the service level agreement.

D. MODELING AND SIMULATION

1. Simulation

According to Shannon, simulation is "the process of designing a model of a real system and conducting experiments with this model for the purpose of either understanding the behavior of the system or of evaluating various strategies […] for the operation of the system" [30].

Reasons to employ simulations in engineering are manifold; some of the most important motivations are:

a) Assessing options during design and development with regard to feasibility, risk-mitigation, and optimization prior to realization of the not (yet) existing system.

b) Inaccessibility of the real system under analysis (e.g., if it is under continuous use, or outside the reach of the analyst, for example, in fielded space systems).

c) The replication of the system under analysis is cost-prohibitive or impossible due to its size or complexity (e.g., the Internet).

Various techniques for simulation exist and depend on the type of model that is underlying the simulation process.

2. Modeling

Modeling is the activity of creating a model as the (simplified) representation of an actual system (existing, under development, or envisioned), capturing the essential

properties of interest. Several classes of useful models in engineering can be categorized, including: physical, analog, and schematic models [31]. Of primary interest in the context of network simulation are (executable) models build in software, representing the behavior of computer networks for analysis. These models can be of several types including mathematical, logical, statistical, or discrete-event models. The first three types represent the model by mathematical or statistical relationships, whereas discrete-event type models "attempt to represent the components of a system and their interactions to such an extent that the objectives of the study are met" [32].

The different modeling alternatives available for network simulation vary in their scalability and fidelity (see Figure 4). The network simulation tools assessed in this research all belong the category of packet level simulations – specifically, "discrete-event simulation models."

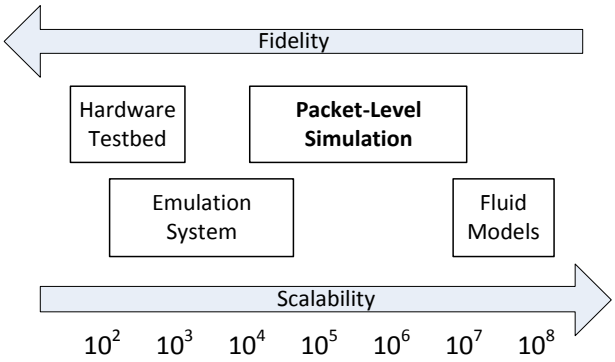


Figure 4. Modeling alternatives in relation to model fidelity and model size (number of nodes). After [33]

3. Discrete-Event Simulation Models

Some general concepts and structural components utilized in discrete-event simulation models are [34]:

"System State Variables" capture all necessary information to characterize a system to an adequate level at any time. In discrete-event models these variables change only at well defined (event) times.

"Entities and Attributes" represent every component or object and detail of the system to be modeled. The purpose of the model, and subsequently the simulation, determines the necessary attributes for each entity and required fidelity.

"Activities, Delays and Events" – activities represent actions performed during simulations. Events mark the beginning and the end of such actions (determined by the simulation), and delays represent an unknown duration of time whose end is triggered, usually by an event.

In brief, a discrete-event simulation model can be described as one in which the system state is represented by state variables that change only at the specific points in time at which events occur as a result of activities, their respective activity time, and delays. The entities of the model and their attributes determine the activities that are executed within the model over the progressing simulated time. Observation and collection of statistics on certain state variables throughout the simulation time allows for the derivation of answers to the questions under analysis [32], [35].

E. COMMON PROTOCOLS UTILIZED IN VDI IMPLEMENTATIONS

The protocols utilized in virtual desktop infrastructure implementation all have in common providing services (mostly the graphical user interface) to the user by connecting through the network with the virtual desktop that resides on the (usually locally) separated host, replicating a traditional local installation as closely as possible. The user equipment can vary from stationary desktop computers (thick clients) running the client application for the virtualized desktop on the workstation operating systems, to specifically designed, stateless clients without local storage capacity and minimal processor power, simply providing physical input (mouse, keyboard) and output (monitor) connectivity (thin clients, zero clients).

1. Remote Desktop Protocol - RDP

Derived from the T.128 ITU-T application sharing protocol as extension RDP is currently implemented in all Microsoft® operating systems. The specifications of the protocol have been recently made public via the Microsoft Developer Network (MSDN)

web resources. The protocol “provides remote display and input capabilities over network connections for Windows-based applications running on a server. RDP is designed to support different types of network topologies and multiple LAN protocols” [36].

In addition to functionality providing seamless connectivity of the remotely hosted application and locally connected devices, features of the protocol include encryption of the data connection and measures to reduce the required network bandwidth to provide a positive end-user-experience. It also has a bandwidth reduction feature comprised of data compression, caching of graphical elements, and network load balancing.

The protocol builds on TCP/IP, utilizing port 3389 by default. The most current version of RDP as implemented by Microsoft is version 7.1, supported by Windows Server 2008R2 SP1 and Windows 7 SP1, further increasing the feature set provided.

Besides Microsoft’s own client implementations, dubbed “Remote Desktop Connection Client” for Windows® and Apple® operating systems, various open-source and commercial versions for other operating are available. As for the supported version of the protocol, only the most current operating systems by Microsoft support version 7.1, all other implementations lag behind and commonly only partially implement the protocol.

There are few publications available to provide information on protocol performance, namely the required bandwidth to ensure an acceptable user experience. Typical of this kind of functionality, the amount of data required to be transmitted over the network depends on various parameters, including screen resolution, color depth, and application usage pattern (determining the degree of changes in the screen depiction from refresh to refresh). Information on scenario-based performance, specifically on bandwidth requirements, can be found in whitepapers published by the vendor comparing the different versions of the protocol (see [37], [38]).

2. High Definition user eXperience – HDX™

Citrix” HDX™ is an umbrella term for the framework of various technologies providing VDI services from the virtualization of single applications to complete desktop operating systems. Each technology in the framework covers different aspects. “HDX

broadcast" is the base service, which is accompanied by (optional) features adding media (video and audio) and advanced graphic capabilities. In addition, secure access and support for user-side peripherals is offered. The various technologies are controlled and balanced through "Adaptive Orchestration," allowing for individual settings to dynamically adapt to the competing requirements of performance, security, flexibility, and network conditions, thereby optimizing usage of available resources [39].

To mitigate the negative impacts of low bandwidth and high latency often found in WAN connections, additional software modules can be deployed to cache, compress, and prioritize network traffic ("WAN optimization") [40].

Citrix" service implementations for VDI (ICA/HDX) utilize TCP port 1494 connections. A detailed description of all utilized IP ports in Citrix installations is available online in the Citrix Knowledgebase [41].

User-side configurations include software clients on common operating systems as well as numerous thin clients and selected zero clients.

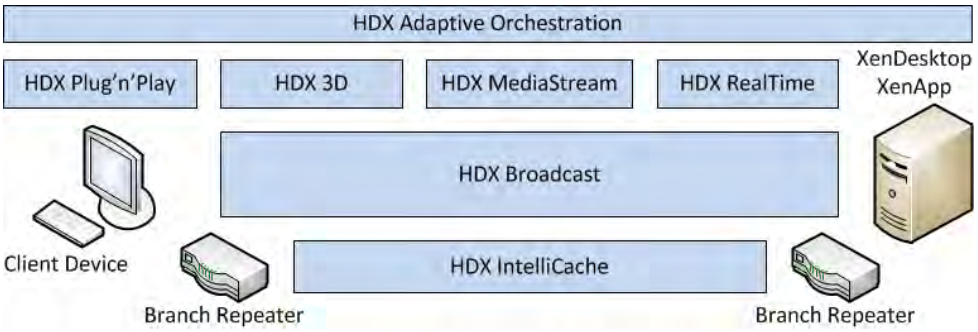


Figure 5. Citrix HDX components/architecture. After [40].

3. User eXtension Protocol – UXP

NComputing's proprietary protocol for desktop virtualization is implemented on the server side via specific software called "vSpace" on top of selected Windows and (limited) Linux operating systems. To minimize server CPU utilization, the virtualization software focuses on virtualizing only those elements of the host OS that are deemed necessary for "a high fidelity experience" on the user side [42]; this approach is described by the vendor as 'selective virtualization" [43].

On the client (user) side, UXP requires proprietary access devices to be built around ARM-based System-on-Chip (SoC), offering different connectivity options to the hosting server (USB direct, PCI card, and Ethernet). The access devices provide physical connectivity for the user's peripherals. UXP data packages are exchanged through a combination of TCP and UDP ports, depending on the reliability requirements for the transported data [44].

The VDI solution offered by NComputing is tailored for less complex small-to-medium installations (up to 30 clients per vSpace installation), and is advertised as a cost-conscious alternative to the offerings of other vendors. To scale towards multiple hosts and leverage from the advantages of server-virtualization, the "vSpace" software supports the VDI infrastructure implementations of VMware and CITRIX [45].

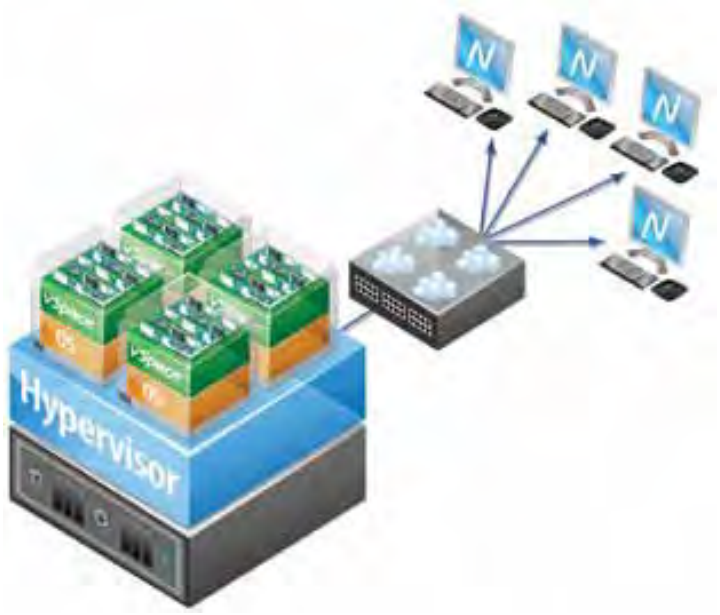


Figure 6. NComputing UXP deployment architecture on virtualized server infrastructure. After [45].

4. Personal Computer over Internet Protocol – PCoIP

The most recent VDI protocol in the market is PCoIP, based on proprietary technology for compressing display data developed by Teradici Corporation. VMware licensed this technology to integrate the protocol in its products as an improvement to the formerly used RDP.

On the host side, PCoIP analyzes the display content, employs a variety of compression encoders (lossless and lossy), and adapts to network conditions to achieve low bandwidth requirements. Both the host and the client side are available in pure software, as well as hardware-augmented implementations. The hosting server can be improved with specific extension cards that off-load CPU and GPU usage to the proprietary silicone. On the client side, VMware implements the protocol in its VMware View™ Client software for Windows operating systems, but thin and zero clients, including in-monitor appliances, are available as well [46].

PCoIP utilizes several TCP and UDP ports in operation for its services [47], since Teradici recently registered specific ports with IANA in the registered port range and current implementations will migrate from the private ports previously used (> #50000) to these (TCP/UDP port 4172) [48].

5. Summary VDI protocols

In general, all VDI protocols/implementations have common attributes with regard to both the information that is exchanged between VDI server and VDI client and its sensitivity to bandwidth and latency constraints.

The traffic load on the network can expected to be asymmetric, with extensively more data sent from the VDI server to the VDI client than vice versa, as the majority of information is video data to present display information on the remote client. Although various lossless and lossy compression techniques for display data are employed to decrease the bandwidth demand, several attributes have strong influence: display resolution, display color depth, display refresh rate, amount of changed display information from frame to frame, and so forth.

In comparison to the dominant display data, other information from the VDI server to the VDI client can be considered of less influence on the required bandwidth for satisfactory performance: audio data (highly compressible), protocol control information (nearly negligible), and, in case it is supported by the protocol, device data (e.g., USB).

The network's latency influences the user experience strongly, as it determines how the user perceives the interaction between his input (mouse, keyboard) and the expected feedback reaction by the desktop environment, mostly the graphical user interface. While textual input via keyboard might allow some acceptable delay between keystroke and display reaction without negative impact on usability, interaction with the mouse is often considered more critical. For example, precise drag and drop actions can be perceived as impossible to conduct if the display information lags too far behind the actual mouse movement by the user.

The different vendors publish only very conservative estimates on the required bandwidth and the maximum latency of a network connection to warrant a "perception free" user experience – one that is perceived equal to the use of local desktop environments [38], [42], [47].

Table 1 presents some characteristics of the aforementioned VDI protocol implementations. As vendors continue to improve their respective products and adapt to market demand, support for additional environments might be available in future releases.

Table 1. Supported environments for VDI implementations

	RDP	HDX	UXP	PCoIP
Supported VDI Server Operating Systems				
Microsoft Windows	✓	✓	✓	✓
Linux	✓	✗	✗	✗
Apple OS X	✓	✗	✗	✗
UNIX like OS	✓	✗	✗	✗
Supported VDI SW Client Operating Systems				
Microsoft Windows	✓	✓	✗	✓
Linux	✓	✓	✗	✗
Apple OS X	✓	✓	✗	✗
UNIX like OS	✓	✗	✗	✗
mobile device OS	✓	✓	✗	✗
Server Hardware Acceleration Available	✗	✗	✗	✓
Thin / Zero Client Available	✓	✓	✓	✓

THIS PAGE INTENTIONALLY LEFT BLANK

III. EVALUATION OF CURRENT M&S TOOLS

This chapter will provide the description of the applied methodology to assess the scope, quality, and integration level of support in modeling and simulation software tools for cloud computing applications regarding network performance parameters. The emphasis will be laid on how the planning, provisioning, maintaining, and management of network infrastructure are supported. A generic, hypothetical scenario for an envisioned Integrated Cloud Application Tool Suite (ICATS) is developed, following the initial steps of the Unified (development) Process [5], [49] to generate an essential set of functional and nonfunctional requirements derived from pragmatic use cases [4]. These requirements will subsequently be used to assess three existing network modeling and simulation tools in the second part of this chapter.

A. INITIAL SET OF GOALS / VISION

Mandated by the "cloud first" policy laid out in [1], governmental agencies and their subordinate organizations are to leverage the advantages of cloud computing for consolidating and acquiring information technology. These efforts include the migration of existing IT infrastructure configurations towards cloud-oriented architectures (e.g., replacing a traditional workplace setup with virtual desktops), to leverage positive effects like increased scalability, gained flexibility, and cost-effectiveness, among others. While much of the focus of current efforts in this infant state of cloud computing is on improving cloud-enabling technologies and business models, all practical implementations of cloud solutions face the limitations and impediments of the available (inter-) network infrastructure. Without sufficient connectivity,[4] all cloud implementations are rendered non-operational.

This simple fact moves the underlying network into the spotlight when implementing cloud-based IT solutions. When managers and engineers collaborate to plan such an architectural "greenfield" – in this case without the constraints and limitations of an

[4] The focus here is on network-bandwidth and –latency, other parameters like reliability, security etc. exceed the scope of this work and are excluded from consideration.

existing production network – they are able to tailor the solution towards the vendor's published best practices and suggestions, which are derived from other, already fielded implementations and prior experiences. However, this is seldom the case; it is rather more common that the envisioned cloud architecture has to be based on existing network infrastructure.

This bears the question: whether the existing network will satisfy the requirements of the virtualized environment, or whether changes and improvements have to be implemented to achieve a viable solution that supports the cloud solution sufficiently for proper user experience – and what these changes are and where will they have to be made.

The goals for the engineering and management of IT projects, not limited to cloud approaches, in general terms are:

(G1) Feasibility analysis (inception),

(G2) Pragmatic, realistic performance assessment of the existing and/or planned network (development),

(G3) Criticality analysis of architecture to enable risk-mitigation (development), and

(G4) Support for identification of migration paths for implementation (deployment), and development of improvements measures (maintenance).

These goals cover all phases of an IT system[5] life cycle from inception, development, and deployment, to maintenance.

