(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104980355 B
(45)授权公告日 2018.04.24

(21)申请号 201510245770.7

(22)申请日 2015.05.14

(65)同一申请的已公布的文献号
申请公布号 CN 104980355 A

(43)申请公布日 2015.10.14

(73)专利权人 华中科技大学
地址 430074 湖北省武汉市洪山区珞喻路1037号

(72)发明人 王芳 冯丹 史庆宇 朱挺炜 万勇

(74)专利代理机构 华中科技大学专利中心 42201
代理人 廖盈春

(51)Int.Cl.
H04L 12/761(2013.01)
H04L 12/801(2013.01)

(56)对比文件
CN 104518973 A,2015.04.15,
CN 103825828 A,2014.05.28,
CN 104486095 A,2015.04.01,
林东豪.一种可控组播实现方案.《福建电脑》.2010,(第5期),全文.
田金川.基于OpenFlow的可控组播的研究与实现.《中国优秀硕士学位论文全文数据库信息科技辑》.2014,(第7期),全文.

审查员 李晨

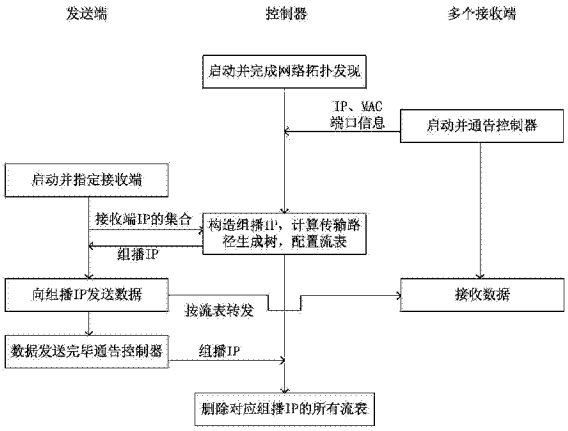
权利要求书1页 说明书7页 附图2页

(54)发明名称

一种SDN环境下的源端可控组播数据传输系统

(57)摘要

本发明公开了一种SDN环境下的源端可控组播数据传输方法及系统。本发明包括:发送端执行数据发送,自主指定接收端主机的集合,更灵活地控制数据传输,提高了数据接收的安全性;控制器完成数据控制服务,基于全局网络拓扑视图,对数据转发路径生成、数据传输等过程,进行集中控制;接收端完成数据接收,不同于传统网络中的组播成员程序,不需要与网络中间设备进行频繁的通告,只与控制器进行至多2次简单通信,其他时间只是等待或接收数据,减轻了网络设备的数据处理压力,节省了链路带宽。本发明还提供了实现上述方法的系统。本发明的源端可控特征,提高了数据传输的管控能力,基于SDN网络进行数据传输,提高了数据传输效率。



CN 104980355 B

1.一种SDN环境下的源端可控组播数据传输系统，其特征在于，包括发送端、接收端和控制器，其中：

所述发送端包括：指定目标接收端模块，用于接受用户输入的多个目标接收端IP，存入适合传输的数据结构；数据传输模块，用于将目标接收端IP以UDP数据包的形式发送给所述控制器，接收所述控制器反馈的携带目的IP的UDP数据包，向反馈的所述目的IP传输UDP数据直到数据发送完毕；以及数据传输结束处理模块，用于在数据发送完毕后，以UDP数据包的形式向所述控制器发送数据传输结束的通告；

所述接收端包括：信息收集模块，用于生成通告数据包并发送给所述控制器；以及数据接收模块，用于打开端口接收数据，当不再接收数据时，向所述控制器发送携带固定数据的通告数据包进行通告；

所述控制器包括：拓扑发现模块，用于通过链路层发现协议动态地发现网络设备的拓扑；流表管理模块，用于根据网络拓扑结构和所述发送端发送过来的目标接收端IP集合，生成合理的数据传输生成树，给传输路径中涉及的SDN网络设备配置流表；以及数据处理模块，解析收到的通告数据包，针对数据包头携带信息的不同，进行不同的处理：如果来自一个新的接收端，则解析出其IP、MAC和连接SDN网络设备的接口，存储到本地；如果来自一个已经存在的接收端，则检查是否为传输数据过程中退出的接收端，若是则重新更新数据传输路径并安装流表到相关SDN网络设备；如果来自所述发送端且为发送数据请求，则解析出其携带的所有目标接收端IP，根据IP查询到目的主机的MAC和连接SDN网络设备的接口信息，同时生成一个组播IP，再根据网络拓扑信息和合适的生成树算法生成数据转发路径，并以流表的方式安装到相关的SDN网络设备，最后将所述组播IP返回给所述发送端；如果来自所述发送端且为数据传输完毕的通告，则解析出携带的组播IP，根据该组播IP删除SDN网络设备中与之相关的流表项。

