Paper

IPv6 in Virtualization Environments

Krzysztof Chudzik and Jan Kwiatkowski

Institute of Informatics, Wrocław University of Technology, Wrocław, Poland

Abstract—The primary network layer protocol on which the operation of most computer networks is based, including the Internet is the Internet protocol version 4 (IPv4). Due to the limitations of this protocol, it is becoming increasingly widespread use of the Internet protocol version 6 (IPv6). The IPv6 implements some new features not available in IPv4. The paper provides a short overview of the key features of IPv6 and discussed the possible levels of network virtualization. The research environment to testing the level of support for IPv6 protocol by virtualization environments is proposed. The results of tests conducted using the proposed research environment for Hyper-V virtualizer are presented.

Keywords*—*Hyper-V virtual machine, IPv6 protocol, virtualization environment.

1. Introduction

The primary network layer protocol on which the operation of most computer networks is based, including the Internet is the Internet protocol version 4 (IPv4). Due to the limitations of this protocol, it is becoming increasingly widespread use of the Internet protocol version 6 (IPv6) [1], [2], [3]. The IPv6 implements some new features not available in IPv4. The interoperability features with IPv4 is not implement in IPv6, then IPv6 creates essentially a independent (parallel) network to IPv4. The traffic between both networks requires special translators, however most of existing operating systems supports both protocols.

Implementation of IPv6 cannot ignore the virtual environments, which the main tasks are abstraction and isolation of widely understood network resources, including computer networks, network devices, computers, operating systems, applications, services, etc.

In the paper the research environment to testing the level of support for IPv6 protocol by virtualization environments is proposed. The results of tests conducted using the proposed research environment for Hyper-V virtualizer are presented.

The paper is organized as follow. Section 2 provides a short overview of the key features of IPv6. Section 3 describes the coexistence of IPv6 and IPv4. The main topic of Section 4 are the levels of network virtualization. In Section 5 interface for IPv6 application using virtual environment is proposed. In Section 6 the research environment for testing the level of support for IPv6 protocol by virtualization environment is proposed. Section 7 presents results of testing the Hyper-V virtualizer. Finally, Section 8 outlines the work and discusses the further works.

2. Key Features of the IPv6

The first documents relating to IPv6 were published in the 90s: RFC 1883 [4] document was issued in 1995 and shortly afterwards obsoleted by RFC 2460 [5] in 1998. The latter is constantly updated. According to information given at the Internet Engineering Task Force (IETF) website[1] the last update of this document occurred in May 2010 (the state of the art at the time of writing in June 2010). As described in [5], IPv6 is a new version of the Internet protocol, designed as the successor to IPv4 [6]. The most significant changes and improvements introduced by IPv6 are as follows [5]:

Expanded addressing capabilities. The IP address size increases from 32 bits to 128 bits. This is the solution of the most pressing problem, namely the shortage and depletion of the pool of free IPv4 address. The resulting new larger pool of addresses can support a much greater number of networks and nodes. By that means, the new protocol permits to avoid temporary solutions as network address translation (NAT), which causes problems because a number of network devices uses and is represented in the Internet by the same public address and there is no way to distinguish between them from the Internet side. Larger address space allows to create new mechanisms to facilitate the configuration of network devices, such as stateless auto-configuration of addresses with the use of the MAC addresses and the IPv6 prefixes advertised by routers. IPv6 allows more levels of addressing hierarchy, which simplifies scalability and routing, including the routing of multicast traffic which replaces the broadcast traffic in IPv4. Some new types of addresses are defined, such as an anycast addresses.

Header format simplification. Some IPv4 header fields have been dropped or made optional to reduce the cost of packet processing and usage of bandwidth [5]. The IPv6 header has fixed length and is optimized for processing up to 64 bits at a time (32 in IPv4). Routers do not calculate any IPv6 header checksum as do that in IPv4. Routers also are not responsible for fragmentation of oversized packets. They only signal the source to send smaller packets [3].

[1] http://www.ietf.org

Improved support for extensions and options. The RFCs [4], [5] assume that changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

Flow labelling capability. IPv6 allows labelling of packets. Packages may be labelled as belonging to a certain type of network traffic, which requires special handling by quality of services (QoS), for example the traffic associated with VoIP services.

Authentication and privacy capabilities. In the RFCs [4], [5], it was stated that extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6. This statement is sometimes a source of serious misunderstandings, because only support is mandatory, not the use for example IPSec. IPv6, in itself, is not more secure than IPv4, but the process of implementing security mechanisms is easier, because there is no address translation (NAT), and some problem may be omitted. In addition, the vast IPv6 address space is not densely populated by nodes, which makes it difficult to scan a network looking for potential victims of the attack [3].

