



(12)发明专利

(10)授权公告号 CN 104253749 B

(45)授权公告日 2018.04.13

(21)申请号 201410479614.2

(56)对比文件

(22)申请日 2014.09.18

US 8787154 B1,2014.07.22,

(65)同一申请的已公布的文献号

审查员 陈莹

申请公布号 CN 104253749 A

(43)申请公布日 2014.12.31

(73)专利权人 华南理工大学

地址 510640 广东省广州市天河区五山路
381号

(72)发明人 官权升 陈天宇 陈晓菲 王妍莹
陈涵

(74)专利代理机构 广州市华学知识产权代理有
限公司 44245

代理人 蔡茂略

(51)Int.Cl.

H04L 12/721(2013.01)

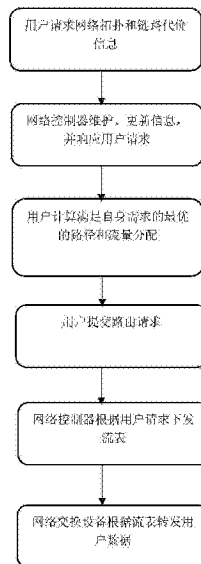
权利要求书1页 说明书4页 附图2页

(54)发明名称

一种基于软件定义网络架构的用户端分布式
路径计算方法

(57)摘要

本发明公开的一种基于软件定义网络架构
的用户端分布式路径计算方法,包含以下步骤:
用户通过控制器的北向接口,向网络控制器获取
网络拓扑和链路代价信息,在此基础上进行路径
计算,确定路由和流量分布,并向控制器提交路
由请求;网络控制器负责更新网络拓扑和链路状
态,并根据用户的路由选择向数据转发设备下发
流表。本发明的方法,用户路径计算使得用户参
与网络控制,实现网络资源使用的分布式优化;
同时,控制器通过向用户提供网络拓扑信息,可
以实现网络资源和功能对用户的虚拟化,达到网
络资源的优化配置,集约使用。本发明使得用户
获得了在网络安全范围内的极大的权利,提高了
网络的透明度,让SDN架构服务于用户。



1. 一种基于软件定义网络架构的用户端分布式路径计算方法,其特征在于,包括以下顺序的步骤:

(1) 用户向网络控制器请求网络拓扑和链路代价信息;

(2) 网络控制器维护和更新网络拓扑和链路代价信息,并响应用户请求;

(3) 用户根据步骤(2)收到的网络拓扑和链路代价信息,计算满足自身需求的最优的路径和流量分配;

(4) 根据步骤(3)的计算结果,用户向网络控制器提交路由请求,所述路由请求携带了步骤(3)的计算结果;

(5) 网络控制器根据用户的路由请求下发流表,所述流表即为步骤(3)的计算结果;

(6) 网络交换设备根据流表转发用户数据。

2. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:用户和网络控制器之间通过软件定义网络的北向接口进行交互。

3. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:步骤(2)中,所述的网络控制器可以与网络交换/转发设备进行通信,控制网络中数据的转发、获取网络的拓扑结构和链路的信息,包括带宽、时延、流量。

4. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:步骤(2)中,所述的网络控制器对网络资源和功能虚拟化管理,并向用户在步骤(1)中的请求返回虚拟化后的网络拓扑信息。

5. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:步骤(2)中,所述的网络控制器通过调整链路而影响用户路径计算,达到网络资源的优化使用。

6. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:步骤(3)中,所述的客户端路径计算确定了该用户业务流在网络中的转发路径和相应路径的流量。

7. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:所述步骤(1)-(6)由用户的网络通信需求触发,并能够多次重复以进行动态路径计算。

8. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:所述步骤(1)、(3)和(4)在用户端执行;所述步骤(2)和(5)在网络控制器中执行。

9. 根据权利要求1所述的基于软件定义网络架构的用户端分布式路径计算方法,其特征在于:所述步骤(3)的路由决策和步骤(4)的路由请求由用户手动完成,或者由软件中间件自动完成。

一种基于软件定义网络架构的用户端分布式路径计算方法

技术领域

[0001] 本发明涉及软件定义网络、网络的控制和管理,以及网络路径计算等领域,特别涉及一种基于软件定义网络架构的用户端分布式路径计算方法。

背景技术

[0002] 传统通信网络依赖OSI七层网络协议栈工作,网络的控制面分散在不同的设备中,网络协议运行的透明度低。对于用户来说,应用程序如何使用网络协议栈、信息的安全性不能直观掌握也无法控制;对于网络管理人员,网络如同黑匣子一样,难以控制网络中的流量分布,不利于对网络设备的维护和优化;而对于网络运营商来说,每一次升级设备、更改运营收费方式需要对所有的网络设备进行更新换代,更是极为不便。