2.如权利要求1所述的SDN环境下的源端可控组播数据传输系统，其特征在于，所述控制器生成的组播IP不属于网络中任何一个设备，所述发送端发出ARP请求，所述ARP请求到达所述控制器，所述控制器对该ARP请求进行回复，使所述发送端进行数据发送。

3.如权利要求1或2所述的SDN环境下的源端可控组播数据传输系统，其特征在于，终止所述接收端程序时，向一个固定IP发送通告数据包，所述固定IP不属于网络中任何设备，其中，所述通告数据包携带描述该接收端的信息。

4.如权利要求1或2所述的SDN环境下的源端可控组播数据传输系统，其特征在于，所述控制器处理来自接收端的ARP请求包，通过目的IP判断是接收端启动时发送的还是终止时发送的：若是接收端启动时发送的ARP请求，则从请求包中解析出接收端主机的IP、MAC以及端口信息；若是接收端终止时发送的ARP请求，则查看数据结构保存的数据传输服务中是否存在该接收端，对存在该接收端的传输服务更新传输路径生成树。

一种SDN环境下的源端可控组播数据传输系统

技术领域

[0001] 本发明属于创新网络的数据传输技术领域，更具体地，涉及一种SDN 环境下的源端可控组播数据传输方法及系统。

背景技术

[0002] 软件定义网络(Software Defined Network，以下简称SDN)是一个创新网络架构，OpenFlow是实现SDN的一种技术方法，通过将数据转发层和控制层分离，实现了网络流量的灵活调度，为新的网络传输方案和应用提供了良好平台。

[0003] 基于SDN的OpenFlow实现技术至少包括三个部分：OpenFlow交换机、控制器和OpenFlow协议。其中，OpenFlow交换机存储了流表，流表由多个流表项构成，每个流表项对应一个数据转发规则；控制器完成网络中所有的数据控制逻辑。开源的控制器有很多，例如python编写的pox、ryu和 Java编写的floodlight等，控制器功能上包括3部分：1、底层通信模块，控制器与OpenFlow交换机之间通过网络套接字(以下简称socket)进行通信；2、OpenFlow协议，socket按照OpenFlow协议规则对数据进行处理； 3、上层应用，通过OpenFlow协议处理数据，在平台上部署自己编写的应用，例如具备2层交换机功能的应用，用户通过可编程的应用设计网络的控制逻辑。

[0004] SDN网络相比传统网络具有更好的数据传输效率，因为SDN减轻了网络中间设备的数据处理压力，使其仅仅充当数据转发设备。传统网络中为了提高数据传输效率，在多媒体应用、群视频服务等存在多个数据接收端的应用场景里，使用了组播技术，然而很多时候需要由发起者决定目的接收群体，仅仅使用传统的组播技术难以达到完全可控和安全性方面的要求。且在数据中心分布式的多副本存储场景中，需要将文件块发送到指定的目的主机，传统方法是进行多次单一的数据传输，浪费了链路带宽。

[0005] 对于上面应用场景的情况，主要问题是传统网络对网络数据流量的控制不够灵活，且传统组播技术无法满足某些应用场景需求。很有必要提高对数据传输的可控性，设计更佳的传输方案。

发明内容

[0006] 针对现有技术的以上缺陷或改进需求，本发明提供一种SDN环境下的源端可控组播数据传输方法及系统，在源端对数据接收对象的选择进行控制，利用SDN网络的高数据转发效率和高度可控性，提高了数据交换效率，同时接收端不用维持频繁的组播通告，减轻了通信压力，提高了链路带宽利用率。

[0007] 为实现上述目的，按照本发明的一个方面，提供了一种SDN环境下的源端可控组播数据传输方法，包括以下步骤：

[0008] 步骤1控制器启动并收集各SDN网络设备的连接信息，动态生成网络拓扑；

[0009] 步骤2各接收端启动并向所述控制器通告本机的IP、MAC和连接SDN 网络设备的接口信息，各接收端分别构造并发送通告数据包至一个固定IP，所述固定IP不属于网络中任

