module.

[0007] In a possible design, multiple virtual machines may run on the host, each virtual machine is connected to at least one VF of the first network port, and the multiple VFs of the first network port have different virtual local area network VLAN identifiers. A VLAN identifier different from a VLAN identifier of another VF is set for each VF. This ensures that a broadcast packet carrying a VLAN identifier is propagated only on a virtual local area network corresponding to the VLAN identifier and is not sent to another VF with a different VLAN identifier.

[0008] In a possible design, the first virtual machine obtains the destination MAC address of the to-be-sent data packet by using an ARP process. Specifically:

[0009] The first virtual machine sends an ARP request packet by using the VF that is connected to the first virtual machine, where the ARP request packet carries a VLAN identifier of the VF that sends the ARP request packet;

after receiving the ARP request packet, the switching equipment of the first network port broadcasts the ARP request packet; after receiving the ARP request packet broadcast by the switching equipment, the virtual bridge on the VMM forwards the ARP request packet to an SDN controller, where the SDN controller records network topology information of a cloud computing system; and when receiving the ARP request packet, the SDN controller constructs a flow entry, and returns the flow entry to the virtual bridge, where the flow entry includes an ARP pickup setting, the ARP pickup setting is used to instruct the virtual bridge to construct an ARP response packet, and the ARP response packet carries the MAC address that is of the uplink port of the virtual bridge and that is used as the destination MAC address of the to-be-sent data packet; and

the virtual bridge returns the ARP response packet to the first virtual machine according to the flow entry, where the ARP response packet carries the MAC address that is of the uplink port of the virtual bridge and that is used as the destination MAC address of the to-be-sent data packet.

[0010] In this embodiment of the present invention, a new ARP process is defined. The destination MAC address of the to-be-sent data packet is set to the MAC address of the uplink port of the virtual bridge, so that the data packet sent by the first virtual machine is forwarded

[0021] The second processor is configured to execute an instruction in the second memory to implement a function of the switching equipment in the first aspect.

[0022] In a possible design, the host and an SDN controller are implemented by a general-purpose or dedicated server. The server includes a processor, a memory, a system bus, and an input/output interface. The processor is configured to support a corresponding function of the host in the system. The input/output interface is configured to communicate with another component in the cloud computing system. The processor executes the instructions stored in the memories.

[0023] According to a fifth aspect, an embodiment of the present invention provides a computer storage medium that is configured to store a computer software instruction used by the foregoing host. The computer software instruction includes a program designed for executing the foregoing aspects.

[0024] According to a sixth aspect, an embodiment of the present invention provides another computer storage medium that is configured to store a computer software instruction used by the switching equipment in the foregoing network interface card. The computer software instruction includes a program designed for executing the foregoing aspects.

[0025] According to a seventh aspect, an embodiment of the present invention provides a computer program. When a physical server runs the computer program, the physical server performs functions of a virtual machine, a virtual bridge, and a virtual network function module in a host.

[0026] According to an eighth aspect, an embodiment of the present invention provides a computer program. When a processor or a programmable logic circuit in a network interface card runs the computer program, the network interface card performs a function of the switching equipment in the foregoing aspects.

BRIEF DESCRIPTION OF DRAWINGS

[0027] To describe the technical solutions in the embodiments of the present invention or in the prior art more clearly, the following briefly describes the accompanying drawings required for describing the embodiments or the prior art. Apparently, the accompanying

bottom layer of the physical server is a hardware layer. The hardware layer mainly includes hardware resources such as a central processing unit (CPU, Central Processing Unit), a memory, a hard disk, and a network interface card. Server virtualization implements a virtualized running environment of multiple virtual machines (Virtual Machine, VM) on the physical server by using virtualization software (for example, VMWare ESX or Citrix XEN). A software layer that is installed on the server to implement a virtualized environment is referred to as a virtual machine monitor (Virtual Machine Monitor, VMM). The VMM running on the hardware layer is responsible for scheduling, allocating, and managing the hardware resources at the hardware layer. Multiple virtual machines VMs run on the VMM. The VMM provides each virtual machine with a hardware environment such as a virtualized CPU, memory, storage, IO device (such as a network interface card), and Ethernet switch, to ensure that the multiple virtual machines run in a manner of isolating from each other.