3. IPv6 and IPv4 Coexistence

Because of significant differences between protocols, the transition from IPv4 to IPv6 can not be done in one step, but must be carried out the migration process. The lack of IPv6 backward compatibility causes the two protocols IPv6 and IPv4 must be used simultaneously, often on the same node. Certain services are available to the node over IPv6, while others over IPv4. This requires a dual stack of IP protocols and in fact is done on nodes that have IPv6 support, for example the operating systems which support both protocols are configured with both types of addressees and are equipped with utility programs that support both protocols. Usually there is configured a kind of priority of IPv6 over IPv4 and IPv4 are used where access to a service or node are not possible by the use of IPv6.

The above resolves the problems of communication in environments where IPv6 is implemented and the routing of IPv6 is supported. In the most cases, in the migration process, communication problems associated with incomplete or no implementation of the IPv6 routing occurs. Since IPv4 and IPv4 routing is usually a fully implemented, the IPv6 network traffic can be tunnelled through existing IPv4 networks. The tunnels use encapsulation of IPv6 in IPv4 packet and usually add the IPv4 header before the IPv6 packet at the point of entry to the tunnel. In this form, the packet is transmitted by the networks that support only IPv4. At the other end of tunnel, the unnecessary IPv4 header is removed and the original packet is transmitted to its IPv6 destination (Fig. 1).

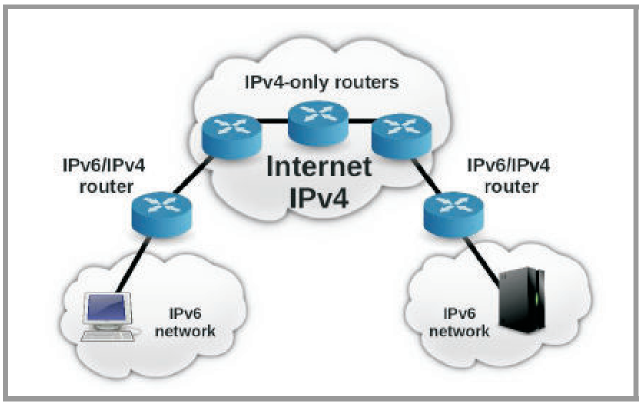


Fig. 1. The IPv6 communication over IPv4 Internet.

One can configure tunnels manually or use automatic configuration. Currently at least five different automation methods are defined [3]:

Transition mechanisms for IPv6 hosts and routers (RFC 2893) [7]. The IPv6 addresses of nodes consist of two parts: the first is a series 96 zero bits (denoted as the prefix ::/96), and the second is the 32-bit IPv4 address, so the transformation between the corresponding IPv4 and IPv6 addresses is trivial. Although not formally deprecated by an IETF standards action, automatic tunnelling should be considered obsolete [3].

6over4: Transmission of IPv6 over IPv4 domains without explicit tunnels (RFC 2529) [8]. 6over4 is used within a single organization or site network. 6over4 treats an IPv4 network as a IPv6 subnet, which delivers basic services for IPv6 hosts, including IPv6 address autoconfiguration and support for multicast. Because of this requirement, the discussed method of tunneling is not often implemented, due the lack of support for multicast traffic in contemporary IPv4 networks [3].

ISATAP: Intra-site automatic tunnel addressing protocol (RFC 5214) [9]. ISATAB, similarly to 6over4, is used within a single organization or site network and that network is treated as non-broadcast multiple access (NBMA). The mechanism is designed for dual-stack nodes to connect them via IPv4-only networks. The IPv6 addresses are constructed in modified EUI-64 format with the use of IPv4 addresses and the universal/local and individual/group bits, which allow take decisions about routing to destination and tunneling [3], [9].

Teredo: Tunneling IPv6 over UDP through NATs (RFC4380) [10]. In RFC 4380 a service is proposed that enables nodes located behind one or more IPv4 network address translations (NATs) to obtain IPv6 connectivity by tunneling packets over UDP. Teredo service requires to operate so-called "Teredo servers" and "Teredo relays". The teredo servers manage a small fraction of the traffic between teredo clients, while the teredo relays act as IPv6 routers between the teredo service and the "native" IPv6 Internet [11].

6to4: Connection of IPv6 domains via IPv4 clouds (RFC 3056) [11]: This is one of the more popular methods, and is implemented in most modern operating systems [3]. This method provides a link between isolated areas of the operation of IPv6 trough the areas of the operation of IPv4.

Each node with a routable IPv4 address can create a special 48-bit long prefix of IPV6 address to communicate which consist of 2002(hex) followed by IPv4 address in the hexadecimal form. Due to the construction of an IPv6 address mutual conversion between IPv4 and IPv6 is trivial. When node sends a packet, that packet is encapsulated in the IPv4 packet and the destination address is taken from IPv6 address. The communication between IPv6 areas is supported by dedicated relays/gateways [3], [11].

4. Introduction to Virtualization Environments

There are many benefits related to using virtualization, the most important are as follows:

- Reduction of hardware maintenance cost due to using the lower number of physical servers.
- Preventing application from impacting another application when upgrades or changes are made.
- By developing a standard virtual server, it is possible in easy way make its duplication which speed up server deployment.
- Better utilization of available resources.
- Deploying multiple operating system technologies on a single hardware platform.