[0003] 传统网络的缺陷诟病已久,弹性计算、虚拟网络技术、MPLS等的出现虽然在一定程度上缓解了传统网络的缺点,但是却大大增加了网络的复杂度,给网络管理人员、研究人员带来了巨大的压力,同时也增加了网络协议升级、修改的难度和成本。这使得传统网络越来越庞大和趋于静态,与当今对动态网络的需求相逆。

[0004] 当前网络发展的趋势是网络的控制面集中化,而且逐步向用户开放。软件定义网络(software defined network,SDN)已经实现了网络控制面的集中化。近年,数据中心等网络需求大户相继出现,他们有参与网络管理,降低网络运营成本的动力。本创新方案提供一种在SDN控制器协调下,用户参与网络控制,具体为参与网络路径计算的机制。

发明内容

[0005] 本发明的目的在于克服现有技术的缺点与不足,提供一种基于软件定义网络架构的用户端分布式路径计算方法。

[0006] 本发明的目的通过以下的技术方案实现:

[0007] 一种基于软件定义网络架构的用户端分布式路径计算方法,包括以下步骤:

[0008] (1)通过软件定义网络的北向接口,用户与网络控制器进行交互,向网络控制器请求网络拓扑和链路代价信息;

[0009] (2)在软件定义网络架构下,网络控制器与网络交换/转发设备通过南向接口(如OpenFlow协议)进行通信,控制网络中数据的转发、获取网络的拓扑结构和链路的信息,包括带宽、时延、流量等;

[0010] 网络控制器周期性的维护和更新网络拓扑和链路代价信息,并对网络资源和功能进行管理,响应步骤(1)的用户请求;

[0011] (3)用户根据收到的网络拓扑和链路代价信息,计算满足自身需求的最优的路径和流量分配,确定了自身业务流在网络中的转发路径和相应路径的流量;

[0012] (4)根据步骤(3)的计算结果,用户通过北向接口向网络控制器提交路由请求,所述路由请求携带了步骤(3)的计算结果;

[0013] (5)网络控制器收到用户的路由请求后下发流表,所述流表即为步骤(3)的计算结

果;流表中包括源节点和目的节点等信息,实现数据包的匹配和转发;

[0014] (6)网络交换设备收到流表后写入设备中,数据包根据流表进行匹配,完成数据的转发。

[0015] 上述步骤(1)-(6)由用户的网络通信需求触发,并能够多次(如周期性)重复以进行动态路径计算。

[0016] 步骤(2)所述网络控制器使用虚拟化(virtualization)技术或覆盖(overlay)技术对网络资源和功能管理;同时调整链路代价信息。该动作能够影响步骤(3)的用户路径计算结果,以达到网络资源的优化使用。

[0017] 所述步骤(3)的路由决策和步骤(4)的路由请求由用户手动完成,或者由软件中间件自动完成。

[0018] 本发明与现有技术相比,具有如下优点和有益效果:

[0019] (1)用户参与网络控制与路径计算。传统网络将网络协议封装在硬件设备中,用户无法了解网络运行的情况,也不能按照自己的需求选择合适的链路和设备。在SDN网络下,网络管理功能上升至SDN控制器。用户可以通过REST API指令控制SDN中央控制器,令控制器下发流表满足选路请求。据此,我们提出让用户参与网络控制与路径计算,改变了传统网络对用户的局限性,大大提高了网络的透明度和用户的权力,使得网络从软件定义走向用户定义。

[0020] 用户向SDN数据中心控制器提出业务请求,控制器分析请求后给出拓扑结构,由用户选路后提交选路请求,控制器再下发流表到数据中心的网络设备。充分的体现SDN网络中,应用程序和网元设备应实现紧耦合的交互,服务开通配置、分析和优化三个阶段应紧密结合的特点,展示了网络可编程性的优越性。

[0021] (2)网络控制器实现网络(资源)虚拟化,达到网络资源的优化配置。本方案中,网络的拓扑信息对用户的行为有至关重要的影响。出于安全或者网络资源优化的考虑,控制器为用户提供基于实际物理承载网(underlay)的叠加(overlay)虚拟网络拓扑结构,实现了网络资源的虚拟化和网络功能虚拟化,实现网络资源的优化配置。

[0022] 用户端的路径计算还可以减轻控制器集中计算的压力,实现分布式计算,从而优化网络的资源使用。

[0023] (3)技术与商业模式结合。传统网络分散的控制面很难提供灵活的网络运营模式。当前的网络运营模式一般有两种:对网络使用时间或对网络流量收费,不能实现对网络资源(包括带宽和链路)的使用实现更精细化运营收费。如根据跳数的不同采取不同的收费标准,根据占用带宽的不同采用不同的收费。本方案中,对资源的使用费用可以体现在链路或路径的代价中,通过控制器发布给用户,使得本技术方案与网络的运营模式紧密结合起来。同时,升级、改变网络设备只需要修改、添加SDN控制的模块即可,大大减少了运营商的运行和维护成本。