[0037] In a virtualized running environment, the VMM creates a virtual network interface card (Virtual Network Interface Card, vNIC) for each virtual machine, and a virtual switch vSwitch provides a capability of communication between the virtual machines and between the virtual machines and an external network. The virtual network interface card of each virtual machine is corresponding to a logical port of the vSwitch. A physical network interface card of the host is corresponding to a port connecting the vSwitch to an external physical switch. When a data packet sent or received by a virtual machine passes through the VMM, a virtual network function module on the VMM performs network function processing on the data packet that passes through the VMM. The virtual network function module is a software module and can be updated as required. Therefore, the virtual network function module on the VMM can provide abundant network functions for a user.

[0038] A network port of the physical network interface card supports a virtualization capability. This can be implemented by means of single-root input/output virtualization (Single Root Input/Output Virtualization, SR-IOV) or multi-root input/output virtualization (Multi Root Input/Output Virtualization, MR-IOV). In this embodiment of the present invention, an SR-IOV technology is used as an example for description. The SR-IOV technology is a hardware-based virtualization solution. A PCIe (Peripheral Component Interconnect Express, peripheral component interconnect express) device can be efficiently

example to describe a data packet processing process. Switching equipment in the network interface card needs to send, to the virtual network function module on the VMM, the data packet from the VM1 for processing. Before sending the data packet, the first virtual machine determines that a destination media access control MAC address of the to-be-sent data packet is a MAC address of the uplink port of the virtual bridge on the VMM. The first virtual machine sends the data packet to the second virtual machine by using the VF that is connected to the first virtual machine. A destination IP address carried in the data packet is an IP address of the second virtual machine, and the destination MAC address carried in the data packet is the MAC address of the uplink port of the virtual bridge. The switching equipment of the first network port receives the data packet that is sent by the first virtual machine by using the VF, and performs layer 2 forwarding on the data packet according to the destination MAC address of the data packet to send the data packet to the virtual bridge by using the uplink port of the virtual bridge. The virtual bridge receives the data packet, changes the destination MAC address of the data packet to a MAC address of the second virtual machine, and sends a changed data packet to the virtual network function module. The destination MAC address of the data packet is changed, so that the changed MAC address may be used in a subsequent transmission process of data packet. This ensures that the data packet can be sent to the second virtual machine. After receiving the changed data packet, the virtual network function module performs network function processing on the changed data packet to provide abundant network function services for a user, and then sends the processed data packet to the second virtual machine according to the MAC address of the second virtual machine.

[0051] In this embodiment of the present invention, a virtual machine sends a data packet from a VF that is connected to the virtual machine. Switching equipment of a first network port forwards the data packet, and sends the data packet to a virtual bridge on a VMM. The virtual bridge transfers the data packet to a virtual network function module. The virtual network function module provides abundant network functions for the data packet. In a network interface card passthrough scenario, after sending a data packet from a passthrough VF, a virtual machine may send the data packet to a virtual network function module on a VMM in the foregoing manner. The software module is used to provide abundant network functions for a user, so as to perform virtual network function processing on the data packet.

the present invention, a VM1 on a host 1 sends a data packet to a VM2 on a host 2. A VMM of the host 1 includes a first virtual bridge and a first virtual network function module. A network interface card of the host 1 includes at least two network ports and first switching equipment. A VMM of the host 2 includes a second virtual bridge and a second virtual network function module. A network interface card of the host 2 includes at least two network ports and second switching equipment. Specifically, structures of the host 1 and the host 2 and functions of the components are shown in FIG. 2. Details are not described herein again. A process of sending the data packet to the VM2 by the VM1 includes:

[0067] Step 401: The VM1 sends, by using a VF that is connected to the VM1, the data packet whose destination is the VM2. A destination IP address of the data packet is an IP address of the VM2, and a destination MAC address of the data packet is a MAC address of an uplink port of the first virtual bridge on the host 1. The data packet carries a VLAN identifier of the VF.

[0068] Step 402: The first switching equipment of the network interface card of the host 1 receives the data packet, and sends the data packet to the first virtual bridge according to the destination MAC address of the data packet by using the uplink port.

[0069] Step 403: The first virtual bridge processes the data packet according to a preset flow entry, removes the VLAN identifier in the data packet, and changes the destination MAC address of the data packet to a MAC address of the VM2.

[0070] Step 404: The first virtual bridge sends a changed data packet to the first virtual network function module.