Additionally virtualization gives opportunity of using the concept of parallel Internets as an innovative way to enable end-to-end service differentiation at the IP level in terms of not only traditional QoS such as delay and loss, but also resilience and availability. Specifically, parallel Internets are coexisting parallel networks composed of interconnected per-domain planes. Network planes are setup to transport traffic flows from services with common connectivity requirements. The traffic delivered within each Network plane has particular treatment in both forwarding and routing [12].

Virtualization, as a mechanism of abstraction and isolation of network resources with support for IPv6, can be implemented at different levels of the network environments; communication and operating system levels respectively.

4.1. Virtualization at the Level of Communication Devices

At the communication device virtualization level, communication and address spaces can be created. The division may occur at layer 2 of ISO/OSI model related to the physical addressing and switching or at layer 3 associated with the logical addressing and routing. Within layer 2, for example, virtual local area networks (VLANs) can be created and IPv6 can be used by the network administrator to have access to switching hardware (Fig. 2).

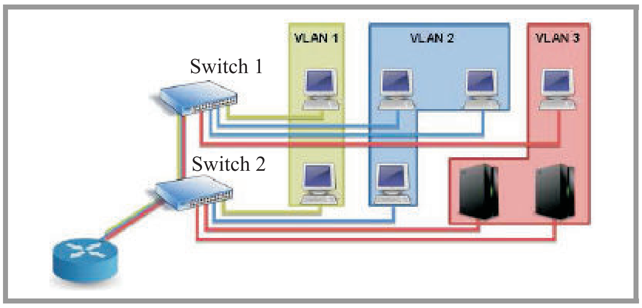


Fig. 2. An example of VLAN virtualization at layer 2.

However, at layer 3, one can create, for example, virtual private networks (VPNs) and IPv6 can be both a transport (tunneling) protocol and transported (tunneled) protocol (Fig. 3).

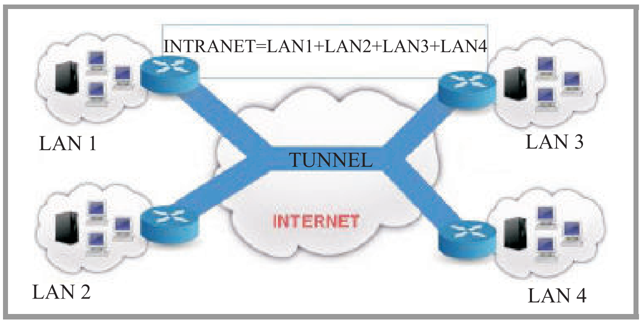


Fig. 3. An example of VPN tunnels virtualization at layer 3.

At the level of communication devices, communication hardware virtualization is possible. For example, multiple routers and/or switches infrastructure, which are interconnected by virtual networks, can be created within a single router or switch. Handling and support for IPv6, using virtualization at the level of communication, are implemented by the functionality of hardware and operating systems (an operating system at this level can operate, for example, as a software firewall and access router to support VPN tunnels).

4.2. Virtualization of Operating Systems, Applications and Services

At the virtualization of operating system level, hardware resources are shared among multiple virtual executive environments. These are the types of virtualization.

Full virtualization. In the base system (host), the total abstraction and emulation of existing or fictitious hardware is carried out. In this abstract emulated environment, an operating system (guest) is executed. Virtualized system operates as if it were running directly on physical hardware.

Paravirtualization. At the base system, hardware abstraction is carried out, but the virtual hardware presented to virtualized system is similar (not necessarily identical) to the real hardware. In addition, paravirtualization will not work with any operating system, but the operating system has to be adapted to this type of virtualization.

Container virtualization. Container virtualization systems provide the option to run multiple applications in isolated environments, with adequate security on a single operating system. The mechanism of operation is based on the creation of many user spaces that are properly isolated.

Service virtualization. Similar in its operation to the container virtualization. The main difference is that the virtual machine is created when the service is requested and then hardware resources and the environment in which the machine can run are localized.

In the case of virtualization at the level of operating systems, applications and services, handling and support for IPv6 can be implemented by virtualized operating systems, applications and services as well as the virtualization environment itself.

5. Interfaces for IPv6 Applications

At the application level different types of applications with different functional as well as non-functional requirements can be used. Moreover, different system and user interfaces can be used depending on application requirements. It causes the necessity of developing the general, flexible interface that can be used by all available applications. Assuming that the global system is built as a federation of independent execution systems connected by the computer network, it causes that the execution systems should hide their internal complexity by offering a common interface to their internal resources. Additionally, each of the execution systems works in the autonomous manner, ensuring efficient local resources utilization. To fulfil above requirements the system that consist of two cooperating modules: the execution virtualization module and resource allocation module is proposed. The proposed system offers a service abstraction on the highest level with efficient resources utilization performed inside each of execution systems [13].