To determine essential features that should be provided by ICATS to achieve the goals above, it is useful to develop a problem-appropriate domain model, capturing the essentials of the problem domain, and identifying important objects and their relations.

From the perspective of a user's virtual desktop infrastructure implementations, the user side (client) connects to the service-providing host through the respective protocol, thereby allowing users to transparently interact with the desktop to fulfill their tasks (see Figure 7).

[5] IT system in broad terms encompassing all components, subsystems and applications that make up the solution to satisfy the underlying, original need for such system.

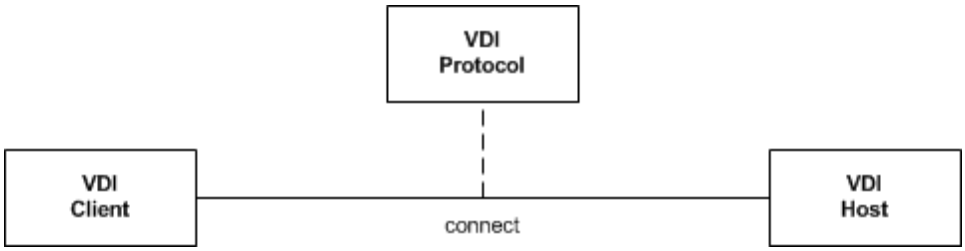


Figure 7. Domain model – VDI aspects

The connection between both nodes, VDI client and VDI host, is realized by the underlying network, which carries the data according to the employed VDI protocol. The network is comprised of several types of interconnected components and can range in complexity from directly connected nodes to arbitrarily sophisticated inter-networks utilizing a wide range of devices and links. Although well documented and known, ISO's OSI [50] and DARPA's TCP/IP [51] reference models include aspects that are overly complex, and unnecessary to address the problems tackled in this thesis; they are therefore not directly utilized to model the problem domain with respect to the attributes of bandwidth and latency.

From an abstracted viewpoint, a network (Local Area Network – LAN) consists of devices and links that connect these devices. Networks themselves can be connected through links with other networks, thereby assembling an inter-network (or Wide Area Network – WAN). The network devices can be either active or passive in the sense that they can be configured or not. Passive devices have predetermined attributes of bandwidth and latency – whereas for configurable devices, these attributes depend on the respective configuration (see Figure 8).

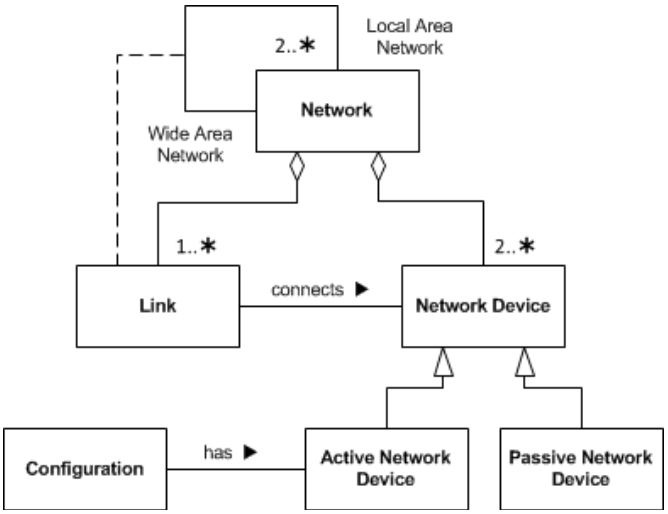


Figure 8. Domain model – network aspects

All nodes connecting to a network do so by an active network device, typically the Network Interface Card (NIC). In general, the VDI client node accesses networks through a physical NIC that is part of the node hardware (workstation, desktop, thin, or zero client). The VDI host side, however, is composed of virtual interface cards that are mapped to one or more physical interfaces by the virtualization environment (see Figure 9).

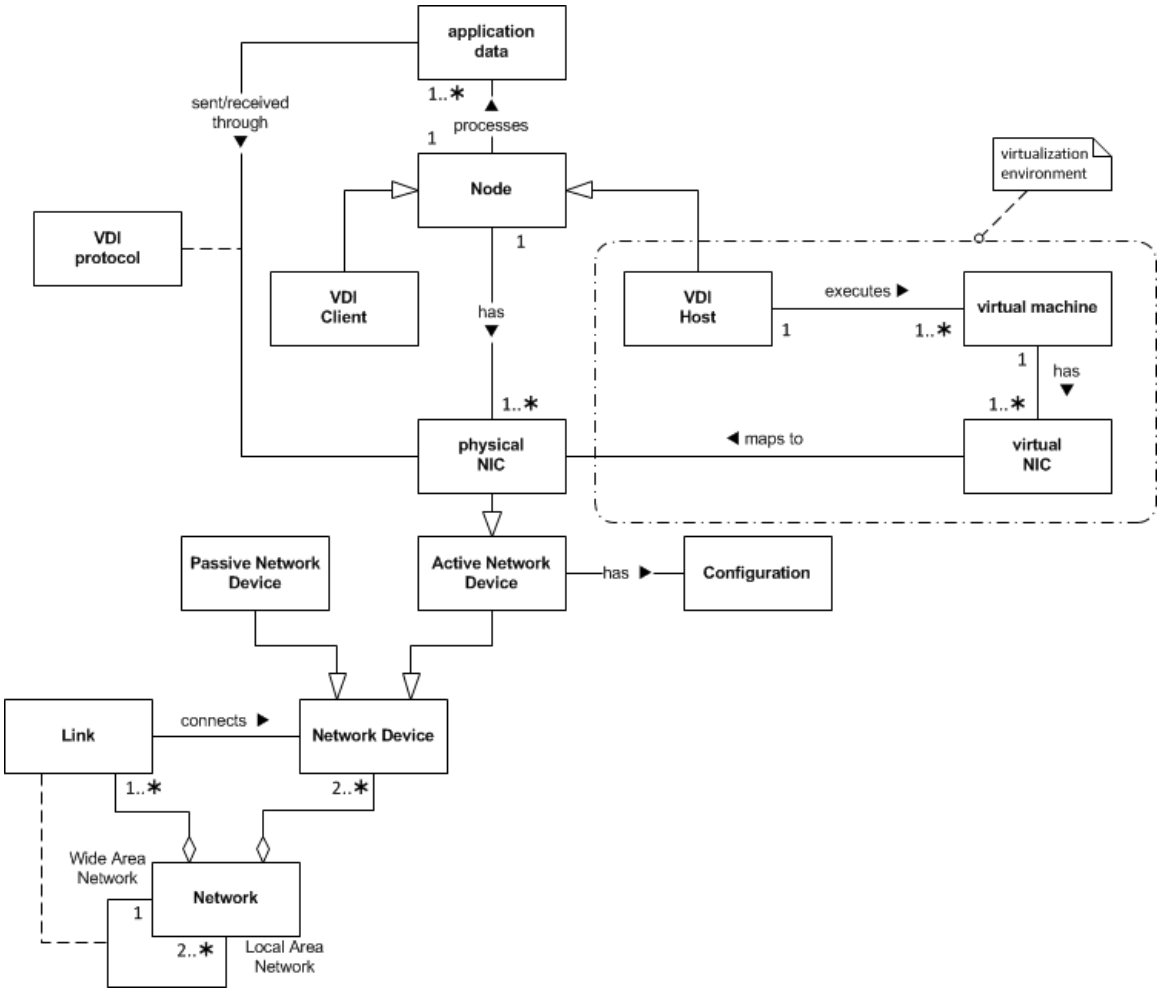


Figure 9. Domain model – complete

The performance of the virtualization environment itself, which incorporates the hardware and software components enabling the virtual desktops, can be assessed and measured with various available vendor-independent[6] and vendor-specific[7] tools that allow stress tests to be run while measuring different performance parameters, including: CPU load, memory usage, storage throughput, and so forth. These tools, however, require the actual setup of the virtualization environment, and primarily are intended and used for configuration and setup validation and testing in the laboratory prior to roll-out as part of a production network.

[6] For example, DeNamek LoadGen or tevron CitraTestVU.

[7] For example, VMware RAWC (Reference Architecture Workload Code) or Citrix EdgeSight.

As the focus of this paper is on the performance (bandwidth, latency) of the network connecting the VDI client and the VDI host, the generic set of requirements developed here concentrates on the modeling and simulation of the network itself, and the virtualization of traffic communicated through it. Along this thought, following the uses cases will be utilized to derive a metric suitable to compare the most common available network M&S applications:

UC1 – Create network model

UC2 – Define node characteristics

UC3 – Execute traffic flow simulation

UC4 – Analyze simulated traffic

UC5 – Modify network model

UC6 – Import traffic load

UC7 – Import network configuration

UC8 – Generate simulation report

B. USE CASES / EVALUATION METRICS

In this segment, the essential elements of the identified use cases are described in brief form, and qualitative metrics for the evaluation are derived.

UC1 – Create network model: The user creates a new network model, then adds active and passive network devices according to the network to be modeled from the library of available device models and configures, where applicable, device attributes accordingly. The user adds nodes from the library of available node models to the network model. The user connects the previously inserted device and node models with links from the library of available link models, in compliance with the available network interfaces and configures the link attributes (where applicable). The system checks the consistency and compatibility of the created model, reports any errors and warnings, and saves the created network for future use.

M1.1 – Richness and fidelity of available device model library

M1.2 – Richness and fidelity of available node model library

M1.3 – Richness and fidelity of available link model library

M1.4 – Ease of network modeling

M1.5 – Support of user-defined node models

UC2 – Define node characteristics: The user opens an existing network model. The user selects a node in the network model and assigns a role/function and traffic load for that node from the library of available roles/functions and traffic loads. The system checks the consistency and compatibility of the node characteristics, reports any error and warnings, and saves the modified network model for future use.

M2.1 – Richness and fidelity of available roles

M2.2 – Richness and fidelity of available traffic loads

M2.3 – Ease of load modeling

M2.4 – Support of user defined traffic loads

UC3 – Execute traffic flow simulation: The user opens an existing network model and defines parameters for the simulation (run time, number of repetitions, flow capture points, etc.). The user activates the simulation, and the system executes the traffic flow simulation in accordance with the network model parameters. The system stores the simulation results for further analysis.

M3.1 – Ease of simulation management

M3.2 – Richness of simulation options

M3.3 – Support for simulation profiles

UC4 – Analyze simulated traffic: The user opens the stored results from a previous simulation run and selects the statistics of interest for detailed analysis. The system presents the selected statistics in a comprehensible format. The system identifies and pre-

sents network elements with traffic congestion (bottlenecks) and provides information on the calculated traffic delay time (latency) between nodes.

M4.1 – Integrated analysis functionality (Y/N)

M4.2 – Ease of traffic analysis

M4.3 – Richness and fidelity of analysis options

M4.4 – Option for exportation of simulation data (Y/N)

UC5 – Modify network model: The user opens an existing network model and adds, deletes, or modifies devices, links, or nodes. The system checks the consistency of the network model and compatibility of the node, device, and link characteristics. The system reports any errors and warnings, and saves the modified network model for future use.

M5.1 – Ease of change to models

M5.2 – Ease of model management functions

M5.3 – Configuration Control and Management

UC6 – Import traffic load: The user opens an existing network model and selects previously recorded life network traffic. The system imports the recorded life network traffic and incorporates the recording as background load to the user defined node or device for future simulation runs. The system saves the modified network model for future use.

M6.1 – Traffic import functionality Y/N

M6.2 – Range of supported recording formats

M6.3 – Ease of incorporation into network model

UC7 – Import network configuration: The user opens an empty or existing network model and selects previously stored network management application information. The system imports the information and generates the respective network model.

The system generates a report summarizing the import. The system checks the consistency of the network model and compatibility of the node, device, and link characteristics. The system reports any errors and warnings, and saves the modified network model for future use.

M7.1 – Import network model functionality (Y/N)

M7.2 – Supported formats / network management information systems

M7.3 – Ease of import into network model

UC8 – Generate simulation report: The system generates a simulation report in document format, summarizing the simulation results and providing analytical information. The user can define format, content, and specific metrics to be included.

M8.1 – Report functionality (Y/N)

M8.2 – Supported formats / options for report

M8.3 – Ease of report generation

The Table 2 provides some additional description, as well as the assigned range of possible values, for each metric applied in the assessment. The values of Y/N indicate the presence or absence of that characteristic; values of 1-5 indicate a Likert-type scale in which the numerical values are assigned to quality levels of that characteristic: 1 – Poor, 2 – Fair, 3 – Good, 4 – Very Good, and 5 – Excellent.

Table 2. Metrics table, aspects under evaluation, value range

ID	**Short Title / Aspects for Assessment**	**Value**
M1.1	Richness and fidelity of available device model library	1-5
	The device model library is assessed with regard to number of supported devices, model fidelity (attribute level), and variety in device vendors supported.	
M1.2	Richness and fidelity of available node model library	1-5
	The node model library is assessed with regard to number of supported nodes	

ID	Short Title / Aspects for Assessment	Value
	(types, roles), model fidelity (attribute level), and flexibility of model (configuration options).	
M1.3	Richness and fidelity of available link model library	1-5
	The link model library is assessed with regard to the number of supported link technologies, model fidelity (attribute level), and flexibility of model (configuration options)	
M1.4	Ease of network modeling	1-5
	The application (suite) is assessed with regard overall usability, (graphical) user interface, automation support, consistency, and level of integration.	
M1.5	Support of user-defined node models	Y/N
	Does the application support the definition of node models by the user or is it limited to the provided library?	
M2.1	Richness and fidelity of available roles	1-5
	The library for node role models is assessed with regard to the number of models, options (functions), and the flexibility of the role models (configuration options).	
M2.2	Richness and fidelity of available traffic loads	1-5
	The library for traffic load models is assessed with regard to the number of models, and the flexibility of the load models (configuration options).	
M2.3	Ease of load modeling	1-5
	Is the modeling of traffic loads well supported by the application or does it require detailed application training and high level of education prerequisites in network engineering? Does the application support scale well from simple to complex load modeling?	
M2.4	Support of user defined traffic loads	Y/N
	Does the application provide for the use of user defined traffic loads or are the options limited to the provided library?	
M3.1	Ease of simulation management	1-5
	Does the application (suite) automate the management of simulations or does it provide for flexible workflow support?	
M3.2	Richness of simulation options	1-5

ID	Short Title / Aspects for Assessment	Value
	Does the application support detailed and flexible options for simulation runs?	
M3.3	Support for simulation profiles	Y/N
	Does the application provide support for the definition of simulation profiles that can be executed on different models (batch processing), thereby ensuring comparability?	
M4.1	Integrated analysis functionality	Y/N
	Does the application suite provide analysis functionality or are the simulation results analyzed by additional, non-integrated applications (e.g., statistical package)?	
M4.2	Ease of traffic analysis	1-5
	How hard is it to derive detailed analysis of the simulation results?	
M4.3	Richness and fidelity of analysis options	1-5
	Does the application analysis support scale well from simple to complex?	
M4.4	Option for exportation of simulation data	1-5
	Can the simulation results be exported to various formats (for analysis by external applications and further processing)?	
M5.1	Ease of change to models	1-5
	How hard is it to change the models? Can changes be scripted?	
M5.2	Ease of model management functions	1-5
	Is the application support for managing models integrated, does it support automated workflows?	
M5.3	Configuration Control and Management	1-5
	The application is assessed with regard to provided model configuration and control management functions (e.g., change control, multi-user collaboration, versioning).	
M6.1	Traffic import functionality	Y/N
	Does the application (suite) provide functionality to import previously recorded traffic load?	
M6.2	Range of supported recording formats	1-5

ID	Short Title / Aspects for Assessment	Value
	How many recording formats are supported? Are the most common formats supported?	
M6.3	Ease of incorporation into network model	1-5
	Does the incorporation require additional conversion steps? How easy can the recorded traffic be adopted for use in the network model (address range, time etc.)?	
M7.1	Import network model functionality	Y/N
	Does the application (suite) support the generation of network models from imported network management systems information?	
M7.2	Supported formats / network management information systems	1-5
	How many network management information systems are supported? Are the most common supported?	
M7.3	Ease of import into network model	1-5
	Does the import of network management systems information require additional preparation/conversion? How is additional information unnecessary for the model handled? How is missing or insufficient information treated?	
M8.1	Report functionality	Y/N
	Does the application (suite) provide for (automated) report generation?	
M8.2	Supported formats / options for report	1-5
	Can the content of the report be tailored flexible? What and how many output formats are supported?	
M8.3	Ease of report generation	1-5
	Does the report generation require much effort in time and resources? Can report content be scripted?	