[0071] Step 405: The first virtual network function module performs network function processing on the data packet, including a security group, QoS, layer 2 tunnel encapsulation, distributed routing, and the like.

[0072] It should be noted that the first virtual network function module may perform network function processing by using various implementations in the prior art. This embodiment of the present invention imposes no limitation thereon.

[0073] Step 406: The first virtual network function module sends the processed data packet to an external physical switch of the host 1 by using a second network port, and the physical switch routes, through a network, the data packet to the host 2 on which the VM2 is

located.

[0074] It should be noted that the data packet carries the IP address and the MAC address of the VM2 and the data packet may be routed, in an existing routing manner, to the host 2 on which the VM2 is located. This embodiment of the present invention does not limit a transmission process of the data packet on the network. For example, a tunnel may be established between the host 1 on which the VM1 is located and the host 2 on which the VM2 is located, and endpoints of the tunnel are respectively the virtual network function module of the host 1 and that of the host 2, so that the tunnel required for data transmission is established between the two hosts for the VM1 and the VM2.

[0075] Step 407: The second virtual network function module of the host 2 on which the VM2 is located performs network function processing on the received data packet, and sends the processed data packet to the second virtual bridge.

[0076] Step 408: The second virtual bridge performs layer 2 forwarding according to the MAC address of the VM2, and sends, by using the second switching equipment, the data packet to a VF that is connected to the VM2, so as to send the data packet to the VM2.

[0077] For a data packet that is sent from an exterior of a host to a destination virtual machine on the host, the data packet is first transferred from a second network port to a virtual network function module; and then after having undergone virtual network function processing, the data packet is transferred to the destination virtual machine of the data packet by using a first network port.

[0078] The embodiment shown in FIG. 4 describes cross-host communication between virtual machines. It may be understood that a source virtual machine and a destination virtual machine of a data packet may be located on a same host. In this case, a data packet transmission process is similar to the process in FIG. 4. A difference lies in that: when the source virtual machine and the destination virtual machine of the data packet are located on a same host, a virtual network function module of the host sends the processed data packet to a virtual bridge after completing network function processing, and the virtual bridge sends the processed data packet to the destination virtual machine according to a MAC address of the destination virtual machine.

[0079] The host and an SDN controller may use a general-purpose computer device. For

example, as shown in FIG. 5, FIG. 5 is a schematic structural diagram of hardware of a computer device according to an embodiment of the present invention. The computer device 500 includes at least one processor 501, a communications bus 502, a memory 503, and at least one communications interface 504.

[0080] The processor 501 may be a general-purpose central processing unit (CPU), a microprocessor, an application-specific integrated circuit (application-specific integrated circuit, ASIC), or one or more integrated circuits configured to control program execution in the solutions of the present invention.

[0081] The communications bus 502 may include a path for transferring information between the foregoing components. The communications interface 504 uses an apparatus such as a transceiver to communicate with another device or a communications network such as an Ethernet, a radio access network (RAN), or a wireless local area network (Wireless Local Area Network, WLAN).

[0082] The memory 503 may be a read-only memory (read-only memory, ROM) or another type of static storage device that can store static information and instructions, a random access memory (random access memory, RAM) or another type of dynamic storage device that can store information and instructions, an electrically erasable programmable read-only memory (Electrically Erasable Programmable Read-Only Memory, EEPROM), a compact disc read-only memory (Compact Disc Read-Only Memory, CD-ROM) or another compact disc storage, an optical disc storage (including a compact disc, a laser disc, an optical disc, a digital versatile disc, a Blu-ray disc, or the like), a magnetic disk storage medium or another magnetic storage device, or any other medium that can be used to carry or store expected program code in a form of an instruction or a data structure and that can be accessed by a computer. No limitation is imposed thereon. The memory may exist alone and is connected to the processor by using the bus. Alternatively, the memory may be integrated with the processor.

[0083] The memory 503 is configured to store application program code for executing the solutions in the present invention, and the processor 501 controls the execution. The processor 501 is configured to execute the application program code stored in the memory 503.

[0084] In a specific implementation, in an embodiment, the processor 501 may include

one or more CPUs, for example, a CPU0 and a CPU1 in FIG. 5.

[0085] In a specific implementation, in an embodiment, the computer device 500 may include multiple processors, for example, the processor 501 and a processor 508 in FIG. 5. Each of the processors may be a single-core (single-core CPU) processor, or may be a multi-core (multi-core CPU) processor. The processor herein may be one or more devices, circuits, and/or processing cores that are configured to process data (for example, a computer program instruction).