何设备；

[0010] 步骤3发送端指定多个接收端的IP地址，将该多个IP地址集合通告给所述控制器；

[0011] 步骤4所述控制器接收来自所述发送端的IP地址集合，根据网络拓扑和网络中的接收端信息，构造出传输路径生成树，生成和配置流表，并向所述发送端返回一个组播IP；

[0012] 步骤5所述发送端接收来自所述控制器的组播IP，向所述组播IP发送数据，数据根据SDN网络设备流表规则进行转发，到达各接收端；

[0013] 步骤6所述发送端通告所述控制器数据传输结束，并将所述组播IP通告控制器；

[0014] 步骤7所述控制器接收到数据传输结束的信息，并根据接收到的所述组播IP删除网络设备中对应的流表。

[0015] 按照本发明的另一方面，提供了一种SDN环境下的源端可控组播数据传输系统，包括发送端、接收端和控制器，其中：

[0016] 所述发送端包括：指定目标接收端模块，用于接受用户输入的多个目标接收端IP，存入适合传输的数据结构；数据传输模块，用于将目标接收端IP以UDP数据包的形式发送给所述控制器，接收所述控制器反馈的携带目的IP的UDP数据包，向反馈的所述目的IP传输UDP数据直到数据发送完毕；以及数据传输结束处理模块，用于在数据发送完毕后，以UDP数据包的形式向所述控制器发送数据传输结束的通告；

[0017] 所述接收端包括：信息收集模块，用于生成通告数据包并发送给所述控制器；以及数据接收模块，用于打开端口接收数据，当不再接收数据时，向所述控制器发送携带固定数据的通告数据包进行通告；

[0018] 所述控制器包括：拓扑发现模块，用于通过链路层发现协议动态地发现网络设备的拓扑；流表管理模块，用于根据网络拓扑结构和所述发送端发送过来的目标接收端IP集合，生成合理的数据传输生成树，给传输路径中涉及的SDN网络设备配置流表；以及数据处理模块，解析收到的通告数据包，针对数据包头携带信息的不同，进行不同的处理：如果来自一个新的接收端，则解析出其IP、MAC和连接SDN网络设备的接口，存储到本地；如果来自一个已经存在的接收端，则检查是否为传输数据过程中退出的接收端，若是则重新更新数据传输路径并安装流表到相关SDN网络设备；如果来自所述发送端且为发送数据请求，则解析出其携带的所有目标接收端 IP，根据IP查询到目的主机的MAC和连接SDN网络设备的接口信息，同时生成一个组播IP，再根据网络拓扑信息和合适的生成树算法生成数据转发路径，并以流表的方式安装到相关的SDN网络设备，最后将所述组播IP返回给所述发送端；如果来自所述发送端且为数据传输完毕的通告，则解析出携带的组播IP，根据该组播IP删除SDN网络设备中与之相关的流表项。

[0019] 总体而言，通过本发明所构思的以上技术方案与现有技术相比，具有以下有益效果：

[0020] 本发明使用了源端可控的组播传输方式，能更加灵活和安全地进行点对多点的数据传输，利用了SDN网络环境，将数据转发层和数据控制层分离，更加合理地管理网络流量，提高了数据传输效率。本发明可应用到特定的网络应用场景，例如多媒体服务、数据中心分布式的多副本存储过程等。

附图说明

[0021] 图1为本发明SDN环境下的源端可控组播数据传输方法的流程图；

[0022] 图2为本发明SDN环境下的源端可控组播数据传输系统的结构框图；

[0023] 图3为本发明利用Mininet构建的实施例的网络拓扑图。

具体实施方式

[0024] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。此外，下面所描述的本发明各个实施方式中所涉及到的技术特征只要彼此之间未构成冲突就可以相互组合。

[0025] 本发明的技术方案为：接收端将自己的IP、MAC和网络端口信息发送给控制器；发送端先通告控制器其目的主机的集合，控制器生成一个目的组播IP，配置SDN网络设备中的流表转发规则，生成合理的数据转发生成树，并将目的组播IP通告给发送端，然后发送端向该目的组播IP发送数据，数据根据SDN网络设备中的流表规则进行正确的转发，最终到达指定目的主机集合。发送端作为源主机接受输入IP来指定接收主机，使系统控制能力更强；接收端仅在启动和终止时与控制器进行简单通信，主要工作是接收数据，与传统网络中的组播成员相比，不存在与交换机或路由器频繁的信息通告，通信过程更加简单，减轻了网络设备处理数据的压力；控制器是数据控制逻辑的核心，主要用于动态获得网络拓扑、计算合理的数据传输路径和配置流表。

