A Virtualization Testbed Framework for Networked Information System

Lixin Zhu and Shuanghua Zhu

Abstract After a review on the existing testbed frameworks for networked information system, we proposed a new virtualization testbed framework which had the merits of scalability, configurability, security, programmability and efficiency. The testbed framework is defined with function components and running view in detail in this paper. The implementation methods for virtualization testbed framework are described and experimental results showed the advantage of virtualization testbed framework.

Keywords Networked information system · Virtualization testbed · Testbed framework

1 Foreword

The networked information system, which has features in network-centric topology and service oriented architecture, has become the mainstream of military information system development. To address the challenges proposed by networked information system development, many system architectures, concepts, services, protocols, and algorithms have been proposed in order to meet the technical requirements. The SOA-based and cloud computing-based system framework have been successful examples [1]. Nevertheless, before a new system architecture or protocol has been put into application, we must build testbed to do full-scale and rigorous experiments. By now, there are three types of testbed framework for networked information system [2].

The first one is simulation-based testbed framework. In simulations, models of real-world components (such as ICT infrastructure) are created, and these models are then made to interact [3]. Simulation has the advantage of being highly scalable

L. Zhu (✉) · S. Zhu
Science and Technology on Information Systems Engineering Laboratory,
Nanjing 210007, China
e-mail: rambozhuzhu@aliyun.com

Q. Liang et al. (eds.), *Proceedings of the 2015 International Conference on Communications, Signal Processing, and Systems*, Lecture Notes in Electrical Engineering 386, DOI 10.1007/978-3-662-49831-6_86

since a large number of host and device models can be run on a single physical machine (assuming simple models). However, a disadvantage is that because these models are often high level abstractions of the real-world objects, the test results can lack fidelity and may not reflect reality. Developing models which capture the complex, dynamic, and stochastic nature of networked information system requires a significant effort.

The second one is overlay network-based testbed. The overlay network testbed operates on top of production networks. The overlay network can dynamically change the topology of the network for applications under test, while the underlying network remains static. It therefore provides flexibility in configuring testbeds for different experiments. Overlays are very common method for adding functionality to an existing service, for example the internet overlays the telephone network. Other networks such as peer-to-peer or cloud services now overlay the internet. FEDERICA project [4] funded by EU has interconnected the separated network widely distributed in Europe to build a specific testbed for next generation network. With this overlay testbed, the cybersecurity, protocol, and application for future Internet could be verified. Using overlays for test networks provide size, cost, and fidelity advantages compared to other approaches. This is because testing can be performed at the same scale as the production network it overlays, real hardware is used for high fidelity, and they avoid the additional cost of building a separate test network or laboratory. However, a disadvantage is the difficulty in performing formal testing including repeatability and experiment control (since the underlying network cannot always be controlled). Another disadvantage is the potential adverse impact of the experiment on the production network.

Recently, the emulation testbed for networked information system was proposed [5], that uses a standalone physical testbed to emulate any number of different experiment configurations. Emulation is the process of mapping a desired experimental networked information system topology and software configuration onto the physical infrastructure of the testbed. While the actual infrastructure may consist of a cluster of machines, the emulation component can configure the cluster to behave as per the desired experiment topology including routing, services and C3I applications. It supports experimentation with scientific rigor. Fidelity is high due to the use of real hardware and having total control of background traffic, workloads, and general events. This contrasts with simulations where it is not always easy to know which parts of the model can be abstracted without influencing the test results. Fidelity is also higher than with overlay networks where experiments are not always repeatable due to the shared infrastructure. Virtualization is commonly used as a tool for flexible emulation. However, the lack of a united platform was the main difficulty to build a emulation testbed for networked information system, which could allocate, deploy, control, and manage this kind of virtualization testbed and its resource.