The execution virtualization module implements an application (service) execution interface, used to hide the underlying hardware-specific details. The virtualization makes it possible to utilize a variety of hardware resources dynamically selected to ensure efficient service execution performed in accordance to the requirements. It aggregates, interprets and utilizes the monitoring data to select the service execution location and execution details. The execution virtualization module virtualizes the necessary resources for the realization of services.

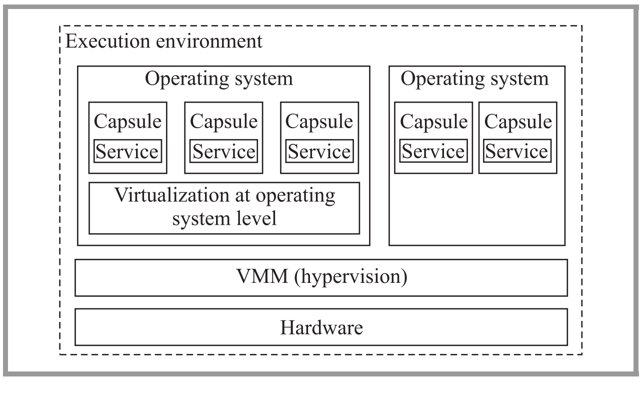


Fig. 4. Two-level virtualization system.

Figure 4 presents two-level virtualization system used by the execution virtualization module. The structure of the system is as follows, over the VMM (hypervisor) level different so called hard virtualizators (full virtualization or paravirtualization) are used. The first level gives opportunity of using different operating system at the same hardware that can provide different services offered by installed application. In the second virtualization level the container virtualizators are used and it enables to have a copy of the same service that can be used by different users.

The resource allocation module is responsible for choosing the most suitable execution system for a service request. The execution system running the service is chosen basing on the information about the available computational resources, virtualizers, operating systems and services.

The benefits of using the proposed system are as follows:

- Reduce system response time by accelerating the time of the service execution through an appropriate resources allocation.
- Reduce the number of running services. It reduces the cost of management.
- Hiding the specific implementation. It gives the opportunity to build distributed systems datacentres adjusted dynamically to the requirements.
- The advantages of virtualization are applied, too.

6. Testbed for Testing IPv6 Support by Virtualization Environments

The verification of handling and active support for IPv6 is required due to the large number of virtualization environments, virtualized systems and their different operation. For given a virtualization platform (the host system) and virtualized systems (the guest systems), which create a virtualization environment, a research virtualization environment is proposed and presented in Fig. 5.

The proposed testing environment consists of three virtual networks and one bridging network between the virtual environment and the host system. The environment allows to test routing and routing protocols to determine the impact of host and the opportunity to interact with external devices. It is the minimal configuration, in which there are no contiguous network. This allows to test the routing protocols, both internal and external manner. If in a small network routing protocol is working correctly, it means that its messages are exchanged correctly and has no negative impacts of the host.

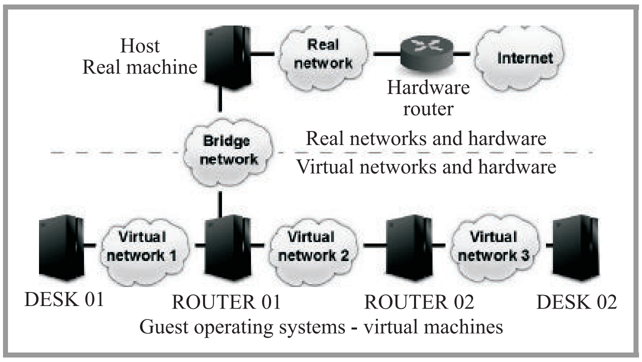


Fig. 5. Example of an IPv6 research environment.

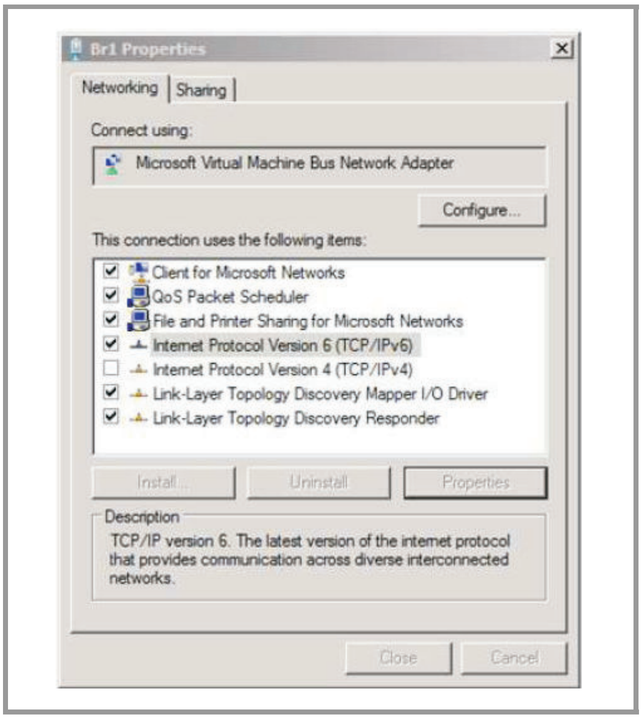
The scope of research conducted with the use of this environment may include:

- Support offered by the virtualization environment to operating systems working on the basis of IPv6. Useful environment should support the main operating systems in the server market, that is, Unix / Linux and MS Windows.
- Creation and configuration of virtual networks with IPv6 as the main protocol at the network layer. Ability of creation networks with different levels of isolation should be taken into account:
  - the networks that are completely isolated. This kind of network should support communication between guest operating systems;
  - the networks for connection between a guest operating system and the host (bridge networks);
  - the networks that are attached directly to the network infrastructure to which the host computer is connected to. This kind of network should provide direct communication between the guest operating systems and real external network environment.
- Support for IPv6 routing and routing protocols, in particular, whether the flow of packets in virtual networks is the same as in real networks, and whether the host system does not interfere with the flow.
- Active support for IPv6 network services, for example, determining whether one is able to start the DHCP server on the side of the host system, which will provide the configuration parameters of the IPv6 network interfaces at the guest systems. Integration of virtual IPv6 networks with real IPv6 networks. The problem involves finding whether there is a possibility of packets routing from internal virtual networks outside the host system and physical access to the host system physical network cards for guest operating system so they can directly use physical networks cards to communication with real networks.
- Support the integration of IPv6 with IPv4 in tunnelling and the direct communications.

7. Testing Hyper-V Virtualizer

The verification of handling and active support for IPv6 in the Hyper-V virtualization environment is required due to the large number of possible applications of this environment. Presented in this section results of the Hyper-V virtualizer testing has been obtained using the research environment presented in the previous section. The proposed by us testing method covers the basic functionality of the Hyper-V environment and simultaneously the MS Windows server 2008R2 operating system (the latest available version), within which we ran Hyper-V as a host operating system. The study used a virtual machines with the same operating system as the guest operating system. The aim of the study is to examine how the Hyper-V environment and the MS Windows server 2008 R2 operating system operate as the host and guest systems using IPv6 as the only communication protocol. The study covered the basic functionalities and services of the IPv6 protocol. Hence the basic idea of the research is related to automatic and static addressing. The examination includes the ability to creating and configuring virtual networks also.

Hyper-V was installed and started in the Microsoft Windows 2008 R2 enterprise edition operation system. The process of installation is easy and limited to installation of a role of Hyper-V with usage of a role installation wizard only. The fully functional virtualization environment is available immediately after the installation. The virtual networks and the virtual machines were created and configured in this environment as shown in Fig. 5. There were no problems with the creation of virtual networks or virtual machines. The host computer and the virtual machines



Connect using:
Microsoft Virtual Machine Bus Network Adapter

This connection uses the following items:

☑ Client for Microsoft Networks
☑ QoS Packet Scheduler
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol Version 6 (TCP/IPv6)
☐ Internet Protocol Version 4 (TCP/IPv4)
☑ Link-Layer Topology Discovery Mapper I/O Driver
☑ Link-Layer Topology Discovery Responder

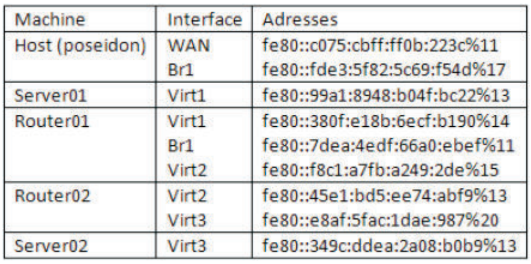
Description
TCP/IP version 6. The latest version of the internet protocol that provides communication across diverse interconnected networks.

Fig. 6. The static IPv6 configuration.

were configured only with IPv6 addresses to avoid IPv4 issues, as shown in Fig. 6.
Additionally, the firewalls were disabled on the host and the virtual machines to avoid the impact of firewalls on the test results.

7.1. The Link-Local Addressing and Connection Test

The purpose of the test is to check the functioning of the "link-local" addresses in the Hyper-V environment on the host and the virtual machines. These addresses are assigned automatically, so the first step is to inventory the assigned addresses. The inventory results are shown in Fig. 7. Obviously, addresses are unique to this particular environment and test.



Machine	Interface	Adresses
Host (poseidon)	WAN Br1	fe80::c075:cbff:ff0b:223c%11 fe80::fde3:5f82:5c69:f54d%17
Server01	Virt1	fe80::99a1:8948:b04f:bc22%13
Router01	Virt1 Br1 Virt2	fe80::380f:e18b:6ecf:b190%14 fe80::7dea:4edf:66a0:ebef%11 fe80::f8c1:a7fb:a249:2de%15
Router02	Virt2 Virt3	fe80::45e1:bd5:ee74:abf9%13 fe80::e8af:5fac:1dae:987%20
Server02	Virt3	fe80::349c:ddea:2a08:b0b9%13

Fig. 7. The local-link addresses assigned to network interfaces during test.