[0026] 本发明实施例在Linux环境下，利用Mininet构建支持OpenFlow的虚拟网络，并采用OpenFlow和Pox开源控制器来实现本发明。

[0027] 图1所示为本发明SDN环境下的源端可控组播数据传输方法的流程图，具体包括以下步骤：

[0028] 步骤1控制器启动并收集各SDN网络设备的连接信息，动态生成网络拓扑；

[0029] 步骤2各接收端启动并向控制器通告本机的IP、MAC和连接SDN网络设备的接口信息，各接收端分别构造并发送通告数据包至一个固定IP，该固定IP不属于网络中任何设备。在本发明实施例中，通告数据包为UDP数据包，但不以此为限。该通告数据包可选择携带更多描述接收端接收过程的信息，例如该接收端本次传输开放的端口，用于细化构造的流表规则，有利于控制器形成更好的数据控制；

[0030] 步骤3发送端指定多个目的主机(即接收端)的IP地址，将该多个 IP地址集合通告给控制器；

[0031] 步骤4控制器接收来自发送端的IP地址集合，根据网络拓扑和网络中的接收端主机信息，构造出传输路径生成树，生成和配置流表，并向发送端返回一个组播IP；

[0032] 步骤5发送端接收到来自控制器的组播IP，向该组播IP发送数据，数据根据SDN网络设备流表规则进行转发，到达各接收端；

[0033] 步骤6发送端通告控制器数据传输结束，并将该组播IP通告控制器；

[0034] 步骤7控制器接收到数据传输结束的信息，并根据接收到的组播IP删除网络设备中对应的流表。

[0035] 图2所示为本发明SDN环境下的源端可控组播数据传输系统的结构框图，包括发送端、接收端和控制器，其中：

[0036] 发送端包括:指定目标接收端模块:用于接受用户输入的多个目标接收端IP,存入适合传输的数据结构;数据传输模块:用于将目标接收端IP 以用户数据报协议(User Datagram Protocol,以下简称UDP)数据包的形式发送给控制器,接收控制器反馈的携带目的IP的UDP数据包,向反馈的目的IP传输UDP数据直到数据发送完毕;数据传输结束处理模块:用于在数据发送完毕后,以UDP数据包的形式向控制器发送数据传输结束的通告;

[0037] 接收端包括:信息收集模块:用于接收端启动后,生成携带固定数据的UDP数据包,发送给控制器;数据接收模块:用于打开端口接收数据,当不再接收数据时,向控制器发送携带固定数据的UDP数据包进行通告;

[0038] 控制器包括:拓扑发现模块:用于通过链路层发现协议动态地发现SDN 网络拓扑结构;流表管理模块:用于根据网络拓扑结构和发送端发送过来的目标接收端IP集合,生成合理的数据传输生成树,给传输路径中涉及的 SDN网络设备配置流表;数据处理模块,用于解析收到的UDP数据包,针对数据包头携带信息的不同,进行不同的处理。如果来自一个新的接收端,则解析出其IP、MAC和连接SDN网络设备的接口,存储到本地;如果来自一个已经存在的接收端,则检查是否为传输数据过程中退出的接收端,若是则重新更新数据传输路径并安装流表到相关SDN网络设备;如果来自发送端且为发送数据请求,则解析出其携带的所有目的IP,根据IP查询到目的主机的MAC和连接SDN网络设备的接口信息,同时生成一个组播IP,再根据网络拓扑信息和合适的生成树算法生成数据转发路径,并以流表的方式安装到相关的SDN网络设备,最后将该组播IP返回给发送端。由于生成的组播IP不属于网络中任何一个设备,故发送端会发出ARP请求,ARP请求到达控制器,控制器对该ARP请求进行回复,使发送端接下来能发送真实的数据;如果来自发送端且为数据传输完毕的通告,则解析出携带的组播IP,根据该组播IP删除SDN网络设备中与之相关的流表项。

[0039] 图3所示为本发明利用Mininet构建的实施例的网络拓扑图,包含4 台OpenFlow交换机、4台主机host0～host3、Pox控制器(用于远程控制 OpenFlow交换机)。