2 Virtualization Testbed Functional Requirements

The networked information system has features in complex topology, large scale, and technology diversity, which lead to the new functional requirements for the virtualization testbed. In addition, the virtualization testbed should afford to hold cybersecurity test that probably induces physical damage to testbed infrastructure. Based on above analysis, we propose the functional requirements for virtualization testbed, including

- Scalability, the testbed infrastructure could scale its capability and the physical resources could extend on demands;
- Configurability, the testbed and its resource could be reconfigured according to the specific task and user's demand while keeping the physical infrastructure static;
- Security, the simultaneous experiments could not affect each other; the damage caused by disruptive test could be separated from the testbed infrastructure; the contaminated resource in the experiments could be refreshed and reused in next experiments;
- Programmability, the virtualization testbed resource could be reallocated and redeployed to build a new test object system; the physical production network could be logically reprogrammed to form a new topology;
- Efficiency, the reproduction of object system and its running environment could be executed time efficiently with the help of a suit of automated testbed tools.

To meet all above functional requirements, we propose a virtualization testbed framework, in which the management of testbed running and resource allocation are unified by the virtualization platform. The virtualization framework also brings the merit of enabling disruptive cyber attack tests.

3 Virtualization Testbed Functional Requirements

3.1 The Virtualization Framework Levels

The framework of the virtualization testbed was illustrated in Fig. 1. It could be divided into three levels including testbed infrastructure, virtualization testbed service, and reproductive testing environment.

The reproductive testing environment is the mapping of a real networked information system consistent with test design. The nature attributes and lifecycle of a test would be described with a set of formal syntax first and the recurrence of real scene would hold in this level. The virtualization testbed service level is the test service provider, which unified the management of LVC test resources and was responsible for reallocate, deploy, configure test resources to reproduce an object system. It has separated the testbed infrastructure from reproductive testing environment, which would lead to disruptive effect in a cybersecurity test probably.

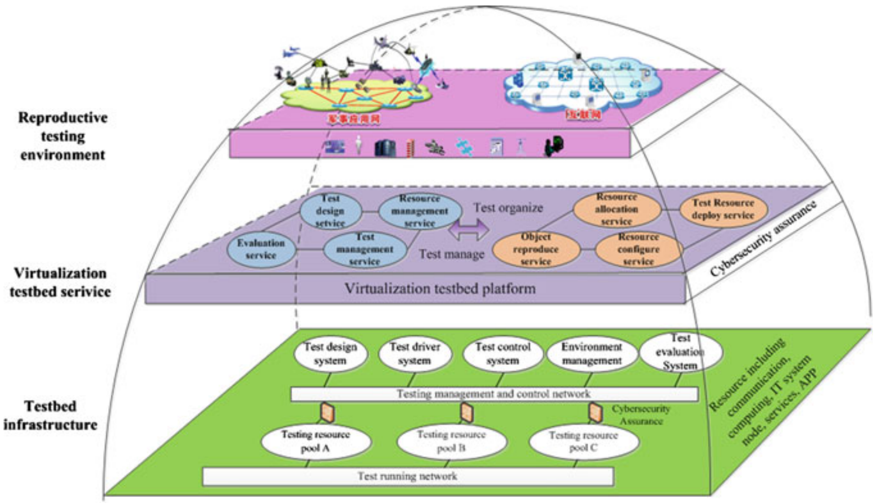
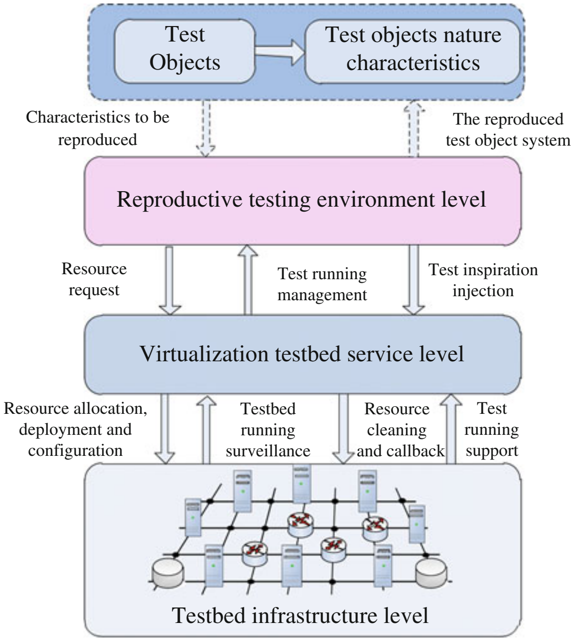