C. EVALUATION OF EXEMPLARY TOOLS

This section begins with the description of the hypothetical (but nonetheless realistic) project by which the exemplary tools are evaluated, followed by a brief description and assessment of each tool. The section concludes with a summary of the findings.

1. Project

The survey of the selected network modeling and simulation tools is based on a hypothetical but realistic project scenario. The scenario layout centers on options for future IT deployments at the Naval Postgraduate School (NPS) utilizing virtual desktop infrastructure technology. Although NPS is a specific example, the scenario is kept generic enough to be generalized for similar distributed institutions outside academia.

NPS as an institution maintains a complex computer network that supports teaching, research, and administrative computing ranging from the office desktop to the school's high performance computing environment. Classrooms, the library, and the campus in general provide secured wireless connectivity to the NPS network. The options for wired connectivity in class-rooms are limited, depending on the building and its infrastructure. The wireless network is primarily used to access the NPS-provided email services, and the school's intranet and online resources, including the collaboration environment and the Internet.

The following map shows the NPS campus and the various buildings with on-site connectivity to the NPS enterprise backbone (ERN). The ERN network infrastructure reaches all buildings with varying bandwidth (with a focus on academic and administrative buildings). The main data center is located in one of the academic buildings; from there NPS connects to off-site resources and the Internet.

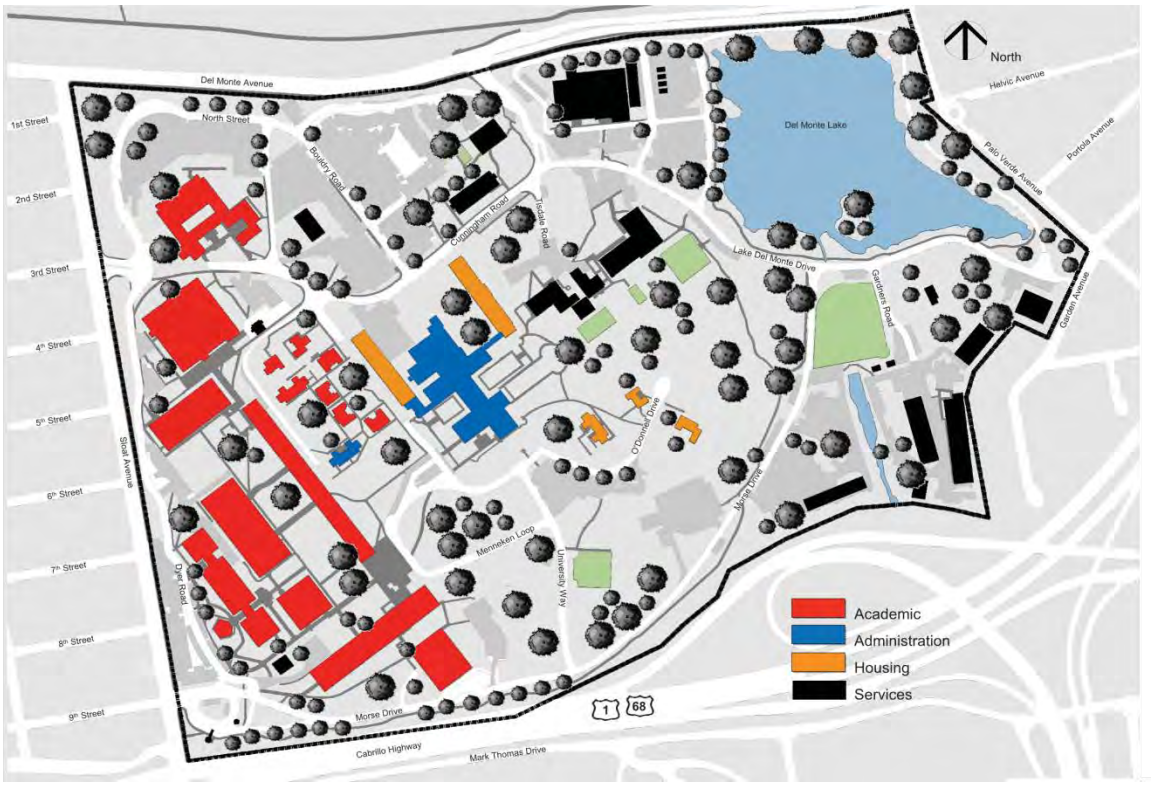


Figure 10. Naval Postgraduate School Campus – buildings colored according to their main functional role: academia, administration, general services, housing (From: NPS Intranet)

NPS offers graduate programs both on and off campus through its four schools, extending its educational reach beyond the residential students to eligible personnel in its distant learning activities. For many of the classes offered by the Graduate School of Operational and Information Sciences (GSOIS), the students are required to use specific software applications, including tools for: web-development, data-base management, application development, business process modeling, and the like. The school's Information Science and Computer Science departments maintain several laboratories providing desktops with application setups for student use; these require extensive maintenance for recurring updates, configurations, and service. For some classes, the required applications can be offered to the students for installation on their privately owned computers (in most cases laptops). Since the students own various types of computers with operating systems that do not always matching the system requirements of the provided applications, lecturers and supporting faculty spend valuable time supporting the students to get their con-

figurations to work as needed. Distant learning students, sometimes located cross-country or even abroad, naturally do not have access to the local laboratory installations. They sometimes experience, in addition to the troubles caused by differences in time-zones, even more struggles while creating the course-required computer environment from their distant sites.

Figure 11 shows examples of typical sites from which NPS distant learning students participate in educational programs and classes. The figure also shows the respective latency values caused by their physical distance as measured by service providers within their core network [52].

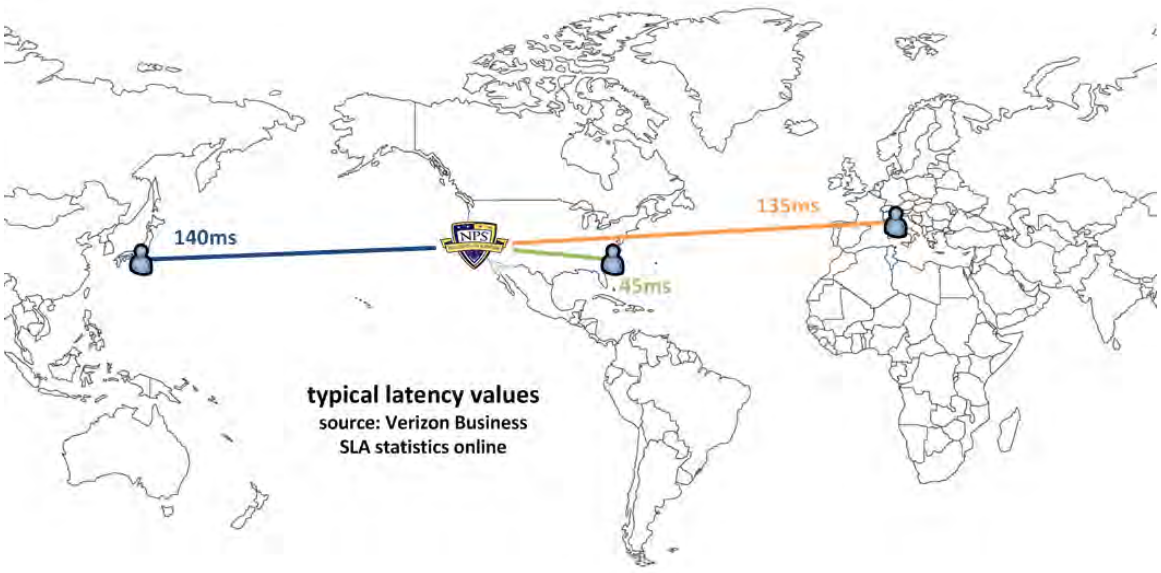


Figure 11. Typical sites from which NPS distant learning students participate

2. Scenarios

With the increasing maturity of cloud computing solutions to consolidate data centers and provide virtualized desktops (that can be accessed via power-efficient thin- and zero-clients or client applications available for most current operating systems), GSOIS is considering a migration of existing laboratories to VDI. The advantages of such a solution would include: lower effort for maintenance, the ability to provision the

amount of required desktops more quickly and flexibly, the ability to provide pre-configured and freshly set up user environments for each quarter, centralized back-up, easier license management, more coherent Information Assurance measures for improved regulatory compliance, and so forth. In short: many desirable properties.

However, GSOIS – as part of the NPS enterprise – uses the network infrastructure provided and maintained by ITACS (the centralized IT service provider at NPS). Modifications, improvements, and configuration of the network are aligned across the requirements of all enterprise entities to ensure de-confliction and efficiency demanded by limited financial resources.

Before committing to the VDI migration – and defining and procuring the respective server infrastructure and VDI clients – GSOIS management needs to gather information to mitigate their investment risks by assessing the impediments that might be posed by the existing network infrastructure. Modeling and simulation is identified as the means to provide such information, as well as to provide network infrastructure criticality reports and identify possible mitigation measures (e.g., network modifications and improvements).

Continuing along with our hypothetical project, the modeling and simulation applications under assessment were evaluated (with the previously described metrics) along the following network scenarios:

Scenario 1 – Classroom: One classroom, 15 students connect wirelessly, 5 students connect via wired connection from various private laptops to their assigned virtualized desktops (combined wireless/wired load from/to one classroom on site).

Scenario 2 – Computer Lab: One computer lab (room), 30 students connect via zero-clients to their assigned virtualized desktops (wired load from/to one computer lab on site).

Scenario 3 – Off Campus: 60 students in the local area off-campus connect via VPN to the NPS network and access their assigned virtualized desktops (aggregated load from/to NPS network).

Scenario 4a – Distant Learning CONUS: One student connects from the Fleet Concentration Area (Norfolk, Virginia) via VPN to the NPS network and accesses his/her assigned virtualized desktop (load with medium expected latency).

Scenario 4b – Distant Learning OCONUS: One student connects from either USPACOM or USEUCOM (load with high expected latency).

Figure 12 depicts the scenarios of the hypothetical VDI project, focusing on the main elements to consider when modeling the computer network for traffic simulation. Critical elements are the increased latency and limited user-available bandwidth introduced by connections from off-campus, as well as bandwidth constraints resulting from a large number of users accessing the VDI infrastructure from single network segments (wired) or through single access points (wireless).

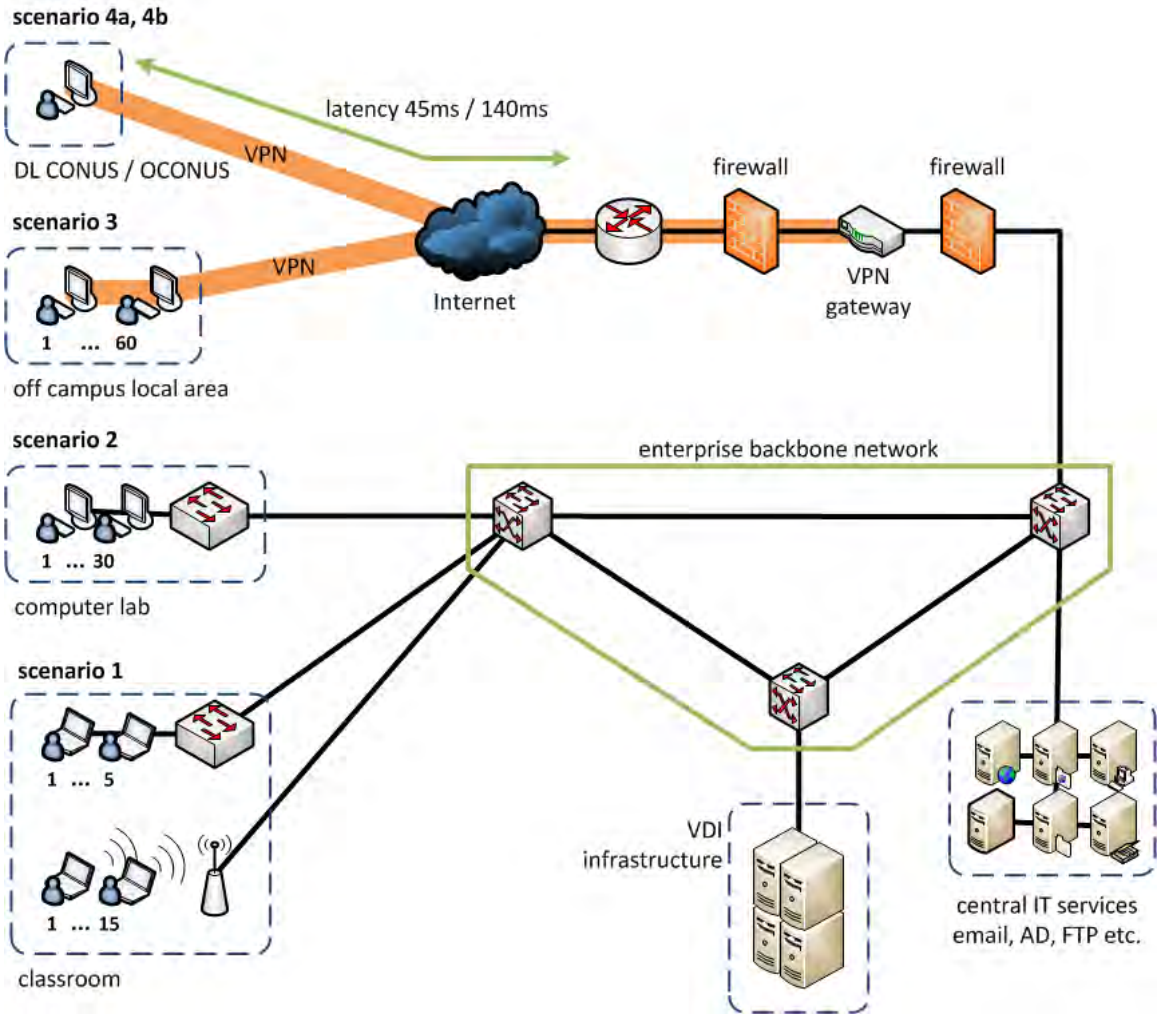


Figure 12. VDI deployment scenarios for hypothetical project

The following sections briefly present the three network simulators that were assessed for this thesis (ns-3, NetSim and OpNet Guru). Each section includes a short explanation of the assessment results, and the results are summarized in the subsequent table.

3. ns-3

a. Description

ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use. [53].

The core contributors to the ns-3 project maintain online resources that support collaboration and provide documentation, access to source code and binaries, and general support for the interested community (at www.nsnam.org). The project was founded in 2006 to develop a successor to ns-2 in an effort to develop an open simulation environment for network research based on a well-documented, easy-to-use and debug simulation core. As ns-3 is a library of network simulation models consisting of objects written in C++, users interact with it by writing C++ or Python applications that define the desired network model (by instantiating the respective models and calls to the simulation core functions to execute the simulation). As ns-3 requires the GNU toolchain [54] for execution, the natural environment for it is a Linux or Linux-like system.

The ns-3 architecture [55] is based on some key abstractions implemented as C++ classes (with Python wrappers for the API access through Python scripts) that can be mapped to the domain model laid out at the beginning of this chapter.

The ns-3 `Node` class represents a basic computing device and provides methods to manage the representation of the device in the simulation, and therefore closely corresponds to the Node concept in the domain model. Functionality is added to `Node` instances in the form of applications and peripheral cards. The `Application` class in ns-3 provides the methods to represent user-level applications in the simulation by generating activity (besides other network traffic, like background data exchanges) to model the simulation relevant behavior of nodes in a network. `Node` instances (modeling computers) with `Application` instances generating VDI traffic correspond to the concept of “VDI client” / “VDI server” in the domain model.