[0040] 结合图1,其具体实施过程如下:

[0041] 接收端接收数据的具体过程为:

[0042] (A1)启动接收端程序,使用socket编程构造通告数据包发送至一个固定IP(记为IP1),IP1不属于网络中任何设备,在本发明实施例中,该通告数据包携带固定的几个字节无用信息。该通告数据包触发的地址解析协议(Address Resolution Protocol,以下简称ARP)请求包到达控制器,控制器进行相应处理;

[0043] (A2)开放端口等待接收数据;

[0044] (A3)终止接收端程序时,向另一固定IP(记为IP2)发送UDP数据包,IP2不属于网络中任何设备,在本发明实施例中,该通告数据包携带固定的几个字节无用信息。该UDP数据包触发的ARP请求包到达控制器,控制器对事件判断并进行相应处理。

[0045] 在本发明实施例中,接收端的行为与传统网络中的组播成员不同,不与网络设备进行频繁的通告信息通信,只与SDN网络中的控制器进行至多2 次通信,主要完成数据接收工作,减轻了中间网络设备的数据处理压力。

[0046] 发送端发送数据的具体过程为:

[0047] (B1)启动发送端程序,接受用户要求的接收端IP集合,将这些IP 按序存放到字符数组,使用socket编程构造携带该字符数组的UDP数据包,向IP1发送。因为网络设备中不存

在有关IP1的流表，所以该UDP数据包到达控制器，由控制器进行相应的处理；

[0048] (B2)发送端利用socket编程开放端口，等待控制器发送过来的组播IP；

[0049] (B3)构造携带数据块且目的IP为收到的组播IP的UDP数据包，发送真正的数据；

[0050] (B4)数据传输完成，构造携带本次传输使用的组播IP的UDP数据包，其目的IP为IP2，同样地，该数据包会到达控制器，由控制器解析并进行相应处理。

[0051] 在本发明实施例中，发送端的行为不同于传统网络中的组播发送方，不是简单地向规定的组播IP发送数据，而是自主指定目的接收端和获得动态组播IP，能更灵活地对传输服务进行控制，也提高了传输的安全性。

[0052] 控制器端进行数据逻辑控制的具体过程为：

[0053] (C1)在本发明的数据传输系统中，控制器最先运行。控制器通过链路层发现协议构造链路层发现包，动态地生成网络拓扑；

[0054] (C2)控制器通过检查收到的所有数据包类型和数据包目的地址等信息，判断该数据包的含义；

[0055] (C3)处理来自接收端的ARP请求包，通过目的IP判断是接收端启动时发送的还是终止时发送的。若是接收端启动时发送的ARP请求，则从请求包中解析出接收端主机的IP、MAC以及端口信息；若是接收端终止时发送的ARP请求，则查看数据结构保存的数据传输服务中是否存在该接收端，凡是存在该接收端的传输服务都需要更新传输路径生成树。针对这两种ARP 请求构造ARP回复包，防止接收端继续发送多余的ARP请求；

[0056] (C4)处理来自发送端的UDP数据包，通过目的IP判断是发送端启动时发送的还是数据传输完成时发送的。若是发送端启动发送的UDP数据包，则构造一个组播IP，从该UDP数据包中解析出所有的目的主机IP，通过IP 查询保存下来的接收端主机信息，再根据网络拓扑结构进行定位，设计构造合理的数据传输生成树，给传输路径中涉及的SDN网络设备配置流表，最后构造携带该组播IP的UDP数据包发送给发送端；若是数据传输完成时发送的UDP数据包，则解析出携带的组播IP，根据组播IP判断该传输服务属于哪个传输路径生成树，把该传输路径生成树的流表项都删除。

[0057] 下文结合图1和图3进行具体说明，在本发明实施例中，数据控制逻辑在Pox控制器的上层应用中实现，数据转发逻辑在OpenFlow交换机的流表中体现，host0运行发送端程序，host1、host2和host3运行接收端程序。实施例的具体运行步骤如下：

[0058] 步骤1编写运行于Pox平台的Pox应用，启动Pox平台。该Pox应用运行后，完成网络拓扑发现，将网络设备的连接信息保存到字典adjacency，通过查询字典adjacency，建立对应每2个OpenFlow交换机间基于最短距离的最短路径字典path_map。