Fig. 1 The virtualization testbed framework

The data also is collected by the testbed virtualization platform efficiently. Testbed infrastructure is the real test capability supplier, which remains under cover the heterogeneous test resources and provides a unified access to them.

The running view of virtualization testbed is illustrated in Fig. 2.

Fig. 2 Testbed running view


Test Objects
Test objects nature characteristics
Characteristics to be reproduced
The reproduced test object system
Reproductive testing environment level
Resource request
Test running management
Test inspiration injection
Virtualization testbed service level
Resource allocation, deployment and configuration
Testbed running surveillance
Resource cleaning and callback
Test running support
Testbed infrastructure level

Step1. The user abstracts the intrinsic characteristics of test object and forms the configuration scripts. The nature characteristics are the most important features and models of the object system to be tested, including system structure, function components, running behavior, and configuration attributes;

Step2. The user puts out the requests of reproducing object system and allocating related resource to virtualization testbed service level, which would allocate resources and organize the test procedure. The virtualization testbed service level would reproduce the object system according to the configuration scripts that described the object attributes and deployment structure;

Step3. After the reproduction of object system had finished, the testing environment for specific task level would load emulation implies to drive the object system running. The resources in the testbed infrastructure level would have been logically occupied by this test and burden the actual computing;

Step4. The virtualization testbed service level records all the test data and events, and it makes preparation for evaluation;

Step5. The virtualization testbed service level organizes the cleaning and call-backing the test resource allocated in the test.

3.2 The Implementation of Virtualization Testbed Platform

The core function of virtualization testbed platform is the implementation of allocation, deployment, configuration and management all kinds of resources. The difficulties lie in three aspects, including the creation of basic test resources, the cybersecurity assurance of virtualization platform, the unified deployment and configuration of LVC resources.

The creation of basic test resources has great difficulty because of the diversity of networked information system components. Each characteristic to be reproduced is quiet different, which leads to the need of applying different way to emulate each

Table 1 The test resource emulation methods

Resource type		Emulation methods
Network resource	Router	Linux container-based router emulation
	Link	Link emulation based on traffic shaping
	Traffic	Traffic emulation based on logical network behaivor mapping
Computing resource	Computer device	Server virtualization method
	Operation system	Platform virtualization method
Service resource	System service	Interface encapsulation and message agent method
Application resource	Sensor	Radar and ESM system emulation
	Weapon	Aeroplane movement emulation
	C2 System	Real C2 system image

basic test resource. All kinds of methods applied in our test resource emulation are illustrated in Table 1.

Among these, the Linux container based router emulation would reproduce large scale network with a great number of emulated routers based on Qugga [6]. The server virtualization is implemented through VMware ESX software and OpenVZ. All above test resource emulation methods are not proposed by us originally and we cite them here to describe our implementation in detail.

The cybersecurity assurance of virtualization platform is an important merit of our framework. This is implemented by three kinds of logical seperations form testbed infrastructure and reproductive test environment, including testbed infrastructure, testbed services, and test data, as illustrated in Fig. 3.

(1) Testbed infrastructure security. In a destructive experiment, the malwares from the object system would probably penetrate residence and attack testbed infrastructure. In order to make sure of the cybersecurity, we have deployed firewall to avoid unlicensed access and misuse. After each test, we have cleaned and polymerized the resource to avoid the negative effect to next test.