附图说明

[0024] 图1为本发明所述的一种基于软件定义网络架构的用户端分布式路径计算方法的流程图;

[0025] 图2为图1所述方法在软件定义网络架构下,采用Floodlight Controller开源控

制器控制下的网络结构图；

[0026] 图3为图1所述方法在Floodlight Controller控制器下展示的示例中建立的拓扑结构图。

具体实施方式

[0027] 下面结合实施例及附图对本发明作进一步详细的描述,但本发明的实施方式不限于此。

[0028] 如图1,一种基于软件定义网络架构的用户端分布式路径计算方法,包括以下步骤:

[0029] (1)通过软件定义网络的北向接口,用户与网络控制器进行交互,向网络控制器请求网络拓扑和链路代价信息;

[0030] (2)在软件定义网络架构下,网络控制器与网络交换/转发设备通过南向接口(如OpenFlow协议)进行通信,控制网络中数据的转发、获取网络的拓扑结构和链路的信息,包括带宽、时延、流量等;

[0031] 网络控制器周期性的维护和更新网络拓扑和链路代价信息,并对网络资源和功能进行管理,响应步骤(1)的用户请求;

[0032] (3)用户根据收到的网络拓扑和链路代价信息,计算满足自身需求的最优的路径和流量分配,确定了自身业务流在网络中的转发路径和相应路径的流量;

[0033] (4)根据步骤(3)的计算结果,用户通过北向接口向网络控制器提交路由请求,所述路由请求携带了步骤(3)的计算结果;

[0034] (5)网络控制器收到用户的路由请求后下发流表,所述流表即为步骤(3)的计算结果;流表中包括源节点和目的节点等信息,实现数据包的匹配和转发;

[0035] (6)网络交换设备收到流表后写入设备中,数据包根据流表进行匹配,完成数据的转发。

[0036] 上述步骤(1)-(6)由用户的网络通信需求触发,并能够多次(如周期性)重复以进行动态路径计算。

[0037] 步骤(2)所述网络控制器使用虚拟化(virtualization)技术或覆盖(overlay)技术对网络资源和功能管理;同时调整链路代价信息。该动作能够影响步骤(3)的用户路径计算结果,以达到网络资源的优化使用。

[0038] 所述步骤(3)的路由决策和步骤(4)的路由请求由用户手动完成,或者由软件中间件自动完成。

[0039] 下面以Floodlight Controller开源控制器与Mininet虚拟网络为实例对本发明作进一步详细的描述,但本发明的实施方式不限于此。

[0040] 为实现Floodlight Controller开源控制器下的用户端分布式路径计算,首先要在PC机器上安装Floodlight,作为软件定义网络架构的控制器。PC机器上VMware Workstation中运行带有Mininet的虚拟机,用来模拟出OpenFlow交换机,主机等自定义的网络拓扑结构,结构如图2所示。

[0041] 具体实现步骤如下:

[0042] (1)在Mininet中新建一个网络拓扑图,输入的指令如下:

[0043] >>sudo mn--controller=remote--ip=192.168.246.1--port=6633--custom/home/openflow/mininet/custom/topo_tlink_4ll.py--topo topo4ll

[0044] 上述指令可以实现建立一个基于OpenFlow协议的Mininet拓扑结构,并由宿主机进行远程控制,数据交换端口为6633。此拓扑结构由四台交换机s1、s2、s3、s4组成,一台主机h5,一台服务器h6组成。Floodlight具有可视化界面,可在localhost:8080/ui/index.html界面中显示。在浏览器打开localhost:8080/ui/index.html获取当前拓扑结构如图3所示。

[0045] (2) 用户通过控制器北向接口REST API指令向网络控制器Floodlight Controller请求网络拓扑和链路代价信息。此处需要使用的REST API指令是GET指令,如下所示。

[0046] `nodedict=nodes.get(data);testlist=links.get(data)`

[0047] (3) 通过GET指令获取到全网拓扑结构之后,网络控制器Floodlight Controller通过内部的监听模块维护和更新网络拓扑和链路代价信息,并响应用户请求。用户可获取的全网拓扑信息,包括网络中所有交换机及其id编号,以及链路的源目的节点id编号和项目的交换机连接端口号等。

[0048] (4) 用户根据步骤(1)收到的网络拓扑和链路代价信息,计算满足自身需求的最优的路径和流量分配。此处计算方法可以根据用户不同的需求进行改变。

[0049] (5) 根据步骤(4)的计算结果,用户通过REST API指令向网络控制器提交路由请求。此处需要使用的REST API指令是PUT指令,如下所示:

[0050] `pusher.set(flow1)`

[0051] 此处flow1代表流表项,流表项由用户选路结果决定。

[0052] (6) 网络控制器Floodlight Controller根据用户的路由请求调用转发模块下发流表;