[0059] 步骤2启动host1(IP:10.0.0.1,MAC:00:00:00:00:00:01)、host2 (IP:10.0.0.2,MAC:00:00:00:00:00:02)和host3(IP:10.0.0.3,MAC: 00:00:00:00:00:03)上的接收端程序，每个host都通过socket UDP方式向IP1:10.0.0.90发送一个UDP数据包，该IP1被定义为特殊IP，其不被网络中任何设备使用。该UDP数据包触发的ARP请求包会到达Pox控制器，被Pox控制器上的PacketIn函数捕获，该函数通过检查ARP请求包的目的地址和本地有关组播成员主机的信息，判断为一个新的组播成员，故将其 IP、MAC等信息保存到字典，接收端程序进入接收数据状态。

[0060] 步骤3启动host0(IP:10.0.0.4,MAC:00:00:00:00:00:04)上的发送端程序，输入

host1、host2和host3的IP(即10.0.0.1、10.0.0.2和 10.0.0.3)，将这3个IP组成一定规则的字符串利用socket UDP数据包发送出去，该UDP数据包的目的IP为10.0.0.90(即IP1)，然后进入等待接收UDP数据包的状态。

[0061] 步骤4上述发送端发送的UDP数据包被Pox控制器上的PacketIn函数捕获，通过检查UDP数据包的目的IP和目的端口，判断为一个发送端请求，故按照固定的格式解析出UDP数据包中携带的3个IP。查询保存了组播成员主机信息的字典adjacency以及最短路径字典path_map，基于组播树最短路径计算合理的数据传输路径，得到每台OpenFlow交换机对应的数据入端口和出端口，同时随机生成一个组播IP(记为IP*:10.0.0.100)，该 IP*不用于网络中的任何设备。将IP*作为key，将OpenFlow交换机id、入端口和出端口作为value，保存到查询字典mst中。根据字典mst对每个对应该组播IP的OpenFlow交换机发送配置流表的通告，通告包含数据包经过该OpenFlow交换机的转发规则，例如数据包的目的IP、入端口、出端口。对于边缘交换机(即直接与主机相连的OpenFlow交换机)，还需要调用Pox 库函数修改目的IP和目的MAC。完成流表的配置后，构造携带IP*的UDP数据包发送给host0。

[0062] 与host0相连的OpenFlow交换机流表：

[0063] 源IP:10.0.0.4，目的IP:10.0.0.100，入端口:1，出端口:4。

[0064] 与host1相连的OpenFlow交换机流表：

[0065] 源IP:10.0.0.4，目的IP:10.0.0.100，入端口:2，出端口:3、4，更改目的IP为:10.0.0.1，更改目的MAC为:00:00:00:00:00:01。

[0066] 因为对流表的处理会按照规则的先后顺序进行，当匹配了源IP、目的 IP和入端口的数据流，进入与host1相连的OpenFlow交换机时，数据从端口3正常输出(未更改目的IP和目的MAC)，然后更改目的IP和目的MAC，从端口4输出。故出端口为3的数据包目的IP和MAC并没有被更改，只有出端口为4的数据包会被更改目的IP和MAC，host1会接收到数据包。

[0067] 与host2、host3相连的OpenFlow交换机流表：

[0068] 源IP:10.0.0.4，目的IP:10.0.0.100，入端口:1，出端口:3，更改目的IP为:10.0.0.3，更改目的MAC为:00:00:00:00:00:03，出端口： 4，更改目的IP为:10.0.0.2，更改目的MAC为:00:00:00:00:00:02。

[0069] 步骤5host0接收到IP*后开始向IP*发送数据包，数据包经OpenFlow 交换机的流表规则进行转发，最终到达host1、host2和host3。

[0070] 步骤6假如在数据传输过程中host1手动终止接收程序，则通过 socket UDP方式向IP2:10.0.0.91发送一个数据包，该IP2被定义为特殊 IP，不被网络中任何设备使用，故ARP请求包会到达Pox控制器，被Pox 控制器上的PacketIn函数捕获，该函数通过判断ARP请求包的目的地址和本地有关组播成员主机的信息，查询字典mst，判断该接收端对应一个正在传输数据的传输路径，故将有关该主机的信息从相关字典中删除，更新字典mst和重新配置OpenFlow交换机流表。

[0071] 当host0输出发送完毕后，通过socket UDP方式向IP2:10.0.0.91发送一个携带IP*的数据包进行通告。

[0072] 步骤7因为网络设备中不存在有关IP2的流表，所以host0发送的携带IP*的UDP数据包会到达Pox控制器，被PacketIn函数捕获，通过检查 UDP数据包的目的IP，判断为一个传输完毕的通告，故解析出数据包中的 IP*，删除字典mst中对应IP*的键值对，同时向相关