(2) Testbed virtualization platform security. When carrying a test on virtualization testbed platform, we have assured the security by validating user's authority through certificate and key. We have also supervised the testing procedure to avoid exceptional behavior that would induce damage.

(3) Test data security. We proposed an agent based test data collection method to avoid the malware promulgated stealthily and to validate the correctness of them.

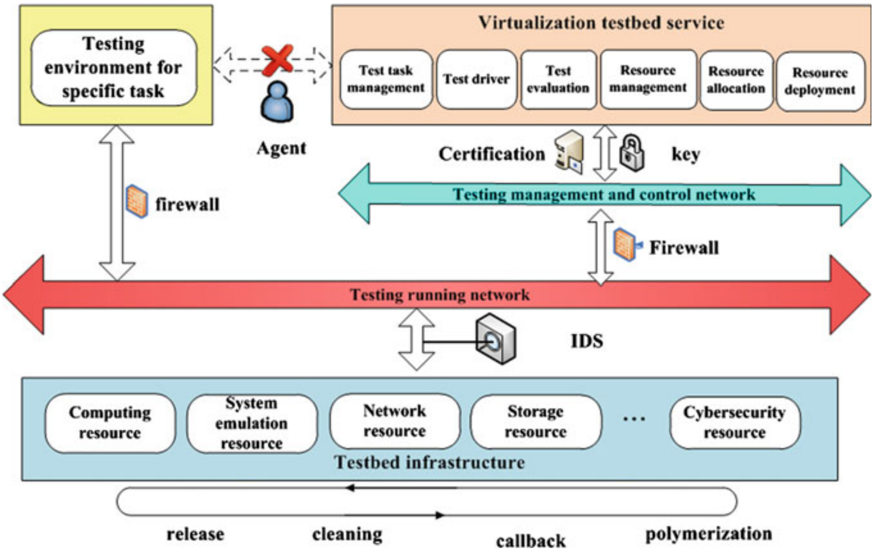


Fig. 3 The cybersecurity design of testbed

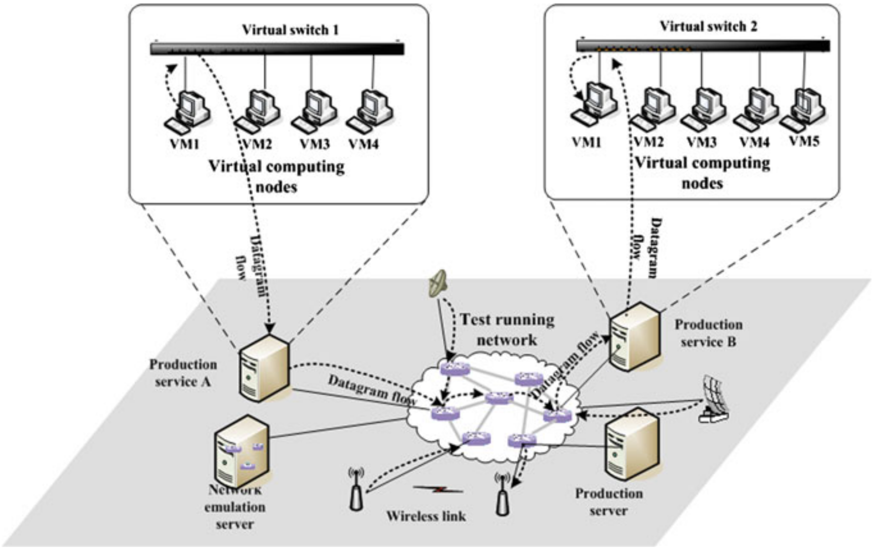


Fig. 4 Test resource virtualization management model

The unified deployment and configuration of LVC resources is another important merit of our testbed platform. It is implemented through port mapping and router redirection, as illustrated in Fig. 4.