[0086] In a specific implementation, in an embodiment, the computer device 500 may further include an output device 505 and an input device 506. The output device 505 communicates with the processor 501, and can display information in multiple manners. For example, the output device 505 may be a liquid crystal display (liquid crystal display, LCD), a light emitting diode (light emitting diode, LED) display device, a cathode ray tube (cathode ray tube, CRT) display device, a projector (projector), or the like. The input device 506 communicates with the processor 501, and may receive a user input in multiple manners. For example, the input device 506 may be a mouse, a keyboard, a touchscreen device, a sensing device, or the like.

[0087] The computer device 500 may be a general-purpose computer device or a dedicated computer device. In a specific implementation, the computer device 500 may be a desktop computer, a portable computer, a network server, a personal digital assistant (Personal Digital Assistant, PDA), a mobile phone, a tablet computer, a wireless terminal device, a communications device, an embedded device, or a device with a structure similar to the structure in FIG. 5. This embodiment of the present invention imposes no limitation on a type of the computer device 500.

[0088] The SDN controller and the host in FIG. 2 may be the device shown in FIG. 5. The memory stores one or more software modules for implementing functions of the host and the SDN controller. The host and the SDN controller may implement, by using the processor and the program code in the memory, the packet processing method disclosed in the embodiments of the present invention.

[0089] It should be noted that the computer device shown in FIG. 5 only provides possible hardware implementations of parts in a cloud computing system. According to differences or

the virtual bridge is a selected VF of the first network port, and the virtual bridge is connected to the network interface card by using the selected VF, where the selected VF that is connected to the virtual bridge is set to a promiscuous mode.

[0105] When the second virtual machine and the first virtual machine run on a same host, the virtual bridge is further configured to receive the processed data packet, and send the processed data packet to the second virtual machine according to the destination MAC address of the data packet by using the switching equipment.

[0106] When the second virtual machine and the first virtual machine run on different hosts, a tunnel is established between a virtual network function module of a host on which the first virtual machine is located and a virtual network function module of a host on which the second virtual machine is located. The processed data packet is transmitted from a second network port to the virtual network function module of the host on which the second virtual machine is located. After completing network function processing on the data packet, the virtual network function module of the host on which the second virtual machine is located sends the data packet to the second virtual machine by using a virtual bridge of the host on which the second virtual machine is located.

[0107] An embodiment of the present invention further provides a computer storage medium that is configured to store a computer software instruction used by the devices in FIG. 2 to FIG. 6. The computer software instruction includes a program designed for executing the foregoing method embodiments. A packet processing method in a cloud computing system may be implemented by executing a stored program.

[0108] In conclusion, according to the packet processing method in a cloud computing system, the host, and the system that are provided in the embodiments of the present invention, a packet sent by a virtual machine that uses a passthrough VF can be directed to a virtual bridge on a VMM, and further a virtual network function module on the VMM performs network function processing on the data packet, so as to provide abundant network functions for a user.

[0109] Although the present invention is described with reference to the embodiments, in a process of implementing the embodiments of the present invention that claims protection, a person skilled in the art may understand and implement another variation of the disclosed

them without departing from the scope of the present invention. Correspondingly, the specification and accompanying drawings are merely illustrative description of the present invention defined by the accompanying claims, and are considered as any of or all modifications, variations, combinations or equivalents that cover the scope of the present invention. Obviously, a person skilled in the art can make various modifications and variations to the present invention without departing from the scope of the present invention. The present invention is intended to cover these modifications and variations provided that they fall within the scope of protection defined by the following claims and their equivalent technologies.

CLAIMS

What is claimed is:

1. A host, comprising:

a virtual machine monitor (VMM) comprising a virtual bridge and a virtual network function module;

at least one network interface card comprising switching equipment and a first network port, wherein the first network port supports a network interface card virtualization capability and the first network port corresponds to at least one physical function (PF) and multiple virtual functions (VFs);

a first virtual machine connected to at least one VF of the first network port and configured to:

determine a destination media access control (MAC) address of a data packet to be sent, wherein the destination MAC address of the data packet is a MAC address of an uplink port of the virtual bridge, and

send the data packet to a second virtual machine by using the VF connected to the first virtual machine, wherein a destination IP address carried in the data packet is an IP address of the second virtual machine, and the destination MAC address carried in the data packet is the MAC address of the uplink port of the virtual bridge;

wherein the switching equipment is configured to receive the data packet, and send the data packet to the virtual bridge according to the destination MAC address of the data packet by using the uplink port of the virtual bridge;

wherein the virtual bridge is configured to receive the data packet, change the destination MAC address of the data packet to a MAC address of the second virtual machine, and send a changed data packet to the virtual network function module; and

wherein the virtual network function module is configured to perform network function processing on the changed data packet, and send a processed data packet, wherein a destination MAC address of the processed data packet is the MAC address of the second virtual machine.

2. The host according to claim 1, wherein the multiple VFs of the first network port have different virtual local area network (VLAN) identifiers.

3. The host according to claim 2, wherein:

the first virtual machine is further configured to send an Address Resolution Protocol (ARP) request packet by using the VF that is connected to the first virtual machine, wherein the ARP request packet carries a VLAN identifier of the VF that sends the ARP request packet;

the switching equipment is further configured to broadcast the ARP request packet, wherein the ARP request packet is sent to the virtual bridge by using the uplink port;

the virtual bridge is further configured to forward the ARP request packet to a software-defined networking (SDN) controller, and receive a flow entry returned by the SDN controller, wherein the flow entry comprises an ARP pickup setting for instructing the virtual bridge to construct an ARP response packet based on the ARP request packet, and the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet; and

the virtual bridge is further configured to return the ARP response packet to the first virtual machine, wherein the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet.

4. The host according to claim 3, wherein the flow entry further comprises a MAC address setting for instructing the virtual bridge to change the destination MAC address of the data packet to the MAC address of the second virtual machine when the virtual bridge receives the data packet that is sent by the first virtual machine to the second virtual machine.

5. The host according to claim 1, wherein:

the uplink port of the virtual bridge is a PF of the first network port, and the virtual bridge is connected to the network interface card by using the PF; or

the uplink port of the virtual bridge is a selected VF of the first network port, and the virtual bridge is connected to the network interface card by using the selected VF, wherein the selected VF that is connected to the virtual bridge is set to a promiscuous mode.

6. The host according to claim 1, wherein:

the second virtual machine and the first virtual machine run on the host; and

the virtual bridge is further configured to receive the processed data packet, and send the processed data packet to the second virtual machine according to the destination MAC address of the processed data packet by using the switching equipment.

7. The host according to claim 1, wherein:

the second virtual machine and the first virtual machine run on different hosts; and

the virtual network function module is configured to establish a tunnel between the virtual network function module and a second virtual network function module on a host on which the second virtual machine is located, send the processed data packet to a second network port of the network interface card, and send, by using the tunnel, the processed data packet to the host on which the second virtual machine is located enabling the second virtual network function module to send the processed data packet to the second virtual machine.

8. A packet processing method in a cloud computing system comprising at least one host comprising a virtual machine monitor (VMM) including a virtual bridge and a virtual network function module, and at least one network interface card including switching equipment and a first network port, wherein the first network port supports a network interface card virtualization capability, the first network port corresponds to at least one physical function (PF) and multiple virtual functions (VFs), and a first virtual machine is connected to at least one VF of the first network port, the method comprising:

determining, by the first virtual machine, a destination MAC address of a data packet to be sent, wherein the MAC address of the data packet is a MAC address of an uplink port of the virtual bridge;

sending, by the first virtual machine, the data packet to a second virtual machine by using the VF connected to the first virtual machine, wherein a destination IP address carried in the data packet is an IP address of the second virtual machine, and the destination MAC address carried in the data packet is the MAC address of the uplink port of the virtual bridge;

receiving, by the switching equipment, the data packet, and sending the data packet to the virtual bridge according to the destination MAC address of the data packet by using the uplink port of the virtual bridge;

receiving, by the virtual bridge, the data packet, changing the destination MAC address of the data packet to a MAC address of the second virtual machine, and sending a changed data packet to the virtual network function module; and

performing, by the virtual network function module, network function processing on the changed data packet, and sending a processed data packet, wherein a destination MAC address of the processed data packet is the MAC address of the second virtual machine.

9. The method according to claim 8, wherein the multiple VFs of the first network port have different virtual local area network (VLAN) identifiers.