The concept of “Link” in the domain model represents the connection of “Network Devices” to compose a “Network”; in the ns-3 architecture, the `Channel` class represents this concept by providing methods for managing communication subnetwork objects and connecting `Nodes`. But the ns-3 abstraction for `Channel` also includes the domain concept of “Passive Network Device,” since `Channel` instantiations may also represent network devices (e.g., Ethernet switches), which have pre-determined functions/behaviors and cannot be configured (in contrast to “Active Network Devices”).

In ns-3, the `NetDevice` class is utilized to model the combination of network hardware and the corresponding driver that controls the hardware. Instances of this class represent the domain model's "Active Network Device" concept, to model items like network interface cards (NIC), routers, or wireless access points (including their configurations).

To support the user of ns-3 in instantiating common network elements when writing the simulation application, the developers provide topology helpers. These objects combine the distinct steps of creating a simulation model element from ns-3 core operations into an easier-to-use model of usual elements and attributes. For example, `BridgeHelper` has methods to create a specialized `NetDevice` instance of the subclass `BridgeNetDevice`, and includes configuration via its `SetDeviceAttribute` method, adding the device to a node and attaching a list of `NetDevices` as ports to that bridge [56].

The application user can (in addition to defining the network to simulate and the simulation management parameters) activate the logging component of ns-3 in each of the elements in the network model for which simulation result output or debugging information is desired. Logging data can be directed to all pipes of the system environment, including the file system, for analysis.

Users of ns-3 need programming skills in C++ and Python to leverage the full capabilities of the network simulator. The simulated network is developed as an application in the provided object-oriented framework. In case the available components of the framework do not provide the desired model or functionality, users will have to develop their own components as needed (and are asked to contribute such components to the repository of the project to increase the versatility of ns-3).

For the purpose of this thesis, the current stable release of ns-3 (ns-3.11) was installed on an Ubuntu 10.04LTE (Lucid Lynx) and used to create network simulations for the outlined VDI usage scenarios with the included model objects to assess ns-3 along the metrics.

b. Assessment

M1 – The “libraries” with which ns-3 is provided contains a variety of sub-classes for `NetDevice` and `Channel` objects, covering the most common elements with emphasis on IP networks in an abstracted, generic version. Depending on the specific specialization class, the fidelity of models ranges from generic and rather coarse to highly sophisticated and very fine. The main focus of the existing libraries is rather research oriented; Node classes and the required Application classes are sparse and very generic. Any traffic model beyond very simple patterns has to be provided and implemented by the user. Since the simulation network is modeled by writing an application using the framework objects at the code level (in absence of assisting elements), the modeling itself has to be considered as complex.

M2 – ns-3 not only allows the definition of traffic loads (in the sense of traffic being the result of activities of the application object), it requires such effort, since the provided models do not contain any load models beyond the most generic. The research nature of the ns-3 project and its open architecture, however, provide for very high flexibility.

M3 – Management of the simulations is the responsibility of the user and determined by the user application. ns-3 does not provide for such functions, nor does it intend to; the same holds true for the support of simulation profiles – the user decides by the architecture and design of the simulation application which parts of code are suited for reuse.

M4 – No analysis of the simulation results is integrated in ns-3 itself; it is designed as a framework to implement network simulations and consequently relies on readily available tools outside the project’s scope for result analysis. The detailed logging and tracing mechanisms allow for the generation of analyzable output; the commonly used pcap format is supported.

M5 – Configuration control and management of simulation models is left to the user; however, as the skills required to develop and write the simulation application include detailed programming knowledge, the appropriate versioning mechanisms for

source code can be utilized to realize adequate configuration control and management in collaborative environments.

M6 – ns-3 does not, per se, support the import of previously recorded live network traffic into a simulation model.

M7 – ns-3 does not support the import of network management information to create network models.

M8 – No report functionality is part of the ns-3 project; it is up to the user to produce reports from the analysis conducted with external applications.

The open source community around the ns-3 project maintains a finely architected framework for the modeling and simulation of computer networks; it is most flexible, well documented throughout, and well suited for scientific research (where it indeed finds its most common use). The focus on object-oriented software design requires the users to be well versed in programming – as well as network engineering – to make full use of the framework. As each network model in ns-3 is itself an application to be executed for simulation, the scalability and reusability for larger networks depends on the user's ability to manage larger source code. The available object models for devices, nodes, and traffic loads are mostly rather generic items contributed from research in the field that do not reflect current commercially available components. ns-3 relies highly on external, existing tools for coding, analysis, and management – it is (as intended by its contributors) a core for the M&S of computer networks.

4. NetSim (Academic / Standard)

a. Description

> NetSim is a comprehensive tool for studying computer networks." [57] It is produced by Tetcos (Bangalore, India) and advertised as "[..] a versatile tool to simulate and analyze computer networks [..]" and offers "[..] comprehensive modeling facility, detailed performance reports and enhanced protocol analytics. [58]

Intended primarily for educational purposes, NetSim includes several functional modules that support the simulation and analysis of computer networks, integrated into one application with a graphical user interface.

The “NetPatrol” module enables the capture of live network traffic for detailed analysis of network protocols and their implementations down to the byte level, including graphical statistics. To further the understanding of protocol implementations, NetSim’s “Development Environment (DEN)” provides programming exercises that include reference implementations of various network algorithms. It also contains the option to connect user-developed model code into the simulation module for verification via simulation execution and traffic analysis.

The “NetSim Project” module allows for the development and research of network protocols based on a library of provided primitives, which can be modified or recombined with existing and new network protocols, executable in the simulation module.

The main module, ‘simulation,” provides a graphical interface that allows users to design a network by dragging and dropping network components onto a canvas. The network components are connected via network links, created by clicking the respective components in sequence of connection. The underlying concept is closely related to real hardware. Since the permissible link technologies are determined by the network component, only technically possible links can be created. Once the network is created, the configuration of network components and links is performed using context-sensitive menus reflecting all of the configuration options and attributes for the respective component and link. The graphical depiction of the network model resembles the physical composition; logical or abstracted architectural diagrams are not available.

Traffic load in NetSim is assigned to source nodes (transmitters), which define the type of traffic, probability distribution, packet size, drain nodes (receivers), and other parameters. The available options for traffic are defined in the node model and depend on the type of equipment that is modeled (router, switch, mobile telephone, etc.).

Before executing the network model simulation, NetSim validates the model for logical errors (e.g., missing connections or un-defined parameters). During

simulation, NetSim captures simulated traffic and performance measurement, as well as conditions of interest (e.g., routing table entries) for later analysis. After simulation completion, the analysis module of NetSim provides detailed information and graphical statistics in pre-defined categories.

NetSim requires a Microsoft Windows XP (and later) operating system, and relies heavily on Adobe Flash for its user interface. The programming module of NetSim utilizes the "minimalist GNU for Windows" (MinGW) [59] development environment for its underlying functions (to create library functions executable in the simulation module) and WinPcap [60] to capture live network traffic in the real-time module.

For the purpose of this thesis, NetSim 5.0 and the required additional commercial and open-source applications were installed on a Windows 7 Professional environment.

b. Assessment

M1 – The extensive device, node, and link libraries (only "NetSim Standard") of NetSim allow the creation of detailed and complex network models that include all common technologies. However, the network components are generic, vendor-independent, and require adaption if the component does not precisely match the equipment being simulated. The modeling of networks is supported by an easy-to-use graphical interface replicating the physical setup of components in a single layer.

M2 – NetSim provides detailed, high-fidelity models of network components. However, the underlying traffic load models and node role models are limited to generic packet generators, and primarily defined through packet size and inter-arrival time, both configurable as various probability distributions.

M3 – Simulations in NetSim are executed in single runs on the currently opened model; the main configuration attribute for the simulation is the simulation run-time. All other attributes are defined in the network model itself prior run-time.

M4 – The analysis functions provided by NetSim are extensive and tailored to the most common statistics in the context of network performance. User-

configurable statistical analysis is not available within NetSim, but simulation results can be exported to Comma Separated Value files for further analysis with external applications.

M5 – Due to the graphical interface, changes in network models can be applied relatively simply, but might require an extensive number of steps (for example, changing the IP scheme in a larger network requires the access to the context-sensitive configuration menu for each single network component). No specific support is provided for the configuration control and management, or the versioning, of network models.

M6 – Although live traffic can be recorded and analyzed with the "NetPatrol" component, the ability to import such traffic into a network model as traffic load is not implemented.

M7 – NetSim does not support the generation of network models from imported network management systems information.

M8 – The most common performance parameters and statistics are a part of NetSim's included analysis functionality. Analysis reports are automatically generated after each simulation run. Export of the analysis reports into external formats is limited; the content of the reports can only be tailored in limited ways.

NetSim provides a superb, flexible suit of tools for the training and education of the computer and network engineer community; the concept is targeted to make the inner workings of computer networks understandable and visible. Devices, protocols, and network applications up-and-down the OSI model are the center of focus for NetSim. It offers all generic, standard-conforming objects in the world of computer networks. The modeling and simulation of real, existing networks requires the expression of commercial components as generic models. The scalability of the model and simulation engine could not be assessed with the version available; however, the user interface hints at limitations in this regard, since it has only a single, plain, and strictly graphic-oriented organization that is strictly limited to single projects.

5. OpNet ITGuru

a. Description

OpNet ITGuru is advertised as part of the vendor's network performance management portfolio, offering support for network planning and optimization throughout the complete project life-cycle from inception, development, and implementation to maintenance at the enterprise level [61].

Specialized off-springs of ITGuru – targeted at enterprise-level network engineering – are available for training and education (ITGuru Academic), and for service providers concerned with their large communication networks (SPGuru).

Workflows in ITGuru center around the "Project Editor" (the main user interface), from which network models are created and managed, simulation details defined, simulations executed, and results analyzed. The ITGuru Project Editor provides a graphical network model editor and an object-oriented tree view; in combination, they provide a comprehensive view of the model. The graphical workspace resembles the physical distribution of model objects, and thereby includes the constraints imposed by distance directly into the model during simulation. The user defines the dimensions of the network to model at the beginning of a project, and the workspace scales respectively; the predefined scales range from a single office to the complete world (including underlying topographical maps). User-defined dimensions, as well as logical networks (with no assigned physical extend), can be selected.

Objects for the network model are provided in object libraries; at the initialization of projects, the components of the project-specific library are selected as a subset of all licensed components and models. Libraries contain all kinds of objects for models, organized in "families," ranging from vendor-specific network devices, nodes, and links to network protocols and configurable or pre-defined traffic loads. Since the objects in the library can be sorted and searched on by all of their properties, the organization and management of a large number of objects is possible.

In addition to the physical model view, objects in the network model can be addressed, edited, and modified in logical view (by selecting all components belong-

ing to one network segment), regardless of their physical location. For larger networks, the graphical network model provides the capability to "fold" objects into groups, enabling abstracted overviews where possible, and detailed views where necessary.

ITGuru is delivered with extensive documentation and provides interactive tutorials for the most important functionalities. The application architecture allows the vendor to modularize the product into packages tailored for specific application fields – scaling the product through optional modules, specialized models and complementing solutions – to integrate into extended network management.

For the assessment, OPNET ITGuru Version 17.0 has been installed on a Windows 7 Professional environment, including the current Java runtime engine (required for access to the interactive, browser-based ITGuru documentation).

b. Assessment

M1 – The provided library in ITGuru is vast and detailed, and includes vendor-specific as well as generic equipment. Users can customize models and easily derive variations from existing objects to meet their requirements. All relevant technologies and network devices are included. The development of network models for simulation is supported through a combination of graphical and textual presentation and editing options.

M2 – The object library contains various options to define traffic loads; these include role-based loads ("Application Traffic Models"), traffic loads ("Baseline Loads," "Traffic Flows"), as well as generic packet loads between nodes. The pre-defined loads cover general applications (email, FTP, etc.), and simple user-defined loads can be created relatively easy either by either deriving them from existing models, or by defining probability statistics.

M3 – ITGuru provides two levels of simulation management: one providing only the most common parameters with default values for instant results, and the second offering access to all simulation options that the discrete event simulation engine has to offer. Simulation settings and options, including the selection of statistics to rec-

ord, can be applied to variations of the network model within a project, thereby enabling the application of profiles on different model scenarios for comparison.

M4 – The included analysis functionality is extensive, and ranges from simple performance overviews to detailed diagrams of specific aspects. Within a project, different scenarios and their respective simulation results can be easily compared. The analysis module provides reports focusing on differences between models or simulation runs.

M5 – Network models are managed within projects that can contain all variations under analysis. Models or sections thereof can be cloned and reused in variations; logical or geographical parts of models can be grouped to combine objects – which can be reused in other parts or variations of the network. Changes to models can be applied in either the graphical or in the object-oriented treeview; groups of objects can be selected to apply modifications in a single step. Although ITGuru provides an auto-save function, no version control system is integrated; ITGuru rather relies on external version control applications for this aspect, and can (for this purpose) export and import its models in XML format.

M6 – Recorded live network traffic can be imported and used as either background load or, through additional modules, converted into application traffic that can be assigned to nodes as traffic flow model for further simulation runs. As the available version of ITGuru does not include the additional modules for extended traffic import, this functionality was not fully assessed.

M7 – As part of an integrated suite, ITGuru provides for the option to import network management information to create network models of existing production networks. The import options not only include the network topology itself, but also the configuration information of network devices like selected routers, switches, and firewalls to fully replicate the network under analysis. The import functionality is primarily tailored to the OpNet VNEServer module for network data management. Imports from other network management information and management systems require conversion steps within VNEServer.

M8 – ITGuru offers extensive customizable reports generated from the statistics recorded during simulation runs. Reports are exported in HTML format as interactive hypertext information with included graphs, interpretable by standard browsers.

ITGuru is part of a complex suite of tools, proven in the context of commercial communication network engineering over the last 25 years. The degree of fidelity of its models, the support of workflows, and the options for comparative analysis reflect the requirements of OpNet's customers. The academic off-spring is not only suitable for the academic environment as a tool for teaching and exploring network engineering, it is also appropriate as an introduction to the concepts and user-interfaces of the broader tool suite. The replication of existing or planned networks in models for executable simulation is enabled by the large library of components – which includes most of the common equipment found in professional network environments.

However, ITGuru does not currently offer the capability to model and simulate cloud computing environments to a fidelity that would allow management and engineers to fully comprehend the risks and mitigation options they need to consider when planning, engineering, and maintaining networks for cloud-based, virtualized environments. Although ITGuru allows for the modification of existing traffic load models in a limited manner to adapt to user requirements, it currently lacks the node and traffic models specific to cloud implementations (VDI servers, clients, and traffic loads) in its available library that would enable full simulation support for cloud computing environments.

Additional components to ITGuru are available that link the simulation model to real networks and IT network components. This option allows, for example, the testing of an existing VDI infrastructure under controlled conditions against the simulation of the production network to optimize configuration prior to deployment. However, this combination of simulation and real implementation would require an investment in cloud infrastructure components. It consequently applies only to the later stages of cloud projects after commitment to a solution and acquisition, and thereby, does not fully leverage the potential of modeling and simulation in earlier phases.