(1) The port mapping was applied to interconnect virtual and real resources. The promiscuous mode was set on the virtual machine network interface first, and then all outstreams from the virtual machine was mapped onto the host network interface.

(2) The router redirection was applied in interoperating emulation and real resources. All emulation result flow would be redirected into real network and make the interoperation possible.

(3) The interoperation between virtual and emulation resources could be interacted directly because both kinds of resource are running on the same infrastructure level.

4 Experiments

With the virtualization testbed platform above, we have built a virtualization testbed prototype with ten servers (IBM M3 series) and one high performance router. On this testbed, we have reproduced a networked information system with 218 nodes (including sensors, routers, computers, and C3I nodes) and have validated the five merits of our testbed.

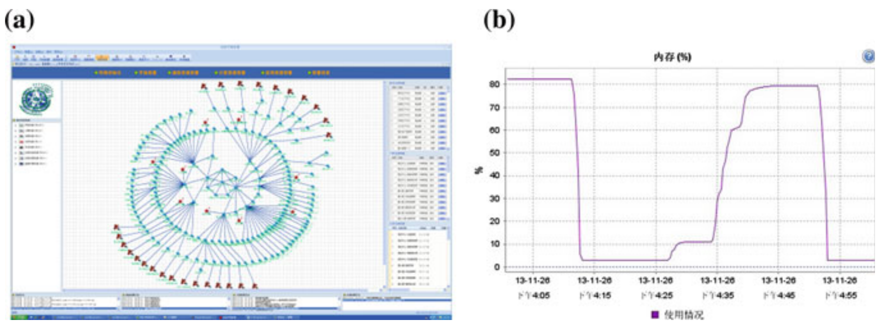


Fig. 5 Virtualization testbed running information **a** Object information system topology **b** Testbed resource changing profile

Table 2 Time on object system reproduction

Reproduction content	Results
Topology reproduction time	2 s
Network reproduction	2 min and 30 s
Computing device reproduction	12 min and 30 s
Applications deployment time	7 min
Test driver loading time	less than 10 s
Test resource callback time	6 min and 50 s
Total time	less than 29 min and 2 s

In Fig. 5a, we showed a networked information system topology that was visually configured by user. All available test resources are listed on the left side of this software and could be used to define a object system. Figure 5b showed testbed resource changing profile in a test. The left part of the profile indicted by dashed line is denoting the callback of memory resource and the right part showed the reallocation process of memory resource in the next test. This profile showed us the programmability of testbed.

The time cost in the reproduction of networked information system with 218 nodes is listed in Table 2, which is less than 30 min and quiet efficient.

5 Conclusion

This paper has analyzed the requirements of networked information system testing and proposed a new virtualization testbed framework, which has the merits of scalability, configurability, security, programmability, and efficiency. The implementation methods for virtualization testbed framework are described and experimental results show the advantage of virtualization testbed framework. Next, we will get further research in networked information system driving methods.

References

1. Yushi L, Heng W, Gangning Z et al (2013) Network-centric architecture for C4ISR system. Command Inf Syst Technol 4(6):1–7
2. Liang J, Lin Z, Ma Y (2013) Future internet experiment platform. Chin J Comput 36 (7):1364–1373
3. Xiaofeng H, Zhiqiang L (2011) The development of national critical infrastructure modeling and simulation. J Univ Shanghai Sci Technol 33(6):687–693
4. Szegedi P (2008) DJRA2.1: architectures for virtual infrastructure, new Internet paradigms and business models [EB/OL], In: FEDERicA project, European Union 7th Framework, Oct 2008. http://www.fp7-federica.eu/documents/FEDERICA-dJRA2.1.pdf
5. Wang J, Li Z, Lu G, et al (2008) Gragon-Lab: next generation Internet Technology Testbed. Sci China Ser E:Inf Sci 38(10):1783–1794
6. Carbone M, Rizzo L (2010) Dummynet revised. ACM SIGCOMM Comput Commun Rev 40 (2):12–20