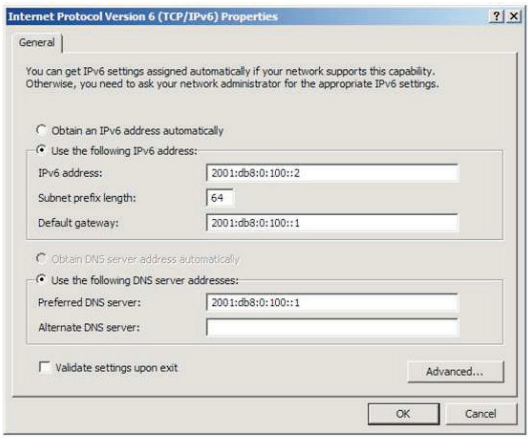
After the inventory, the connection between all the interfaces working in the same virtual networks have been tested using the ping utility. There were no communication issues. Furthermore, it was found that in contrast to Linux, the machines with multiple network cards do not need to enter the identifier of the interface through which communication is to take place. The ping command in Linux is as follows:

- ping6 [-I ⟨device ⟩]⟨link-local-ipv6address⟩,
- ping6 -I eth0 fe80::c075:cbff:ff0b:223c.

The usage of the ping command in Windows remains typical: ping fe80::c075:cbff:ff0b:223c.

7.2. The Static Addressing Test

The purpose of the test is to check the functioning of the manually configured addresses in the Hyper-V environment on the host and the virtual machines. The addresses are configured manually in the dialog windows presented in Fig. 8 according to the address schema presented in Fig. 9.
After the configuration of addresses the tests of connectivity between all the interfaces working in the same virtual networks have been carried out using the ping utility. There were no communication issues.



Internet Protocol Version 6 (TCP/IPv6) Properties

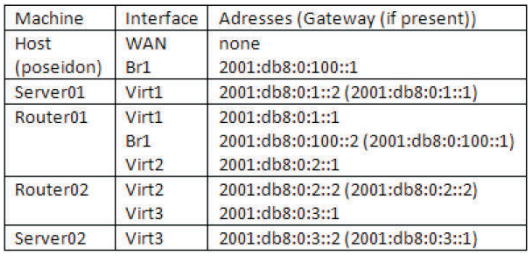
You can get IPv6 settings assigned automatically if your network supports this capability.
Otherwise, you need to ask your network administrator for the appropriate IPv6 settings.

○ Obtain an IPv6 address automatically
● Use the following IPv6 address:
IPv6 address: 2001:db8:0:100::2
Subnet prefix length: 64
Default gateway: 2001:db8:0:100::1

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 2001:db8:0:100::1
Alternate DNS server:

□ Validate settings upon exit

Fig. 8. IPv6 address configuration dialog window.



Machine	Interface	Adresses (Gateway (if present))
Host (poseidon)	WAN Br1	none 2001:db8:0:100::1
Server01	Virt1	2001:db8:0:1::2 (2001:db8:0:1::1)
Router01	Virt1 Br1 Virt2	2001:db8:0:1::1 2001:db8:0:100::2 (2001:db8:0:100::1) 2001:db8:0:2::1
Router02	Virt2 Virt3	2001:db8:0:2::2 (2001:db8:0:2::2) 2001:db8:0:3::1
Server02	Virt3	2001:db8:0:3::2 (2001:db8:0:3::1)

Fig. 9. The static IPv6 address.

7.3. The Static Routing Test

The purpose of the test is to check the functioning of static routing in the Hyper-V environment. The network interface addresses were configured according to Fig. 9 and routers were configured as shown in Fig. 10.

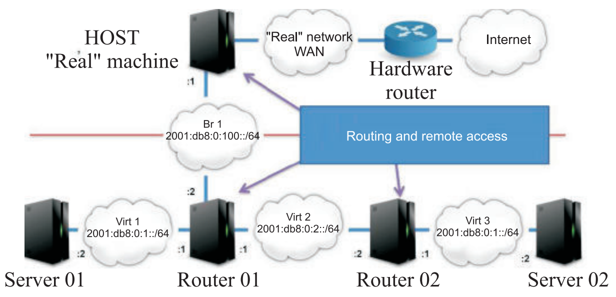


Fig. 10. The router placement in the test environment.

The connectivity test were run between the host computer and the virtual machine and between the virtual machines as shown in Fig. 11. The conclusion is that the static routing works correctly and there are no connection issues. Since the routing protocols are not implemented in MS Windows server 2008 R2, the tests of routing protocols have not been carried out.