6. Summary

The following table lists the assessment results as qualitative measures for the evaluated network modeling and simulation applications:

Table 3. Qualitative assessment results

	nc-3	**NetSim**	**ITGuru**
UC1 – **Create network model**			
M1.1 – Richness and fidelity of available device model library	2	3	5
M1.2 – Richness and fidelity of available node model library	1	2	4
M1.3 – Richness and fidelity of available link model library	2	3	5
M1.4 – Ease of network modeling	1	3	3
M1.5 – Support of user-defined node models	Yes	Yes	Yes
UC2 – **Define node characteristics**			
M2.1 – Richness and fidelity of available roles	1	2	4
M2.2 – Richness and fidelity of available traffic loads	1	2	4
M2.3 – Ease of load modeling	2	2	4
M2.4 – Support of user defined traffic loads	Yes	Yes	Yes
UC3 – **Execute traffic flow simulation**			
M3.1 – Ease of simulation management	2	3	4
M3.2 – Richness of simulation options	2	3	5
M3.3 – Support for simulation profiles	No	No	Yes
UC4 – **Analyze simulated traffic**			
M4.1 – Integrated analysis functionality	No	Yes	Yes
M4.2 – Ease of traffic analysis	n/a	3	5
M4.3 – Richness and fidelity of analysis options	n/a	3	4
M4.4 – Option for exportation of simulation data	n/a	2	2
UC5 – **Modify network model**			
M5.1 – Ease of change to models	2	3	5
M5.2 – Ease of model management functions	2	2	4
M5.3 – Configuration Control and Management	2	2	3

	ns-3	**NetSim**	**ITGuru**
UC6 – **Import traffic load**			
M6.1 – Functionality of traffic import	No	No	Yes
M6.2 – Range of supported recording formats	n/a	n/a	3
M6.3 – Ease of incorporation into network model	n/a	n/a	3
UC7 – **Import network configuration**			
M7.1 – Import functionality	No	No	Yes
M7.2 – Supported formats / network management info systems	n/a	n/a	2
M7.3 – Ease of import into network model	n/a	n/a	3
UC8 – **Generate simulation report**			
M8.1 – Report functionality	No	No	Yes
M8.2 – Supported formats / options for report	n/a	n/a	3
M8.3 – Ease of report generation	n/a	n/a	4

All three of the exemplary tools have been assessed along the outlined metrics with the intent to evaluate the suitability for modeling and simulation of practical cloud computing solutions to support the management and engineering of "cloud projects." Each tool showed its own strength, mostly determined by the origin: ns-3 with the solid academic background in research and science as flexible and open framework focusing on smaller aspects of networks; NetSim with strong ties to the practical arena of engineering schools and their requirements for making the inner workings of computer networks visible and comprehensible – and offering the simulation of medium sized networks; and ITGuru as the answer to practical engineering and management requirements in the commercial world of enterprise communication networks.

The surveyed modeling and simulation tools are under permanent development for further improvement. However, for practical use in upcoming IT projects that wish to further leverage the promises of cloud computing, none of them is currently fully suitable. Closest to the task is ITGuru and its complement of additional applications in the respective network management suite, but even it falls somewhat short. Offers to provide

assessments and risk mitigation for cloud projects – available from the vendor as consulting services – emphasize the identified, but not fully satisfied, need for such solutions.

The problem-appropriate, realistic modeling and simulation of emerging, complex technologies in practical terms is the complementary effort to gathering best practices and experience when realizing IT projects. In the case of cloud technologies and their inherent dependency on network connectivity, it becomes clear that this capability needs to be addressed.

IV. EXPERIMENT

A. GOALS OF THE EXPERIMENT

The previous chapter identified the existing capability gap for modeling and simulation tool support for planning, engineering, and maintaining computer networks for emerging cloud solutions. The increase in available bandwidth for computer networks falls far behind the foreseeable increase in bandwidth demand, even considering mitigation technology improvements (e.g., data compression or optimized protocols). Latency in computer networks, introduced by delays in processing and, more dominantly, as the product of physically-limited velocity of signals across large distances, is increasing with the separation of user and computing power that comes with the implementation of the cloud paradigm. Consequently, with the increasing number of implementations of information systems using cloud computing paradigms, the importance of underlying computer networks (and the impediments they might create) increases the need for planning, implementing, and managing such solutions.

This chapter approaches a possible solution to the identified gap in available modeling and simulation tools by developing a structured methodology to capture the influence of bandwidth and latency on VDI solutions on a proof-of-concept level. The methodology can be used to derive information that can be utilized to further develop load models for M&S tools that replicate the relevant behavior of complex VDI protocols sufficiently to assess the impact of the modeled, existing, or planned computer network on the projected VDI implementation in the early stages of such projects.

B. METHODOLOGY

For this experimental part of the thesis, a small but complete virtual desktop environment was created allowing the use of two major VDI protocols, RDP and PCoIP. The experiment's environment included a configurable network connection between VDI server and VDI client, allowing the replication of the user scenarios outlined in Chapter III (see Figure 12) in a controlled, repeatable manner with regard to both bandwidth and

latency. Methods to measure performance parameters, and to record and analyze the network traffic of interest, have been incorporated.

1. User Environment

The general properties of virtual desktop infrastructure protocol traffic in networks have been laid out in Chapter II: asymmetric traffic between VDI server and VDI client, combination of lossless and lossy compressed data, and so forth. The major factors that influence this type of traffic with regard to required bandwidth for a seamless user experience (regardless of the specific implementation) are:

- Virtual desktop screen resolution and color depth
- Type of displayed data (text – low change rate, video – high change rate)
- Available computational power for analysis and compression of data (server side)
- Available computational power for decompression of data (client side)

For the experiment, four different VDI client configurations were selected to represent typical user environments with which virtual desktops are used, and which differ widely in their computing power and desktop parameters. The essential properties of the clients are listed in Table 4. Client configuration 1 and 4 are considered typical for temporary use in the hypothetical project (e.g., students), whereas configurations 2 and 3 resemble what can be expected when VDI implementations are deployed in office or enterprise environments.

Table 4. VDI client configurations for the experiment

Client Configuration 1 **Laptop**	Intel Core2Duo Processor, 4GB memory, display resolution 1280x800, Windows 7 Professional, Microsoft Remote Desktop Connection (RDP), VMware View Client 4.6 (PCoIP)
Client Configuration 2 **Zero Client**	Samsung NC240, 24" Monitor w/ built-in Teradici zero-client, display resolution 1920x1080, Teradici Firmware 3.3.1 (RDP and PCoIP)
Client Configuration 3 **Nettop**[8]	Intel Atom Processor, 2GB memory, display resolution 1024x768, Windows 7 Professional, Microsoft Remote Desktop Connection (RDP), VMware View Client 4.6 (PCoIP)
Client Configuration 4 **Mobile**	Apple iPad2, A5 Processor, 512MB memory, display resolution 1024x768, iOS 4.3.1, VMware View Client (PCoIP), Wyse Pocket Cloud (RDP)

2. Usability Metrics

For a more desirable and seamless user experience, the visual feedback for input activity, including keyboard strokes and mouse movements and actions, is of the essence. The most important factor here is the time that is required by the system to provide visual

[8] Nettop is a marketing term derived from Netbook (= low-cost mobile computer tailored for use with web-applications due to limited computing power) and Desktop; it describes a small form-factor, energy-efficient desktop with limited computing power.

feedback in sufficient form for user actions. In predominantly graphical user interfaces of windows style, the selection and movement of objects for drag-and-drop operations requires a bounded maximum feedback time in order to be acceptable by the user. For input operations that require a precise mouse movement, the maximum acceptable time can be considered shorter than the maximum time for operations that, for example, open or close an application. The delay between user action and (visual) feedback of the system is primarily determined by the delay (round-trip-time) in information exchange – primarily (in computer networks) the packet latency from client to server and back. Unless the desktop operating system or the application itself suffers from performance issues, the delay between input and feedback within the (server) can be neglected.

This feedback time, or responsiveness, aspect of usability is never completely objective, but rather depends on the user's personal preferences, experience, and task at hand, to name just a few factors. Consequently, no generally accepted point value can be operationalized as a reference or metric to assess a suitable feedback time for such systems. Instead, qualitative, soft measures are required to capture this quality of a VDI implementation. Reference for the level of usability is: the experience a user makes when working directly on a dedicated desktop at the workplace, unconstrained by the network performance, with regard to interaction with the desktop environment. If the experience with the virtualized desktop is perceived to be on the same level, and user productivity is not negatively impacted, then the user perception is determined to be 'seamless." For the assessment, quality levels of a Likert-like scale listed in Table 5 are used to categorize the user experience.

Table 5. User experience categories

5 – 'seamless"	++	User experience is on par with dedicated desktop in workspace.
4 – "acceptable"	+	User experiences slight differences to dedicated desktop in workspace in certain aspects of task; productivity is not diminished; interaction requires minor adaption to system.
3 – "limited"	**0**	User clearly experiences differences to dedicated desktop in workspace; productivity is constricted, not suitable for permanent work; interaction requires some adaption to system.
2 – 'strongly limited"	**–**	User perceives virtual desktop as usable only for short periods of time for specific, limited tasks; productivity is strongly impacted; interaction requires strong adaption to system.
1 – "unusable"	**– –**	User perceives virtual desktop as unacceptable for use.

C. USER LOAD SCENARIOS

As described earlier, the bandwidth demand for a seamless experience of the virtualized desktop strongly depends on the task being performed by the user. For this experiment, three task sets have been identified to represent practical categories of computer activities, primarily varying in the intensity of desktop interaction and display content changes involved. These task sets include only short activities, as the primary concern of this test is to assess the user experience under various bandwidth and latency constraints, not a performance measurement of the virtual desktop per se.

1. Basic Task Set

The user opens a PDF document (NPS student handbook), maximizes the viewer application, and scrolls through the document page-by-page from beginning to end by

clicking the scrollbar, and waiting for the new section to be displayed completely before scrolling further. The user then activates the application's search function, searches for the word 'student," and clicks forward through all search results one by one. The user closes the viewer application.

As a second task, the user opens a text document (NPS student handbook in Word-format), scrolls to a section in the middle of the document, and adds a few words to a new paragraph. The user then highlights one paragraph, and moves the selected paragraph behind the following one. After finishing, the user closes the word processor application.

2. Complex Task Set

The user opens a network diagram drawing in full screen mode, selects several objects, and aligns them in a new way (e.g., horizontally or vertically). He continues to add a few more objects from the device palette, connects them to the existing objects, manipulates selected properties, and conducts additional drag-and-drop operations in the diagram. The user then changes to the presentation application, opens an existing presentation, and manipulates objects on several slides, including rotating and resizing. Next, the user opens a software development environment, loading a small project. He browses through selected files of the source code, adds and deletes code elements, and starts a compilation run with execution. After finishing, the user closes all three applications.

3. Multimedia Task Set

The user opens a presentation document with animations and starts the presentation in full screen mode. After clicking through the presentation slide-by-slide, the user closes the presentation application. Then the user opens and plays a 60sec video file (534x300pixel, 29fps, U.S. Navys "All Hands Update"). Afterwards, the user opens and plays a 30sec high definition video (1280x720 pixel, 29fps, Microsoft sample video "Wilderness"). After the video is finished playing, the user closes the video player application.

D. EXPERIMENT SETUP

The principal setup for the experiment includes the virtual desktop environment, selected client devices, the network emulator component, and recording and analysis components. The virtual desktop environment includes five host computers, network-attached storage, and management server, as well as one switch for VDI internal traffic, and one switch connecting the virtual machines to the production environment. Both the network emulator and the recording and analysis component consist of a single server with multiple network interface cards, and the respective operating system / application for its functionality.

1. Virtual desktop infrastructure

The five host computers for the virtual desktop infrastructure are Dell Poweredge 1855 blades in a blade chassis providing consolidated power, cooling, and network connectivity for each blade's two network interface cards. Each Poweredge 1855 blade is equipped with 2 Xeon 3.6 GHz processors, 16GB of memory, and 2x 146GB SCSI hard drives in RAID1 (mirror) configuration. VMware ESXi 4.1.0 was installed as a "bare-metal" hypervisor on each blade to enable the execution of multiple virtual machines on each host – one network cards connected to the VDI management network, and the other one to the production network (ERN) each through a 48 port Gigabit Ethernet switch. To manage the host cluster of ESXi hypervisors, VMware vSphere was installed on a Dell Poweredge 1955 server (which at the same time acted as controller for the network attached storage device EMC2 AX150i), providing access to a total of 6TB storage (configured in three RAID1 partitions with total available storage of about 4TB) through four Gigabit network connections.

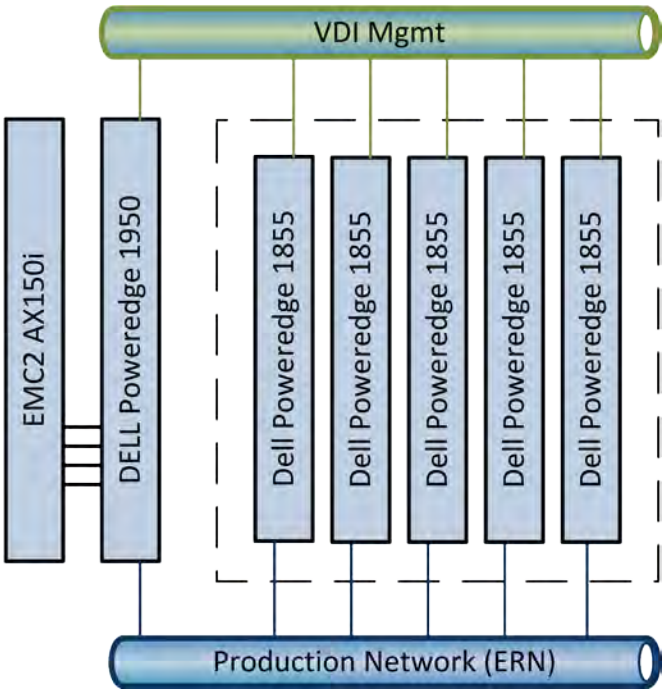


Figure 13. VDI infrastructure schematic

The VMware vSphere server application is accessed via a VMware vSphere client, which provides a graphical interface to manage the virtualization environment (cloud). The hypervisor runs on a physical host, provisioning the resources to run several virtual machines on one physical hardware instance. For management purposes, multiple hosts are combined into logical "clusters," and clusters can be combined into "data centers," all manageable via the vSphere Server. Virtual machines can, within certain restrictions, be cloned and migrated freely within the virtual environment, enabling flexibility in provisioning and load-balancing. Figure 14 shows a snapshot of the management client interface with an overview of the virtual resources and their statuses; the tree view on the left side depicts the hosts and their respective virtual machines, running several different operating systems.

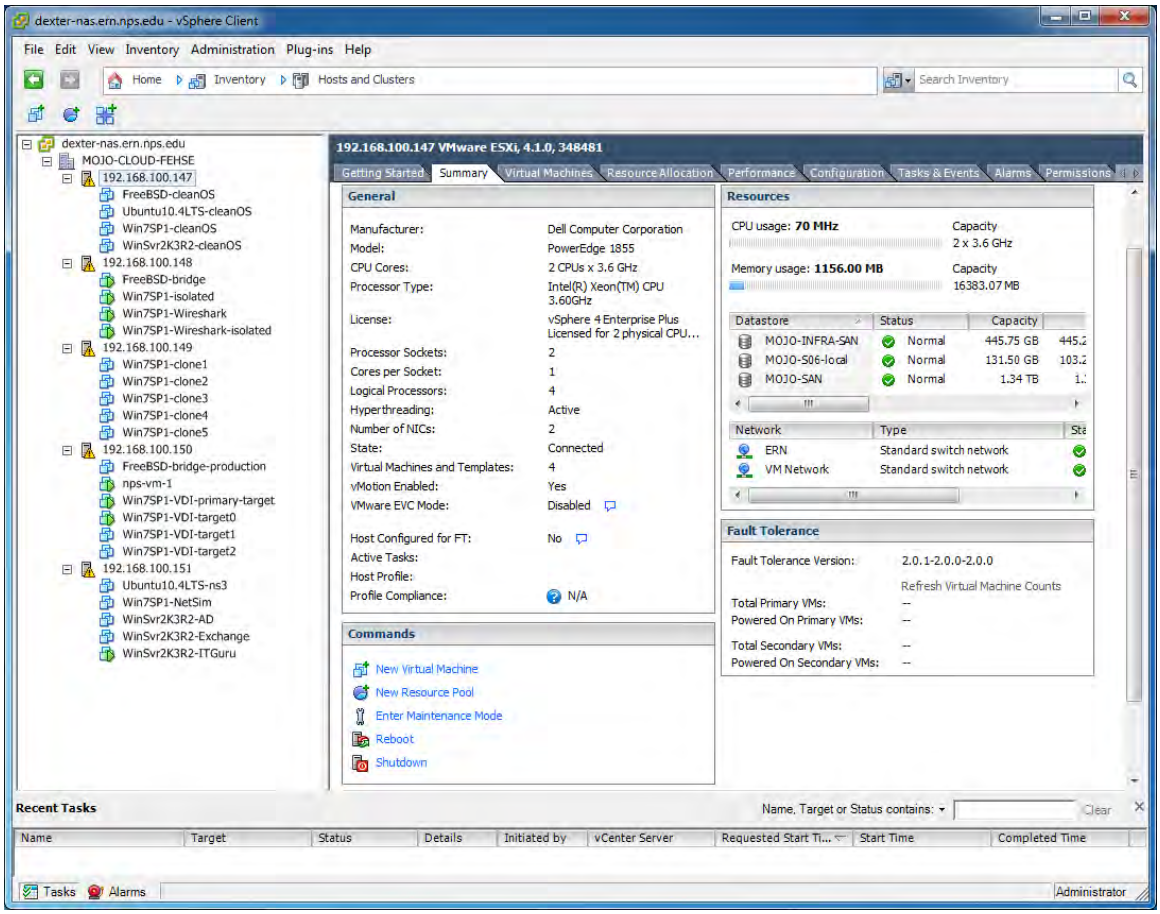


Figure 14. VMware vSphere client snapshot

2. Network Bridge – Bandwidth and Latency Emulator

Several sophisticated network emulators/simulators, both software and hardware, are commercially available to create (validated) conditions of live networks in laboratory environments. For the purpose of this experiment, however, freely available tools are sufficiently functional, as only limits in bandwidth and traffic delays are of concern. The basic principle for most of these simulators is that traffic between two local network cards is internally shaped to artificially impose bandwidth limits and transportation delays. One such representative (available as open source component and part of several open source operating systems), is "dummynet" [62]. Part of the current FreeBSD 8.1 kernel, dummynet is manipulating the internal IP stack as an extension to the firewall daemon.