的OpenFlow交换机发送删除相关流表项的通告。

[0073] 本领域的技术人员容易理解，以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

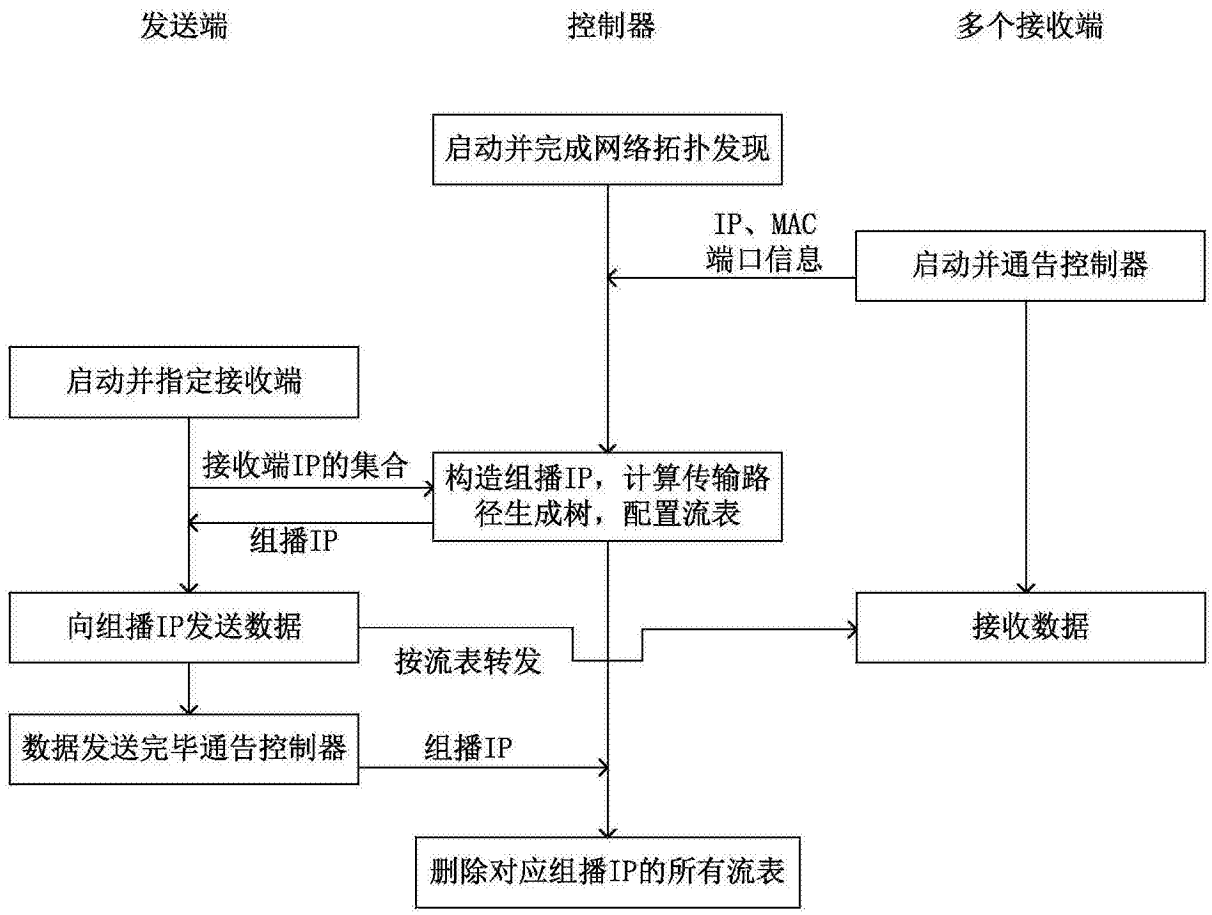

发送端
控制器
多个接收端
启动并完成网络拓扑发现
IP、MAC
端口信息
启动并通告控制器
启动并指定接收端
接收端IP的集合
构造组播IP，计算传输路径生成树，配置流表
组播IP
向组播IP发送数据
按流表转发
接收数据
数据发送完毕通告控制器
组播IP
删除对应组播IP的所有流表