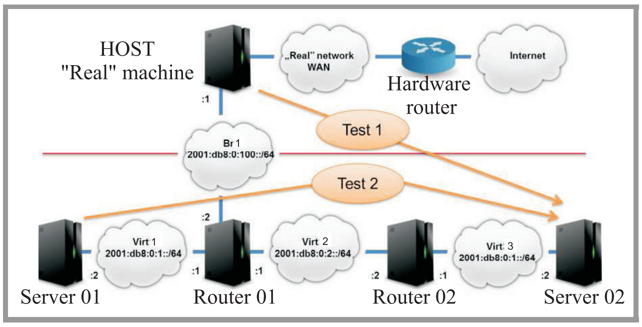


Fig. 11. The test of the static routing.

7.4. The Automatic Address Configuration Tests

The purpose of the tests is to check the functioning of the automatic configuration of the network interface addresses. Three methods of configurations were tested and results are presented: the stateless addressing, the stateful addressing and the configuration through the relay agent. The tests check the readiness of the virtual environment and the virtual machines based on MS Windows server 2008 R2 to the migration processes. If the automatic address configuration is carried out correctly, there is no need to configure manually the network interfaces on the new host machine after the migration process. The configuration of the test environment is presented in Fig. 12.

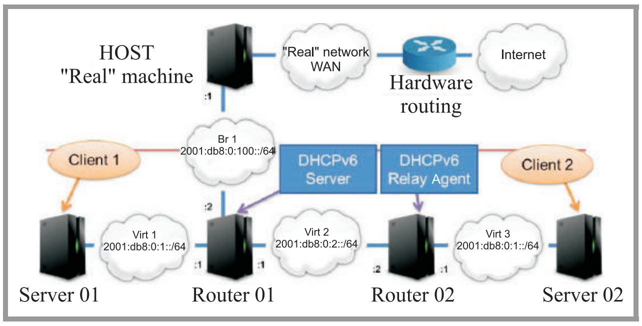
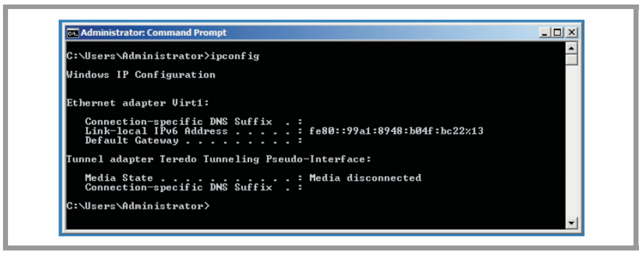


Fig. 12. The test environment for the automatic address configuration.

7.5. The Stateless Addressing Test

The stateless address configuration is very convenient if any sophisticated parameters have not to be configured on the network interface and a quick and easy method of configuration, without administrator intervention, is demanded. A computer which interface is to be configured is called the client. The stateless configuration is the process in which the client interface is configured to obtain the IPv6 address automatically and the configuration process is based on the exchange of messages between the client computer and the router in the network the interface is connected to. There is no DHCP server. Server 01 as the client (Client 1 in Fig. 12) and Router 01 as the router to exchange messages were configured in the test. It was expected, that client would be configured with the unique IPv6 address and the default gateway. The DNS server addresses are configured automatically as fec0:0:0:ffff::1%1, fec0:0:0:ffff::2%1, fec0:0:0:ffff::3%1. The initial configuration of the client interfaces is shown in Fig. 13.



```
C:\Users\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Virt1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::99a1:8948:b04f:bc22%13
   Default Gateway . . . . . . . . . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\Administrator>
```

Fig. 13. The initial configuration of the client interfaces.

The interface Virt 1 was configured. After configuration process, the client was correctly configured and the check of connectivity was carried out. The default gateway was configured as the link-local address. There were no issues with the stateless addressing.

7.6. The Stateful Addressing Test

The stateful configuration can be used for more sophisticated configuration of the network interface, where the stateless configuration is insufficient and/or we want to reserve addresses. The DHCP service is used to configure clients. In the test environment Router 01 was configured as the DHCP server and Server 01 acted as the client (Fig. 12). It was expected, that the client would be configured by the DHCP server with the IPv6 address, the address of the DNS servers and the DNS domain the machine belongs to. The default gateway would be configured by the router. Both cases of address selection were tested: any address from the scope and the reserved.
The DHCP server was configured with the scope with the following parameters:

– the network prefix: 2001:db8:0:1::/64,

– the excluded addresses 2001:db8:0:1:: to 2001:db8:0:1::ffff,

– the DNS server address: 2001:db8:0:100::1.

Since the DHCP server is the router, the advertisements of the default rout ware enabled:

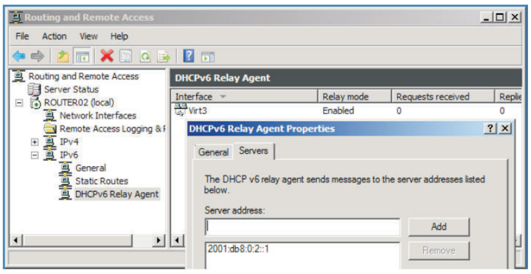
– netsh interface ipv6 set interface Virt 1 advertisedefaultroute=enabled,

– netsh interface ipv6 set interface Virt 1 advertise=enabled The reservations of IPv6 addresses were performed.