FreeBSD 8.3 was installed on an older Dell Poweredge 2800 Server, which provides two internal Intel Gigabit network cards. The network was configured as a bridge between the two network cards in promiscuous mode, thereby allowing all network traffic to pass transparently and bi-directionally from one interface to the other. One of the FreeBSD firewalls, "ipfw," was activated to have access to all packets passing through the IP stack. However, since no firewalling is required, the firewall was configured "open" by flushing all default firewall rules.

Dummynet is configured through ipfw commands, as the module becomes part of the firewall after loading it (either during boot or during run-time). It provides access to the network data stream by "pipes," which can be configured to introduce bandwidth limits and latency.

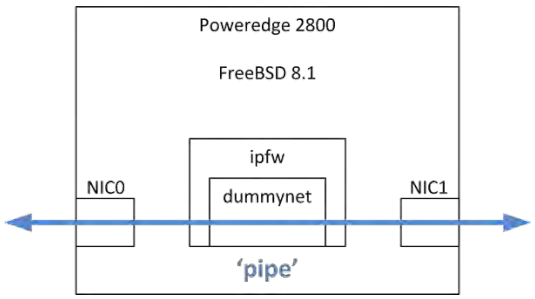


Figure 15. "dummynet" pipe principle

Which packets pass which pipe is determined by firewall rules, expressed in a manner similar to common allow / deny rules. This allows for the creation of different treatments / shapes of traffic depending on protocol, port, direction, and so forth – thereby simulating, for example, asymmetric network connections like asymmetric digital subscriber lines (A-DSL).

Dummynet pipes are configured by ipfw commands, which determine bandwidth, and delay and loss for each pipe. The command

```
ipfw add 10000 create pipe 1 ip from any to any
```

configures all IP traffic to be passed through pipe 1. Note that “10000” determines the position in the numbered firewall rule set, which is processed in order (the first matching rule is executed).

The pipe itself can now be configured for bandwidth and latency constraints. The command

```
ipfw config pipe 1 bw 2Mbit/s delay 20ms
```

configures pipe 1 with a bandwidth limit of 2 Mbit/sec and a latency of 20 milliseconds. The configuration of the dummynet pipes can take place dynamically and does not require the firewall or the network interfaces to be re-started or re-initialized, thereby ensuring continuous operation.

To validate the dummynet settings for the measurements, two tools have been used: “ping” [63] to measure the delay incurred by the bridge, and “iperf” [64] to assess the available bandwidth. The configurations of the “network condition simulator” have been determined to be correct throughout the experiment (within the expected margin of error). Throughput performance of the bridge without configured impediments has been found to exceed 100Mbit/sec as available bandwidth, and the processing delay was found to be typical at 2ms, and less than 5ms even under full traffic load.

3. Network Recording and Analysis

The network traffic recording and analysis component of the setup was realized with a single server equipped with three network interface cards, running the freely available analyzer Wireshark [65] in version 1.6.1 under Windows 7. The onboard network card was connected to the production network, and the two additional Intel Pro/100S Gigabit network interface cards were configured in promiscuous mode without IP assignment as network taps, and connected to the two switches separating the physical network segments bridged by the network condition simulator. To capture all relevant traffic, the respective ports on the switches were configured as trunk ports, mirroring the traffic of the ports of interest.

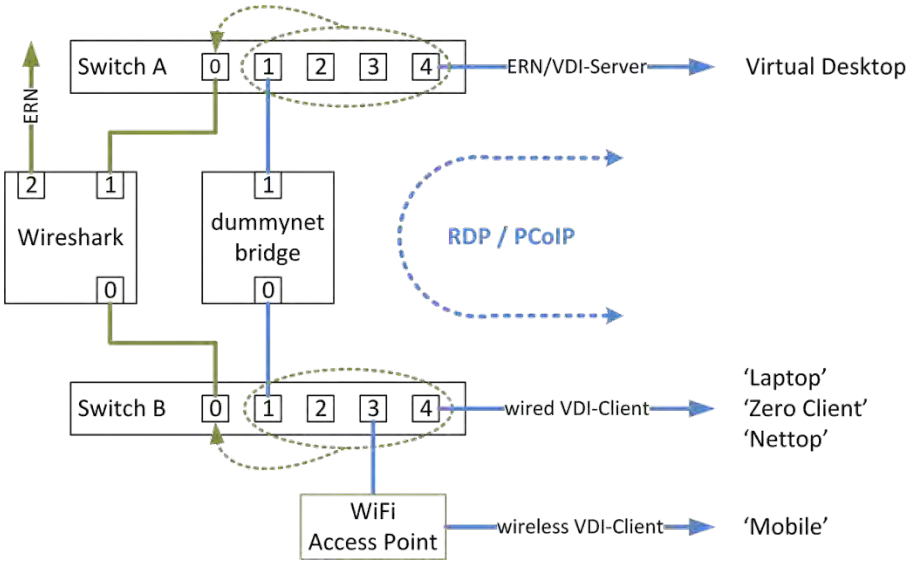


Figure 16. Experiment setup schematic - network simulator, analysis components, VDI client/server connection

By tapping both sides of the bridge, it was possible to easily confirm latency values through the bridge by comparing packets captured on the different ports, as both recordings were made based on the same system clock, without the need to synchronize time sources.

E. MEASUREMENTS

For each of the client devices #1 to #4 (Laptop, Zero Client, Nettop, and Mobile) test runs with both available protocols (RDP, PCoIP) where conducted for each task set (Basic, Complex, Multimedia), resulting in 24 test groups.

Within each test group specific bandwidth / latency conditions were assessed with regard to the user experience, using the aforementioned categories (5 – 'seamless," 4 – "acceptable," 3 – "limited," 2 – 'strongly limited," 1 – "unusable"). For the bandwidth limits, the following values were chosen to reflect typical real values for connectivity: unlimited (resulting in the maximum throughput of the bridge, measured at about 100Mbit/sec), 25Mbit/sec (the common vendor suggestion for the provisioning of networks), 10Mbit/sec, 5 Mbit/sec, 2Mbit/sec, and 1Mbit/sec. Measurements of the achieved

maximum bandwidth to and from the mobile device resulted in an average effective transfer rate of 25Mbit/sec. The device had exclusive connection to the access point, but the noisy, high-frequency environment in the laboratory – in combination with relatively low processing power – limited the throughput. Tests runs with no limit on bandwidth enforced by the bridge have therefore not been conducted.

Derived from the scenarios of the hypothetical project, latency values of 0ms (direct connection), 20ms (common experienced latency for on-campus wireless and off-campus wired connections), 45ms (CONUS distant learning), 140ms (OCONUS distant learning with good connectivity), 300ms (intercontinental long-distance connection or delay through, for example, several changes in communication medium).

To gain a quick overview of the results of the analysis of each test group, a depiction was used to present the user experience category for each combination of impediments in one graphic. As the decrease in available bandwidth and the increase in latency both result in degraded user experience, the results were expected to reflect the characteristics shown in the generic graph of figure xyz. The question mark in the center of the generic graph represents the test results, showing which combination of both impediments will dominate to impact user experience.

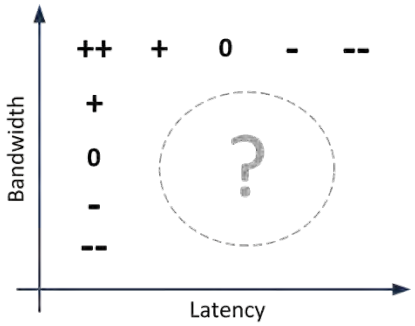


Figure 17. Generic graph test results

The detailed results for all 24 test groups can be found in Annex A1 to A4.

F. ANALYSIS

The experiment has been conducted with a single person as user to assess the quality of the user experience, and is therefore biased towards the expectations of that

one person. In addition, the user experience is influenced by the general performance of the VDI server (tests against different server configurations, e.g., with virtualization support by the processor or dedicated accelerator hardware, will result in measurements that vary from the ones in this thesis). Conclusions drawn from the results therefore cannot be generalized too far; they reflect tendencies rather than absolute facts.

For each of the client devices, it was found that no difference in user experience could be measured for available network bandwidths larger than 25Mbit/sec, for both RDP and PCoIP. Although the load on the unrestricted network was observed to reach traffic bursts of 50-75Mbit/sec, no significant increase in user experience quality in comparison to network conditions with a 25Mbit/sec bandwidth ceiling could be found. This finding matches the suggestions of different vendors for network provisioning [37], [47].

Inference 1: For bandwidths larger than 25Mbit/sec, the dominant factors for user experience are latency and task; further increase of bandwidth does not gain significant improvement. Models of VDI traffic loads can be limited to a maximum of 25Mbit/sec server to client data transfer.

The expected strong relation between user task and user experience was confirmed, regardless of the client device. This expected finding emphasizes the importance of defining user tasks selected for comparison of VDI performance and experience. How far the performance of the VDI server influenced the overall disappointing user experience in the multimedia task set could not be conclusively determined, since only one VDI server hardware platform was available for this thesis. In general, the utilized client devices (“Laptop,” “Nettop,” and “Mobile”) are fully capable of displaying streaming media with properties similar to the ones used during the tests. The bandwidths required to do so are in the range of 25Mbit/sec, as found in several trials that were conducted to have a closer look into the causes of the observed test results. It seems reasonable to assume that the optimized compression and streaming protocols for the specific purpose of media consumption allows for a better user experience. Compression of display information for VDI occurs “on-the-fly” on the server, as the protocol neither can presume any future changes of the display, nor buffer (and thereby delay) the display information.

Inference 2: The type of user task and the resulting changes of display information determine the bandwidth necessary for an acceptable user experience. Models of VDI traffic will have to reflect user tasks of the type and level of detail that allow meaningful interpretation of simulation results.

User experience was found to depend significantly on available bandwidth for latency values below 50ms. Slight impacts in usability can be recognized in certain cases for latency around 140ms, mainly in the complex task set, which requires precise cursor movement and high quality perception of visual feedback. Latency larger than 300ms limited the user experience to a maximum perception of "limited," and in most cases resulted in user experience that was determined to be 'strongly limited" or "unusable." However, at what latency value the user experience is impacted and rendered unsatisfying will vary from individual to individual.

Inference 3: The correlation of latency and bandwidth influence on user experience is limited; below certain values for latency, bandwidth is dominating the quality of user experience, and above certain values the user experience is diminished regardless of the available bandwidth. The analysis of VDI simulation results has to take the critical latency values into consideration.

Although the tests are not, per se, suitable for the comparison of the two VDI protocol representatives, it was found that for the basic task set, both protocols performed very comparable. In the multimedia task set, slight advantages of PCoIP have been observed, regardless of bandwidth and latency constraints. A clear difference was found with the complex task set, in which PCoIP outperformed RDP by providing a slight advantage under the comparable network bandwidth conditions. The advantage in perceived user experience can probably be attributed to the design of PCoIP. The protocol implements a progressive build strategy which transmits the display content gradually. First, display data is prioritized and delivered with lossy, high-rate compression in lower quality, and then gradually with lossless compression up to "perception-free" full quality. The user perceives this trade "quality against speed" as positive when their interaction with the desktop requires short feedback times.

Inference 4: Although models for VDI traffic loads should be able to use simplified, statistical data generation to emulate VDI traffic in general, without being concerned with the respective protocol implementation strategies, the analysis of the simulation results will have to take into account the different protocols and task sets emulated in order to assess the user experience resulting from the modeled network.

During the test runs, the network traffic between VDI server and VDI client was recorded. Quick analysis of the traffic-flows shows, as expected, the direct correlation between display changes and amount of transferred data from VDI server to VDI client. In both the basic and complex scenarios, the network traffic occurred in "bursts" to convey the rendered information, followed by low throughput traffic when the user paused in his activities. In the case of the multimedia task set, the full bandwidth was consumed during video replay, and the intensity of traffic showed nearly no interruption. Traffic recorded from VDI client to VDI server was confirmed to require minimal bandwidth, since no additional devices were connected to the client device (e.g., web-cam, USB storage, or microphone) adding traffic.

Inference 5: Recorded live-session VDI network traffic can be utilized to generate the statistical properties of data streams emulating VDI traffic loads in simulations, without the need to emulate the complex protocols themselves.

G. SUMMARY

Preliminary experiments show that the methodology proposed for the experimental part of this thesis can generate valuable information needed to derive practical, problem-appropriate models in network simulations for cloud computing implementations.

The proposed methodology follows a modular approach, breaking out the cloud service infrastructure into the following components: investigation, network emulation, network traffic recording, and user interface (client). Each of the components can be exchanged for one that may be better suited to different experiment objectives or environ

ments. For example, if a more sophisticated or accredited network emulator is desired, it can replace the one presented here without requiring changes to the rest of the components.

This flexibility also allows for other uses. For example, as component for mobile virtualization setups, since they are currently under research at NPS for use in deployed military environments or disaster recovery situations. During the setup for this thesis, the recording and analysis component, as well as the network emulator, were setup within the cloud on virtualized machines, allowing simplified access for software configuration purposes. The cloud-internal network switching configuration was used to logically separate selected virtual machines. Another virtual machine was configured with two virtual network cards to bridge the separated virtual machines back into the operational network under controlled conditions, thereby allowing the emulation of adverse conditions for connecting clients without the need of additional equipment (see Figure 18, which shows a screen capture of the virtualized experiment setup as viewed within the cloud management console VMware vSphere Client).

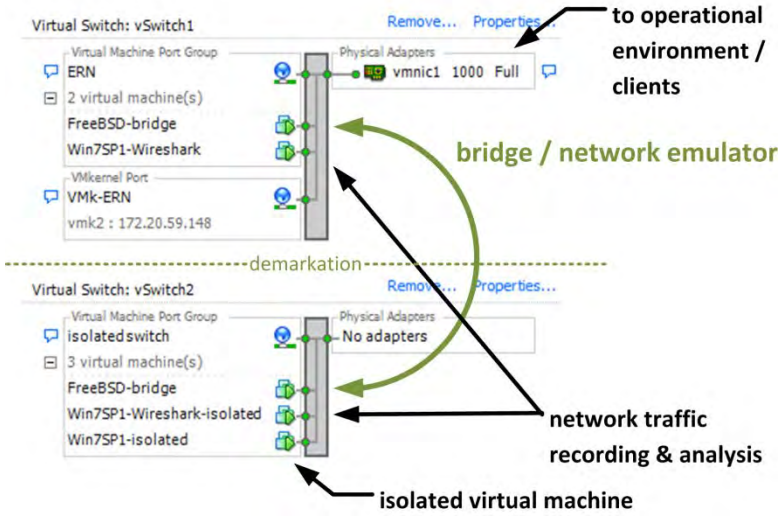


Figure 18. Virtualized experiment setup

Including such configuration within a cluster of virtualized machines allows for the optimization of the internal configuration of the VDI infrastructure for anticipated conditions in the field prior to actual deployment. In addition, it leverages the possibility

of having users experience different network environments and their impact on their virtualized working environment, enabling improved expectation management.