图1

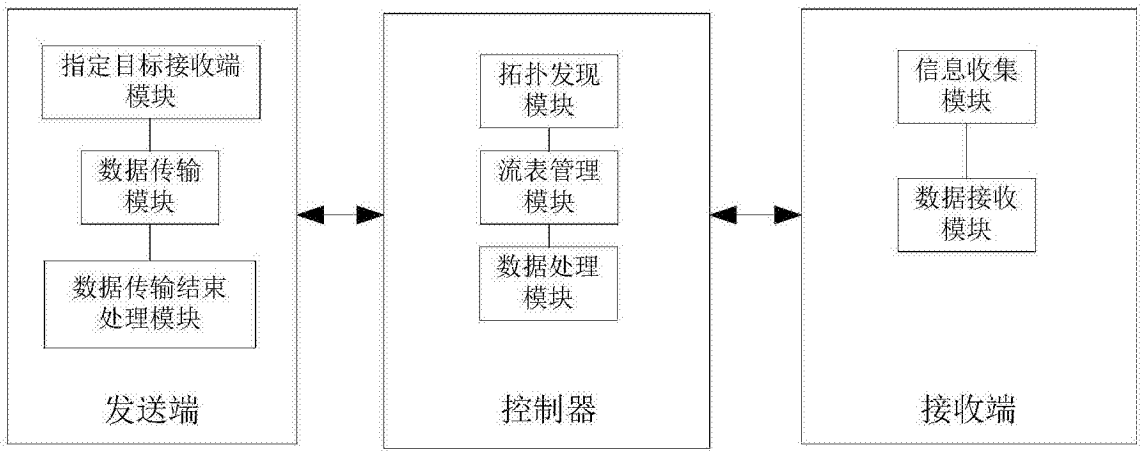

指定目标接收端模块
数据传输模块
数据传输结束处理模块
发送端
拓扑发现模块
流表管理模块
数据处理模块
控制器
信息收集模块
数据接收模块
接收端

图2

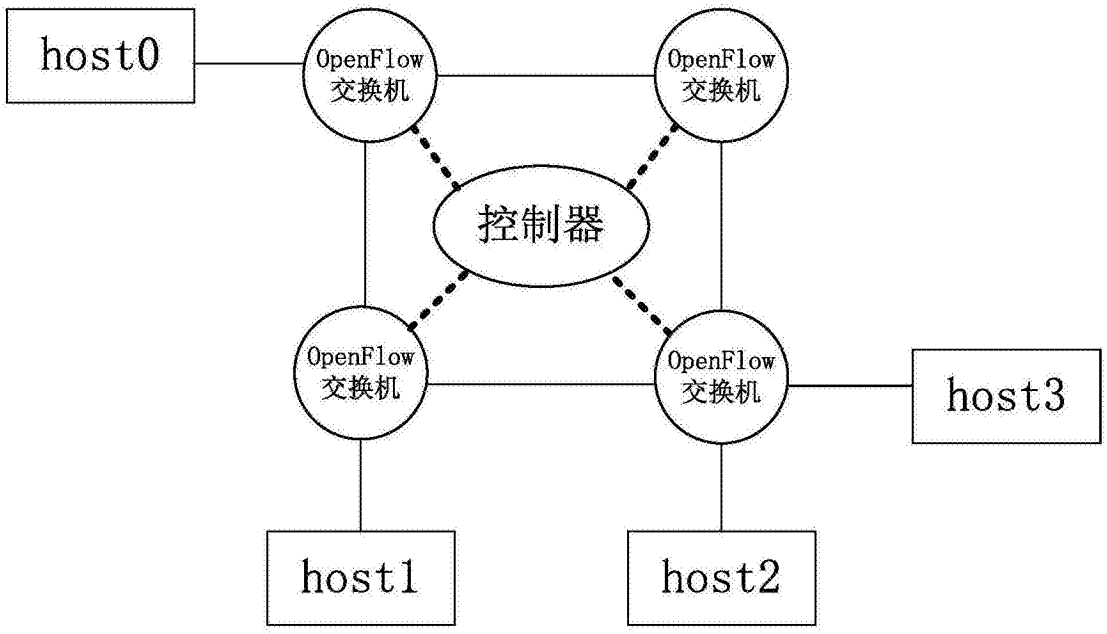

host0
OpenFlow
交换机
OpenFlow
交换机
控制器
OpenFlow
交换机
OpenFlow
交换机
host3
host1
host2

图3