10. The method according to claim 9, wherein determining, by the first virtual machine, a destination MAC address of a data packet to be sent comprises:

sending, by the first virtual machine, an ARP request packet by using the VF connected to the first virtual machine, wherein the ARP request packet carries a VLAN identifier of the VF that sends the ARP request packet;

broadcasting, by the switching equipment, the ARP request packet, wherein the ARP request packet is sent to the virtual bridge by using the uplink port;

forwarding, by the virtual bridge, the ARP request packet to a software-defined network (SDN) controller, and receiving a flow entry returned by the SDN controller, wherein the flow entry comprises an ARP pickup setting for instructing the virtual bridge to construct an ARP response packet based on the ARP request packet, and the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet; and

returning, by the virtual bridge, the ARP response packet to the first virtual machine, wherein the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet.

11. The method according to claim 10, wherein the flow entry further comprises a MAC address setting for instructing the virtual bridge to change the destination MAC address of the data packet to the MAC address of the second virtual machine when the virtual bridge receives the data packet sent by the first virtual machine to the second virtual machine.

12. The method according to claim 8, wherein:

the uplink port of the virtual bridge is a PF of the first network port, and the virtual bridge is connected to the network interface card by using the PF; or

the uplink port of the virtual bridge is a selected VF of the first network port, and the virtual bridge is connected to the network interface card by using the selected VF, wherein the selected VF that is connected to the virtual bridge is set to a promiscuous mode.

13. The method according to claim 8, wherein:

the second virtual machine and the first virtual machine run on the host; and

the method further comprises: receiving, by the virtual bridge, the processed data packet, and sending the processed data packet to the second virtual machine according to the destination MAC address of the processed data packet by using the switching equipment.

14. The method according to claim 8, wherein:

the second virtual machine and the first virtual machine run on different hosts; and

the method further comprises: establishing, by the virtual network function module, a tunnel between the virtual network function module and a second virtual network function module on a host on which the second virtual machine is located, sending the processed data packet to a second network port of the network interface card, and sending, by using the tunnel, the processed data packet to the host on which the second virtual machine is located for enabling the second virtual network function module to send the processed data packet to the second virtual machine.

15. A cloud computing system, comprising:

at least one host, comprising:

a virtual machine monitor (VMM) comprising a virtual bridge and a virtual network function module,

at least one network interface card comprising switching equipment and a first network port, wherein the first network port supports a network interface card virtualization capability, the first network port is corresponding to at least one physical function (PF) and multiple virtual functions (VFs), and

a first virtual machine connected to at least one VF of the first network port;

a software-defined network (SDN) controller configured to receive an ARP request packet from the first virtual machine that is forwarded by the virtual bridge, and return a flow entry to the virtual bridge, wherein the flow entry comprises an ARP pickup setting and a MAC address setting for instructing the virtual bridge to construct an ARP response packet for the first virtual machine, and the ARP response packet carries a MAC address of an uplink port of the virtual bridge that is used as a destination MAC address of a data packet to be sent;

wherein the first virtual machine is configured to send the data packet to a second virtual machine by using the VF that is connected to the first virtual machine, wherein a destination IP address carried in the data packet is an IP address of the second virtual machine, and the destination MAC address carried in the data packet is the MAC address of the uplink port of the virtual bridge;

wherein the switching equipment is configured to receive the data packet, and send the data packet to the virtual bridge according to the destination MAC address of the data packet by using the uplink port of the virtual bridge;

wherein the virtual bridge is configured to receive the data packet, change the destination MAC address of the data packet to a MAC address of the second virtual machine, and send a changed data packet to the virtual network function module; and

wherein the virtual network function module is configured to perform network function processing on the changed data packet, and send the processed data packet, wherein a destination MAC address of the processed data packet is the MAC address of the second virtual machine.

16. The system according to claim 15, wherein the multiple VFs of the first network port have different virtual local area network VLAN identifiers.

17. The system according to claim 16, wherein:

the first virtual machine is further configured to send the ARP request packet by using the VF that is connected to the first virtual machine, wherein the ARP request packet carries a VLAN identifier of the VF that sends the ARP request packet;

the switching equipment is further configured to broadcast the ARP request packet, wherein the ARP request packet is sent to the virtual bridge by using the uplink port;

the virtual bridge is further configured to forward the ARP request packet to the SDN controller, and receive the flow entry returned by the SDN controller, wherein the flow entry comprises the ARP pickup setting for instructing the virtual bridge to construct the ARP response packet based on the ARP request packet, and the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet; and

the virtual bridge is further configured to return the ARP response packet to the first virtual machine, wherein the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet.