The experimental setup is not limited to conducting measurements on a single VDI client/server connection, but can scale to multiple concurrent sessions through a bandwidth/latency impaired network, so the load balancing mechanisms of the respective VDI protocol implementation can also be observed and analyzed for inclusion in derived simulation models.

V. CONCLUSION, FURTHER RESEARCH AND RECOMMENDATIONS

A. CONCLUSION

This thesis approached the question of how current computer network modeling and simulation tools support the emerging cloud computing paradigm. The assessment of selected modeling and simulation tools utilized a well-structured approach following the initial steps of the Unified Development Process, generating a set of functional and non-functional requirements derived from pragmatic use-cases. The assessment resulted in the discovery of a capability gap that precludes the successful use of modeling and simulation in early stages of projects for cloud computing implementations.

With the discovery of the existing capability gap in mind, the research for this thesis continued to propose a modular designed methodology to measure the essential properties necessary to develop appropriate cloud computing network traffic models. Categories for the quality of user experience were introduced, and appropriate scenarios for test cases were developed. As a proof-of-concept, the proposed methodology was applied to a virtual desktop environment created for test and research purposes. The methodology was found to be problem-appropriate and suitable to gather information necessary to develop practical cloud computing traffic models.

B. FURTHER RESEARCH AND RECOMMENDATIONS

1. Network Emulator Virtual Machine

During the configuration of the experiment, it has been found to be tremendously useful to use virtual machines instead of physical hardware when assessing several available open source tools for their suitability in tests. With the increased interest of agencies to employ virtualized infrastructure components in scenarios of disaster relief, flexible law enforcement response, or military operations, it is suggested that they include the capability to emulate adverse network conditions into the virtualization infrastructure as virtual machines. This will allow administrators to configure the virtualization setup for anticipated conditions in the field prior to operational deployment. In addition, users are

able to assess the capabilities and performance – and get adapted to the working environment they can expect in the field – without having to worry about discrepancies between laboratory configuration under ideal circumstances, and adverse conditions once deployed.

As this approach calls for the use of best practices and the sharing of experience to derive at a standardized, accredited solution, it is further recommended to develop and issue a pre-configured network emulation environment as a virtual machine image or a bootable disk image based on open source operating system distributions and applications. Such a virtual machine image should provide network emulation functionality – wrapped preferably in an easy-to-use, web-based control interface to lower the skill requirements for practical usage.

2. User Experience Categories

The success of cloud computing solutions depends decisively on acceptance by users. Without a satisfying user experience, it is predictable that the loss of productivity – in comparison to existing solutions – will counter all advantages and savings gained by migrating to cloud computing solutions.

The user experience categories proposed and used in this thesis for the proof-of-concept experiment are not sophisticated enough to derive generalizable results and conclusions for broader application and use. It is therefore suggested to conduct further research (for example, in the field of human machine interfaces), to propose practical user experience categories and definitions, and to create a reference framework for comparable test results across larger user communities.

To assess and compare the performance of applications and computing configurations, standardized benchmark tests have been in use throughout the IT industry for quite some time. These benchmarks allow the comparison of test results on an objective basis, and can identify strengths and weaknesses as the basis for continued development and improvement. In general, such benchmark suites are executed in an automated manner, without intervention or interaction with users, to keep the influence of independent variables at a minimum.

As the influence of network latency on the user experience has been found to be an important factor, an investment in further research should be made to investigate the possibilities and limitations of automated user experience assessment. Such an approach would allow for the operationalization of the independent variable “human being” into a controlled variable, and the achievement of objective test results, while decreasing the need for larger user groups in elaborate and expensive test environments.

3. Standardized Task Sets

The task sets used during the proof-of-concept experiment have been the result of the author’s personal experience and suggested use cases from vendors of virtualization solutions (e.g., [38], [47]). The variance in the composition of such task sets means that the results of tests conducted with different task sets can be compared only with caution, if at all. Virtual desktop implementations are gaining market share, and the federal government is considering mandates for cloud computing solutions for all new IT projects [1]. It would be wasteful to have each entity in the federal or local government define sets of tasks specifically for their own implementation projects, while not being able to leverage the experience and proven best-practices of other members in the same community of interest.

The purpose of task sets in the context of virtualized infrastructure is to baseline the various activities users are executing in their computing environment to derive practical load scenarios to assess overall system performance. It is of less importance, from the aspect of network performance, which exact text processor is used when the user conducts text editing. Of significance, is how much rendered display information needs to be transferred (bandwidth), or what required level of visual feedback is necessary (latency).

It is therefore suggested that standardized task sets representing typical user behavior and interaction should be developed – to be used to assess cloud computing performance on a normalized base, allowing for a true comparison of the various offerings by the market. For cases in which these standardized task sets are deemed not directly

applicable, small scale comparison tests can be conducted to find the closest match between personalized and standardized task sets, without the need to conduct full-scale measurements.

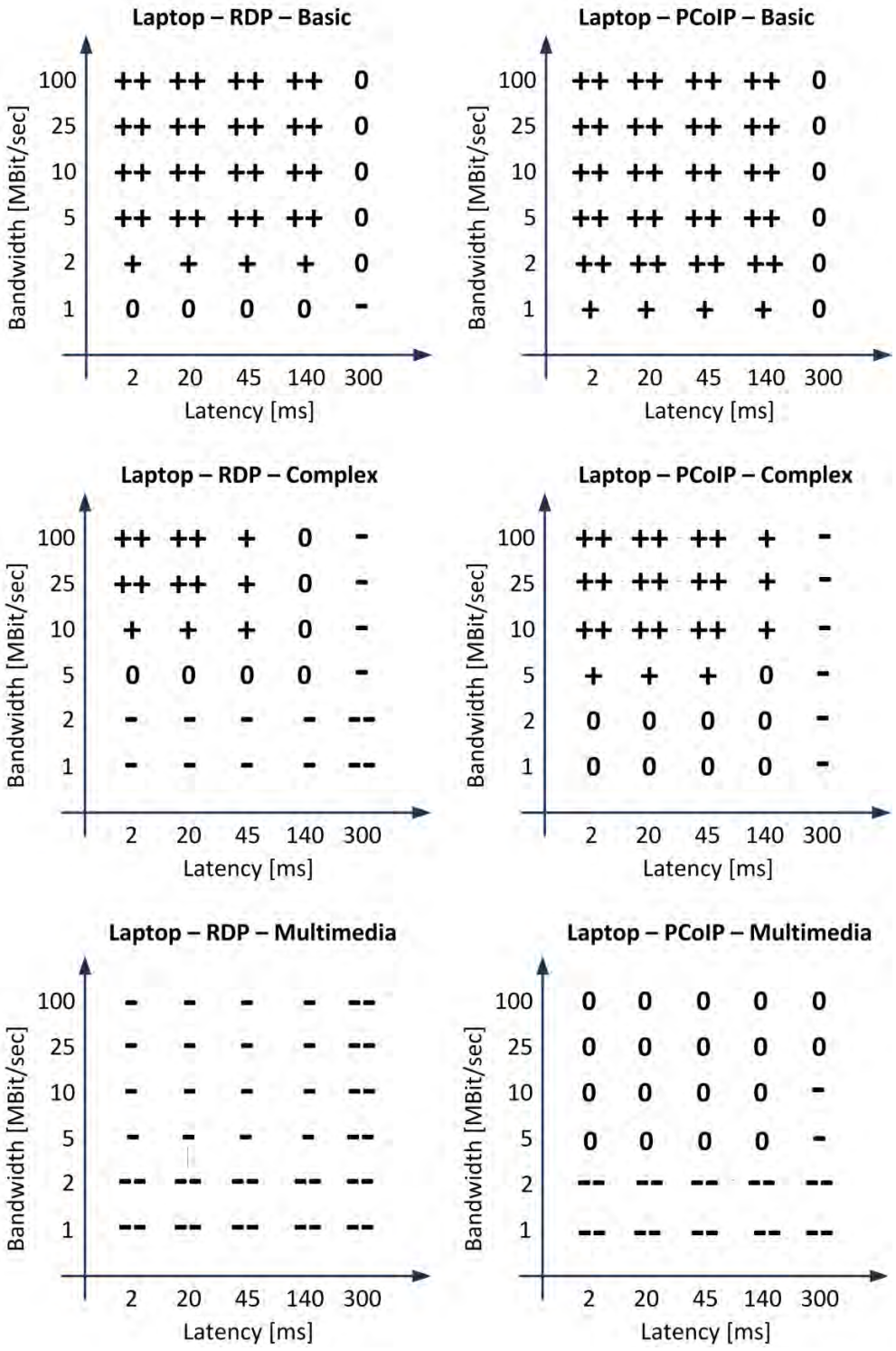
4. Development of Models for Simulation

This thesis discovered the currently existing capability gap for computer network modeling and simulation tools with regard to cloud computing implementations. As of now, the full potential of modeling and simulation cannot be applied to early stages of cloud computing projects in practical terms. This shifts the potential risks to later project phases, and endangers successful outcomes.

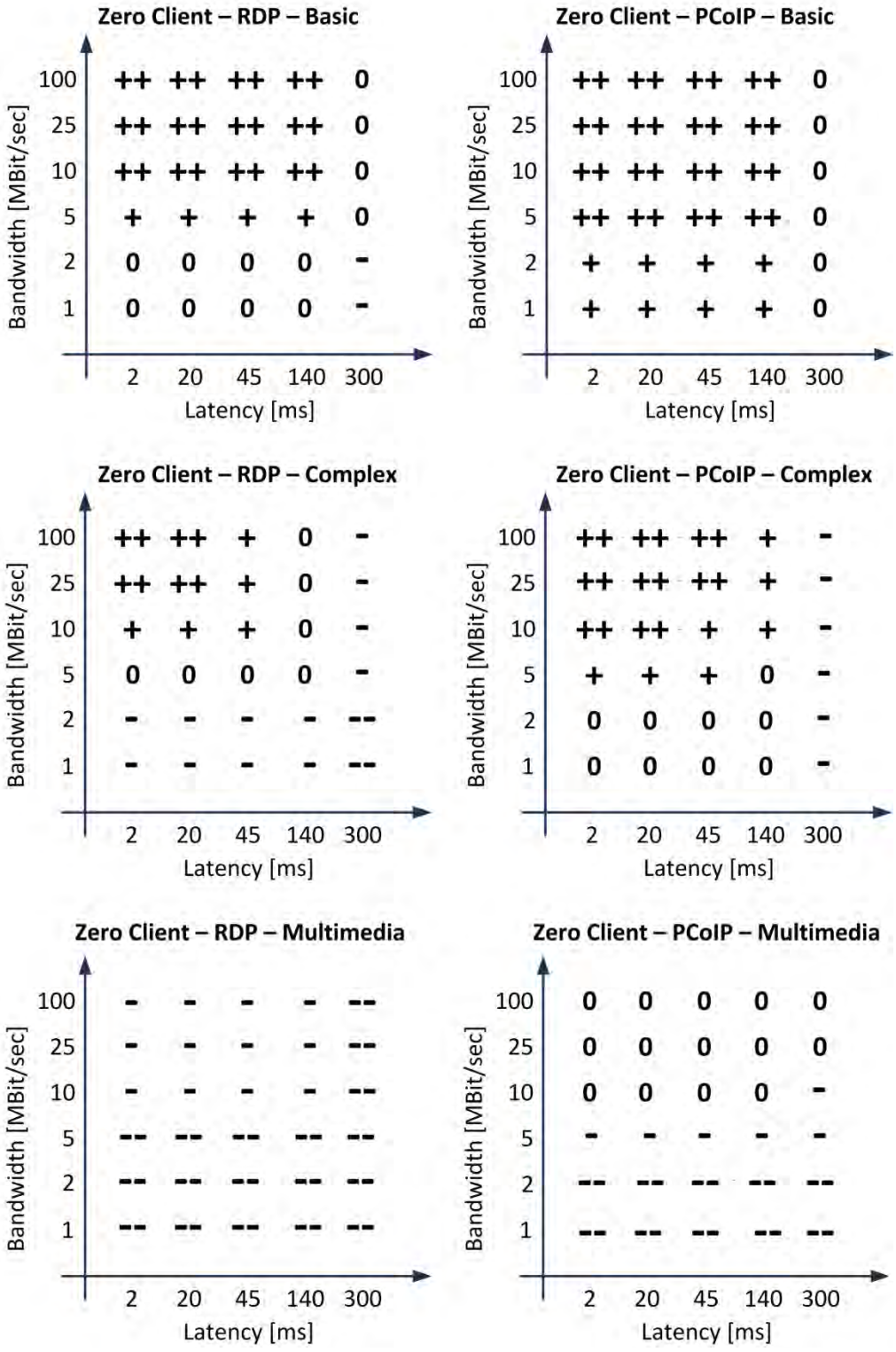
It is consequently recommended that the proposed methodology be applied to derive the essential properties of network traffic in cloud computing implementations, and develop appropriate models that can be used in network modeling and simulation applications. As a first increment to such an effort, it is further suggested that recorded network traffic be analyzed with regard to its statistical properties. The respective results should be used to configure the existing, generic models for traffic loads within the ITGuru library to resemble, as closely as possible, the properties of the live data. The quality of such configured models should then be assessed in comparison tests with live networks, to evaluate if they can be successfully used to support the development of cloud computing implementations without the need for an early commitment to – or procurement of – physical components on large scale.