During the test, the client was correctly assigned all the demanded parameters. The default gateway was configured as the link-local address. There were no communication issues.

7.7. The DHCP Relay Agent Test

The DHCP relay agent test has the same assumptions as stateful addressing test, but additionally we assume that the DHCPv6 server is not connected to the subnet of the client and the realy agent is demanded between client and the DHCP server. In this test, Server 02 act as the client (Client 2) and Router 02 is configured with the relay agent role (Fig. 12). The relay agent was configured as in Fig. 14.



Interface	Relay mode	Requests received	Replie
Virt3	Enabled	0	0

The DHCP v6 relay agent sends messages to the server addresses listed below.

Server address:

2001:db8:0:2::1

Fig. 14. The relay agent configuration.

Additionally, the options advertise, advertisedefaultroute, managedaddress oraz otherstateful should be enabled for the Virt 3 interface on Router 03. The client was configured correctly.

7.8. The DNS Integration Test

The client computers and virtual machines can register itself in their DNS domain without the intervention of the network administrator. The DNS integration test consists of checking the registration process. This feature facilitates the migration of virtual machines. If that process works correctly, we can assume, that even the IPv6 address changes during the migration process, we can still contact the migrated machine via the fully qualified domain name which remain unchanged and is correctly translated to the current client machine IPv6 address by the DNS server.
The corporate model of the MS Windows implementation assumes operations in Active Directory for the security reason. During the integration test, it was stated that the virtual machines have to join the Active Directory domain for proper registration in DNS domain. There were no problems with the self-registration process in DNS.

8. Conclusions

The results of tests conducted for the Hyper-V virtualizer can be summarized as:

- The Hyper-V virtualization environment and MS Windows 2008R2 server (tested on enterprise edition) in the role of the host and the virtual machines, working exclusively with IPv6, behave properly with addresses assigned statically and dynamically (including stateless and stateful configuration). There is no issues with manually configured routing.
- There are no communication problems at the border of the host computer (host) and virtual machines (guests).
- MS Windows 2008R2 server used in the proposed testing environment does not support dynamic routing, therefore dynamic routing has not been tested.

As a second step of presented research the experiments of Hyper-V virtualizer will be conducted using experimental network that consists of two physical servers Server1 and Server 2 combining different virtual entities representing intranet connections and a physical Switch 1 to communicate between the two physical servers. All the virtual network traffic is multiplexed over the physical 10/100/1000 Mbit/s Ethernet interfaces of Server 1 and Server 2.

Acknowledgement

The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.01.02-00-045/09-00.

References

[1] Y. Mun and H. K. Lee, *Understanding IPv6*. New York: Springer, 2005.

[2] *An IPv6 Deployment Guide*. M. Dunmore, Ed. The 6NET Consortium, Sept. 2006.

[3] I. van Beijnum, *Running IPv6*. Berkeley: Apress, 2005.

[4] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification". RFC 1883 (proposed standard), obsoleted by RFC 2460, IETF, Dec. 1995.

[5] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification". RFC 2460 (draft standard), updated by RFCs 5095, 5722, 5871, IETF, Dec. 1998.

[6] J. Postel, "Internet protocol". RFC 791 (standard), updated by RFC 1349, IETF, Sept. 1981.

[7] R. Gilligan and E. Nordmark, "Transition mechanisms for IPv6 hosts and routers." RFC 2893 (proposed standard), obsoleted by RFC 4213, IETF, Aug. 2000.

[8] B. Carpenter and C. Jung, "Transmission of IPv6 over IPv4 domains without explicit tunnels". RFC 2529 (proposed standard), IETF, March 1999.

[9] F. Templin, T. Gleeson, and D. Thaler, "Intra-site automatic tunnel addressing protocol (ISATAP)". RFC 5214 (informational), IETF, March 2008.

[10] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)". RFC 4380 (proposed standard), IETF, Feb. 2006.

[11] B. Carpenter and K. Moore, "Connection of IPv6 domains via IPv4 clouds." RFC 3056 (proposed standard), IETF, Feb. 2001.

[12] N. Wang *et al.*, "A framework for lightweight QoS provisioning: Network planes and parallel internets", in *Proc. 10th Int. Symp. on Integrated Network Management*, IEEE, 2007, pp. 797–800.

[13] J. Kwiatkowski, M. Pawlik, M. Fras, M. Konieczny, A. Wasilewski, "Design of SOA-based distribution system", in *SOA Infrastructure Tools – Concepts and Methods*, Poznań University of Economics Press, 2010, pp. 263–288.

Krzysztof Chudzik – for biography, see this issue, p. 12.

Jan Kwiatkowski – for biography, see this issue, p. 13.