18. The system according to claim 17, wherein the flow entry further comprises the MAC address setting for instructing the virtual bridge to change the destination MAC address of the data packet to the MAC address of the second virtual machine when the virtual bridge receives the data packet that is sent by the first virtual machine to the second virtual machine.

19. The system according to claim 15, wherein:

the uplink port of the virtual bridge is a PF of the first network port, and the virtual bridge is connected to the network interface card by using the PF; or

the uplink port of the virtual bridge is a selected VF of the first network port, and the virtual bridge is connected to the network interface card by using the selected VF, wherein the selected VF that is connected to the virtual bridge is set to a promiscuous mode.

20. The system according to claim 15, wherein:

the second virtual machine and the first virtual machine run on the at least one host; and

the virtual bridge is further configured to receive the processed data packet, and send the processed data packet to the second virtual machine according to the destination MAC address of the processed data packet by using the switching equipment.

21. The system according to claim 15, wherein:

the second virtual machine and the first virtual machine run on different hosts; and

the virtual network function module is configured to establish a tunnel between the virtual network function module and a second virtual network function module on a host on which the second virtual machine is located, send the processed data packet to a second network port of the network interface card, and send, by using the tunnel, the processed data packet to the host on which the second virtual machine is located for enabling the second virtual network function module to send the processed data packet to the second virtual machine.

22. A host, comprising:

a first processor;

a first memory;

at least one network interface card comprising a second processor, a second memory, and a first network port, wherein the first network port supports a network interface card virtualization capability and corresponds to at least one physical (PF) and multiple virtual functions (VFs);

wherein the first memory stores instructions which, when executed by the first processor, cause the host to:

implement a function of a first virtual machine, wherein the first virtual machine is connected to at least one VF of the first network port,

implement a function of a virtual bridge, and

implement a function of a virtual network function module;

wherein the second memory stores instructions which, when executed by the second processor, cause the host to:

implement a function of switching equipment;

wherein the first memory stores instructions, when executed by the first processor, cause the host to:

determine a destination MAC address of a data packet to be sent, wherein the MAC address of the data packet is a MAC address of an uplink port of the virtual bridge, and

send the data packet to a second virtual machine by using the VF that is connected to the first virtual machine, wherein a destination IP address carried in the data packet is an IP address of the second virtual machine, and the destination MAC address carried in the data packet is the MAC address of the uplink port of the virtual bridge;

wherein the second memory stores instructions, when executed by the second processor, further cause the host to:

receive the data packet, and send the data packet to the virtual bridge according to the destination MAC address of the data packet by using the uplink port of the virtual bridge; and

wherein the first memory stores instructions, when executed by the first processor, cause the host to:

receive the data packet, and change the destination MAC address of the data packet to a MAC address of the second virtual machine, and

perform network function processing on a changed data packet, and send the processed data packet, wherein a destination MAC address of the processed data packet is the MAC address of the second virtual machine.

23. The host according to claim 22, wherein:

the multiple VFs of the first network port have different virtual local area network (VLAN) identifiers;

the first memory stores instructions, when executed by the first processor, cause the host to:

send an ARP request packet by using the VF that is connected to the first virtual machine, wherein the ARP request packet carries a VLAN identifier of the VF that sends the ARP request packet;

the second memory stores instructions, when executed by the second processor, further cause the host to:

broadcast the ARP request packet, and send the ARP request packet to the virtual bridge by using the uplink port; and

the first memory stores instructions, when executed by the first processor, cause the host to:

forward the ARP request packet to a software-defined network (SDN) controller,

receive a flow entry returned by the SDN controller, wherein the flow entry comprises an ARP pickup setting for instructing the virtual bridge to construct an ARP response packet based on the ARP request packet, and the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet, and

return the ARP response packet to the first virtual machine, wherein the ARP response packet carries the MAC address of the uplink port of the virtual bridge that is used as the destination MAC address of the data packet.

24. The host according to claim 22, wherein the flow entry further comprises a MAC address setting for instructing the virtual bridge to change the destination MAC address of the data packet to the MAC address of the second virtual machine when the virtual bridge receives the data packet that is sent by the first virtual machine to the second virtual machine.