APPENDIX

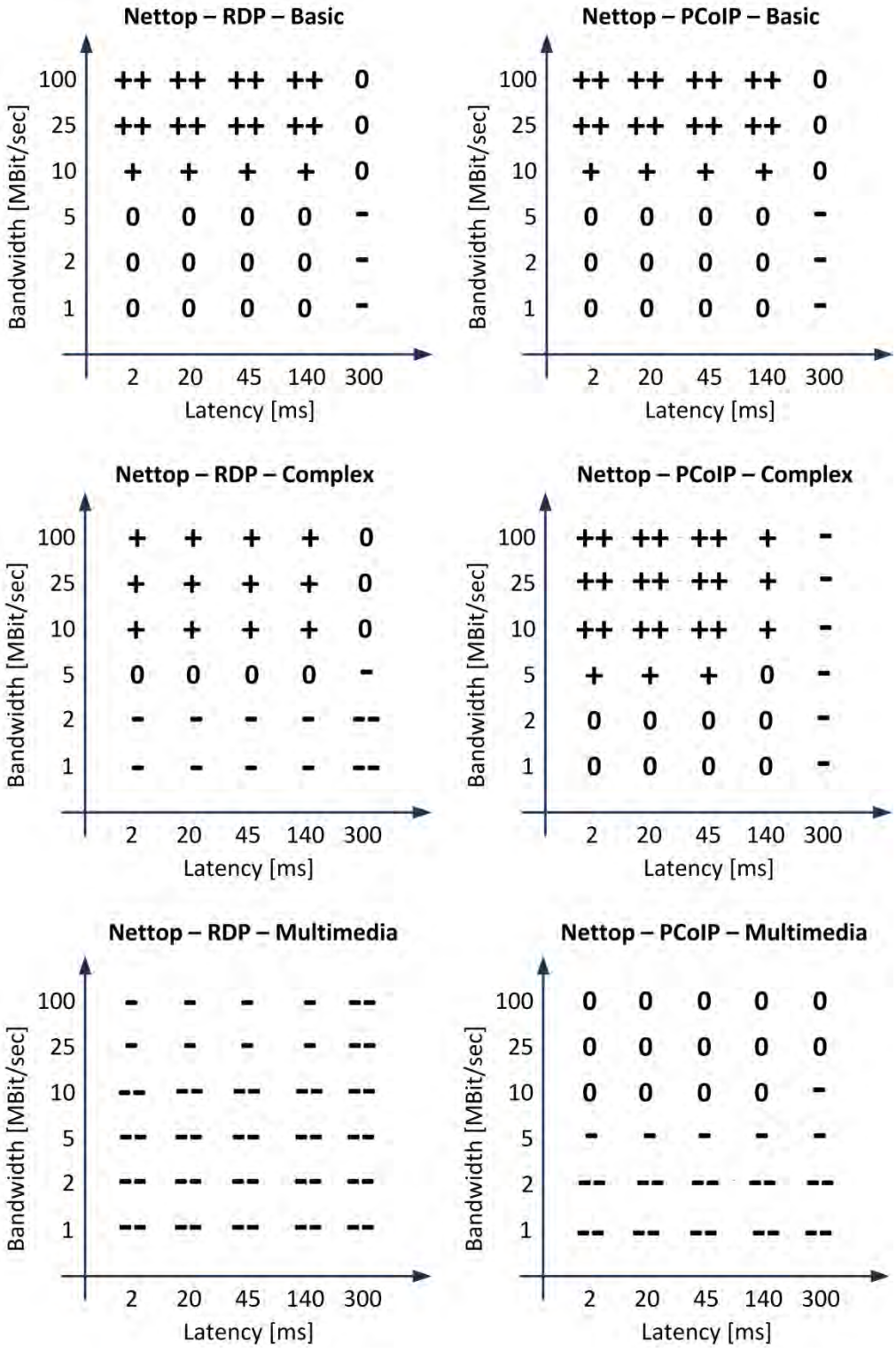
A. EXPERIMENT RESULTS "LAPTOP"



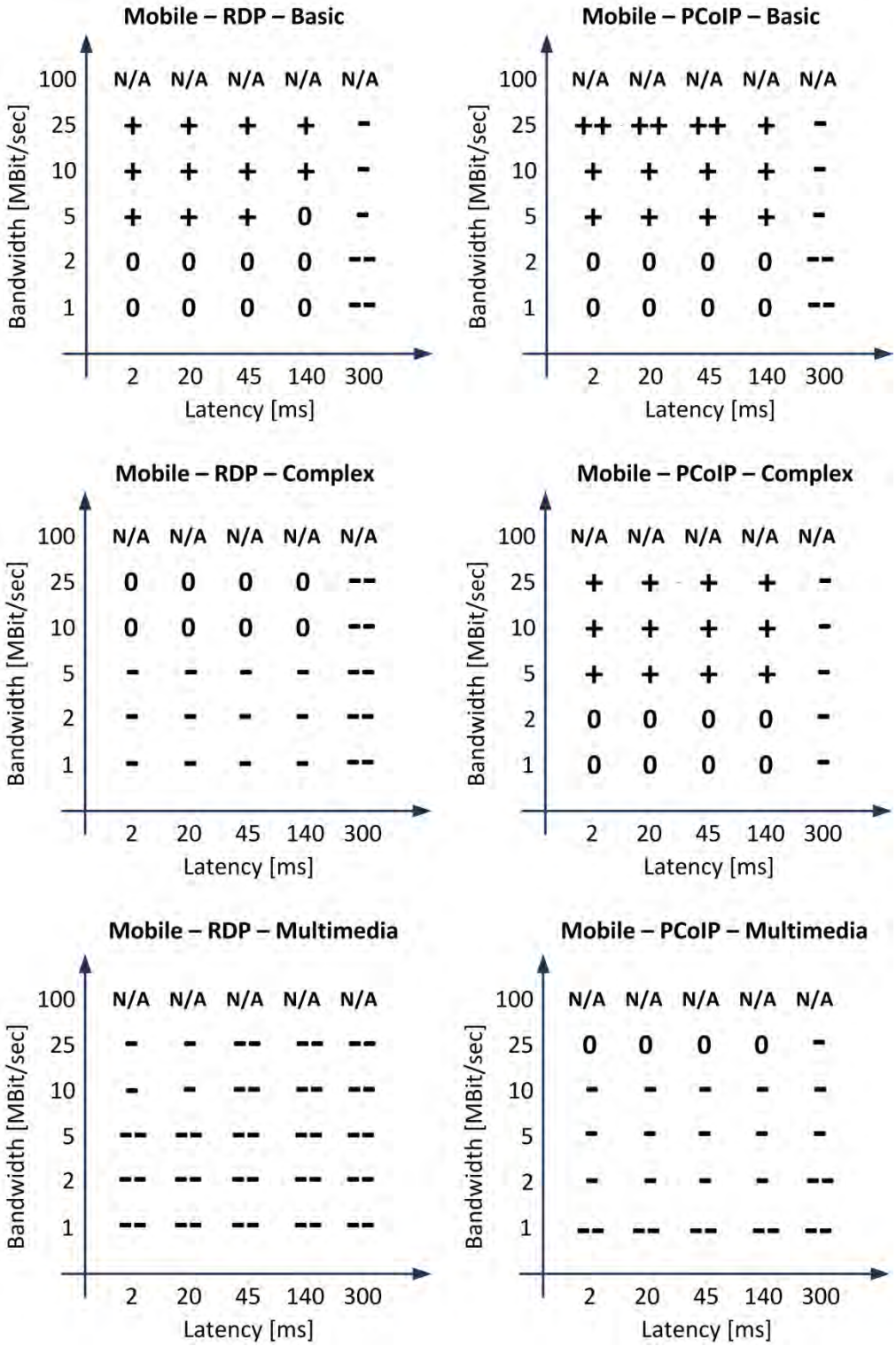
B. EXPERIMENT RESULTS "ZERO CLIENT"



C. EXPERIMENT RESULTS "NETTOP"



D. EXPERIMENT RESULTS "MOBILE"



LIST OF REFERENCES

[1] Vivek Kundra, "25 Point Implementation Plan to Reform Federal Information Technology Management," Office of the U.S. Chief Information Officer, Washington, D.C., Policy 2010.

[2] Deputy Secretary of the Department of Defense, "A New Approach for Delivering Information Technology Capabilities in the Department of Defense," Washingtin, D.C., 2010.

[3] The Association for Enterprise Information, "Industry Perspectives in the Future of DoD IT Acquisition," Arlington, VA, 2010.

[4] Craig Larmann, *Applying UML and Patterns*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education.

[5] Kendall Scott, *The Unified Process Explained*, 1st ed. Boston, MA, USA: Addison-Wesley, 2002.

[6] Dan Farber. (2008, September) Outside the Lines - Oracle's Ellison nails cloud computing. [Online]. http://news.cnet.com/8301-13953_3-10052188-80.html

[7] Bobbie Johnson. (2008, September) The Guardian - Technology News. [Online]. http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman

[8] NIST - National Institute of Standards and Technology. (2011, January) NIST Cloud Computing Collaboration Site. [Online]. http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/WebHome

[9] NIST - National Institute of Standards and Technology. (2011, January) The NIST Definition of Cloud Computing (Draft). [Online]. http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf

[10] NIST - National Institute of Standards and Technology. (2011, January) Cloud Architecture Reference Models: A Survey. [Online]. http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/Meeting4AReferenceArchtecture013111/NIST_CCRATWG_004v2_ExistentReferenceModels_01182011.pdf

[11] Gerald Kaefer. (2010, May) SEI Software Engineering Institute - Library. [Online]. http://www.sei.cmu.edu/library/assets/presentations/Cloud%20Computing%20Architecture%20-%20Gerald%20Kaefer.pdf

[12] IBM Corporation. (2011, February) IBM Cloud Computing Reference Architecture 2.0. [Online]. https://www.opengroup.org/cloudcomputing/uploads/40/23840/CCRA.IBMSubmission.02282011.doc

[13] Google Inc. (2011) Google Docs. [Online]. http://docs.google.com

[14] Microsoft Corp. (2011) Microsoft Office Web Apps. [Online]. http://office.microsoft.com/en-us/web-apps/

[15] Amazon Web Services LLC. (2011) Amazon Elastic Compute Cloud (Amazon EC2). [Online]. http://aws.amazon.com/ec2/

[16] Google Inc. (2011) Google App Engine. [Online]. http://code.google.com/appengine/

[17] Amazon Web Services LLC. (2011) Amazon Web Services. [Online]. http://aws.amazon.com/

[18] Flexiant Ltd. (2011) FlexiScale public cloud. [Online]. http://www.flexiant.com/products/flexiscale/

[19] Renato Figueiredo, Peter A. Dinda, and Jose Fortes, "Resource Virtualization Renaissance," *IEEE - Computer*, no. May, pp. 28-31, 2005.

[20] Intel Corp. (2011) Intel Virtualization Technology. [Online]. http://www.intel.com/technology/virtualization/technology.htm?iid=tech_vt+tech

[21] AMD Corp. (2011) AMD Virtualization Technology. [Online]. http://sites.amd.com/us/business/it-solutions/virtualization/Pages/amd-v.aspx

[22] IBM Corp. (2011) IBM PowerVM. [Online]. http://www-03.ibm.com/systems/power/software/virtualization/

[23] VMware. (2007, July) Understanding Full Virtualization, Paravirtualization, and Hardware Assist - White Paper. [Online]. http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

[24] Citrix. (2011) Xen Hypervisor. [Online]. http://xen.org/products/xenhyp.html

[25] Micorosoft Corp. (2011) Microsoft Server and Cloud Platform. [Online]. http://www.microsoft.com/en-us/server-cloud/hyper-v-server/default.aspx

[26] VMware. (2011) VMware vSphere. [Online]. http://www.vmware.com/products/vsphere/overview.html

[27] William Stallings, *Data and Computer Networks*, 7th ed. Upper Saddle River, New Jersey, USA: Pearson Prentice Hall, 2004.

[28] Roger L. Freeman, "Bits, Symbols, Bauds, and Bandwidth," *IEEE Communications Magazine*, pp. 96-99, April 1998.

[29] Andrew S. Tanenbaum and David J. Wetherall, *Computer Networks*, 5th ed. Seattle, WA, USA: Prentice Hall, 2010.

[30] Robert E. Shannon, *Systems Simulation: The Art and Science*. Upper Saddle River, NJ, USA: Prentice Hall, 1975.

[31] Benjamin S. Blanchard and Wolter J. Fabrycky, *Systems Engineering and Analysis*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2011.

[32] Jerry Banks, "Introduction to Simulation," in *Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, 2000.

[33] *Fujimoto, Richard M.; Riley, George; Perumalla, Kalyan*. San Rafael, CA, USA: Morgan and Claypool, 2006.

[34] Karsten M. Reineck, "Evaluation and Compariosn of Network Simulation Tools," Department of Computer Sciences, University of Applied Sciences Bonn-Rhein-Sieg, Bonn, Thesis 2008.

[35] Ricki G. Ingalls, "Introduction to Simulation," in *Proceedings of the 2008 Winter Simulation Conference*, Miami, FL, 2008.

[36] Microsoft. (2011, July) Microsoft Developer Network (MSDN). [Online]. http://msdn.microsoft.com/en-us/library/aa383015(VS.85).aspx

[37] Microsoft. (2008, October) Microsoft Developer Network (MSDN). [Online]. http://download.microsoft.com/download/4/d/9/4d9ae285-3431-4335-a86e-969e7a146d1b/RDP_Performance_WhitePaper.docx

[38] Microsoft. (2010, January) Microsoft Download Center. [Online]. http://www.microsoft.com/download/en/details.aspx?id=23236

[39] Michael Harries. (2009, March) The Citrix Blog. [Online]. http://blogs.citrix.com/2009/03/05/a-hype-free-introduction-to-hdx/

[40] Citrix. (2010, December) Citrix Knowledge Center. [Online]. http://support.citrix.com/productdocs/

[41] Citrix. (2011, July) Citrix Knowledge Center. [Online]. http://support.citrix.com/article/CTX101810

[42] NComputing. (2011, January) NComputing Inc - Documentation. [Online]. http://www.ncomputing.com/docs/guides/en/guide_server_virtualization_deployment.pdf

[43] NComputing. (2010, October) NComputing - Documentation. [Online]. http://www.ncomputing.com/docs/guides/en/L300_eval_guide.pdf

[44] NComputing. (2011, March) NComputing Knowledge Base. [Online]. http://www.ncomputing.com/kb/Configuring-Firewall-and-Antivirus-for-NComputing-Products_63.html

[45] NComputing. (2010, June) NComputing - Documentation. [Online]. http://www.ncomputing.com/docs/whitepapers/en/whitepaper_enterprisevdi.pdf?mkt_tok=3RkMMJWWfF9wsRonva%2FIZ

[46] Teradici. (2011, July) Teradici - Products. [Online]. http://www.teradici.com/pcoip/pcoip-products.php

[47] Taradici. (2008, July) PCoIP® Technology User Guide. [Online]. http://www.dell.com/downloads/global/products/precn/en/remote_access_device_task_based_user_guide.pdf

[48] Internet Assigned Numbers Authority. (2011, July) IANA Internet Protocol Port Numbers. [Online]. http://www.iana.org/assignments/port-numbers

[49] Phillipe Kruchten, *The Rational Unified Process: An Introduction*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2004.

[50] International Standards Organization, *Open Systems Interconnection - Reference Model*, Edition E ed. Geneve, Swiss: ISO/IEC, 1994.

[51] Defense Advanced Research Projects Agency. (1981, September) RFC 791 - Internet Protocol DARPA Internet Program Protocol Specification. [Online]. http://www.ietf.org/rfc/rfc0791.txt?number=791

[52] Verizon. (2011, August) Verizon Business Network Statistics. [Online]. http://www.verizonbusiness.com/about/network/latency/#overview

[53] The ns-3 Project. (2011, August) ns-3. [Online]. http://www.nsnam.org/

[54] The GNU project. (2011, August) gcc - the GNU Compiler Collection. [Online]. http://gcc.gnu.org/

[55] The ns-3 Project. (2011, August) ns-3 Tutorial. [Online]. http://www.nsnam.org/docs/release/3.11/tutorial/ns-3-tutorial.pdf

[56] The ns-3 Project. (2011, August) ns-3 Doxygen. [Online]. http://www.nsnam.org/docs/release/3.11/doxygen/

[57] Tetcos Inc. (2011, August) Tetcos NetSim Software. [Online]. http://www.tetcos.com/software.html

[58] Tetcos Inc. (2011, August) NetSim Brochure. [Online]. http://tetcos.com/brochure_sv.pdf

[59] The MinGW project. (2011, August) The MinGW project. [Online]. http://www.mingw.org/

[60] Riverbed Technologies. (2011, August) WinPcap - the industry-standard windows packet capture library. [Online]. http://www.winpcap.org/

[61] OPNET Technologies Inc. (2011, August) Network Performance Management for Enterprises. [Online]. http://www.opnet.com/solutions/brochures/NPM_IT.pdf

[62] Luigi Rizzo. (2010, March) dummynet. [Online]. http://info.iet.unipi.it/~luigi/dummynet/

[63] Mike Muuss. The story of the PING program. [Online]. http://ftp.arl.mil/~mike/ping.html

[64] NLANR/DAST. (2011, June) sourceforge.net - iperf. [Online]. http://sourceforge.net/projects/iperf/

[65] Riverbed Technologies. (2011, July) Wireshark. [Online]. http://www.wireshark.org/

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Man-Tak Shing
   Naval Postgraduate School
   Monterey, California

4. Prof. Albert Barreto III
   Naval Postgraduate School
   Monterey, California

5. Dr. Daniel C. Boger
   Chairman, Department of Information Science
   Naval Postgraduate School
   Monterey, California

6. Dr. Peter J. Denning
   Chairman, Department of Computer Science
   Naval Postgraduate School
   Monterey, California

7. Dr. Bret Michael
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California

8. Dr. Thomas Otani
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California

9. Mr. John Shea
   Office of the DoD CIO
   Arlington, Virginia