[0053] (7) 网络交换设备根据流表转发用户数据。

[0054] 上述实施例为本发明较佳的实施方式,但本发明的实施方式并不受上述实施例的限制,其他的任何未背离本发明的精神实质与原理下所作的改变、修饰、替代、组合、简化,均应为等效的置换方式,都包含在本发明的保护范围之内。

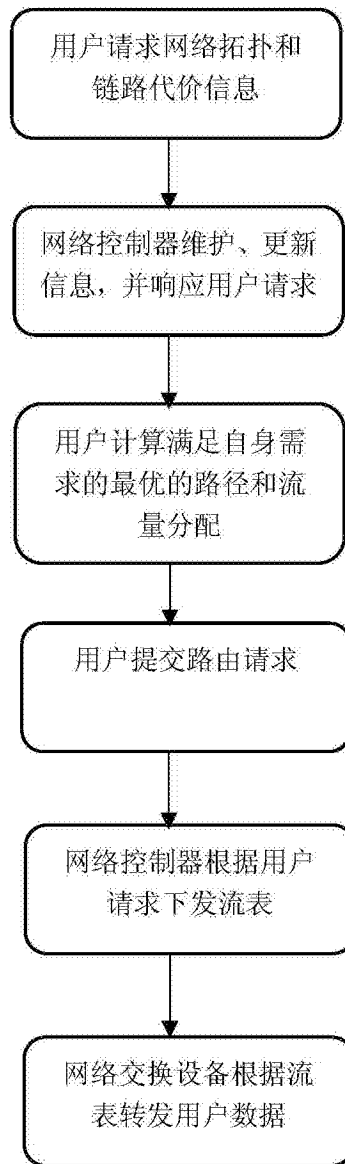


图1

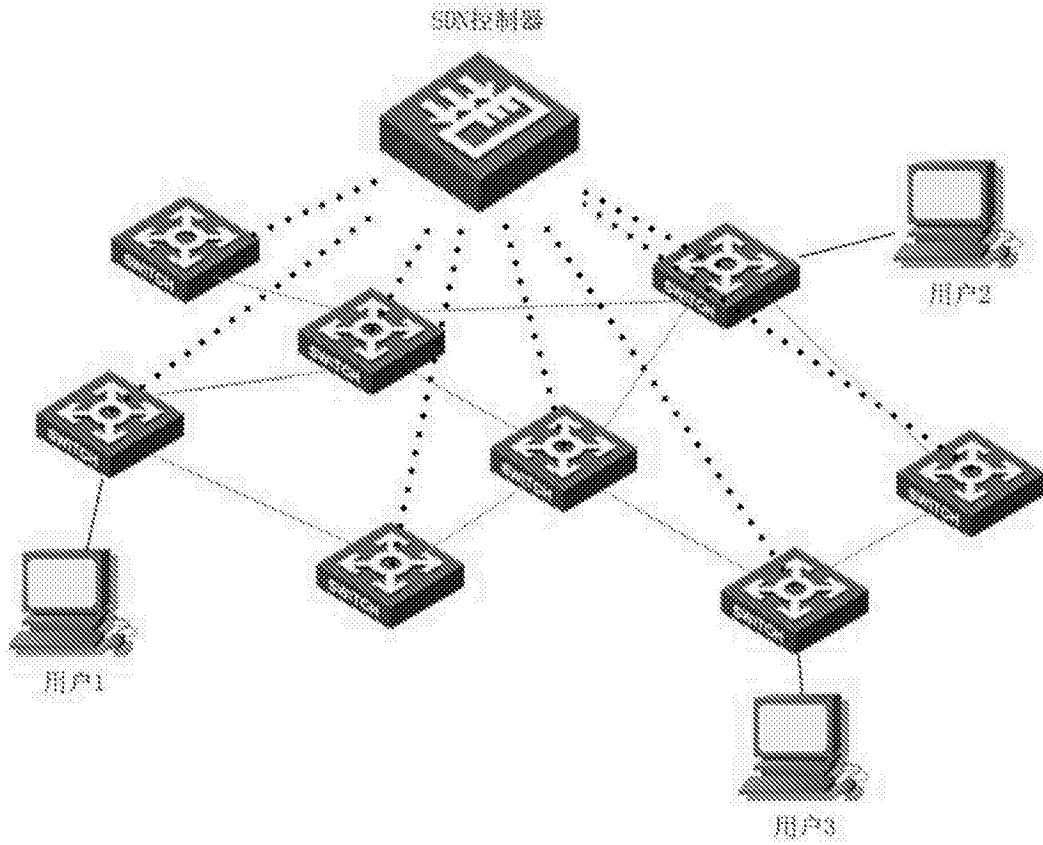


图2

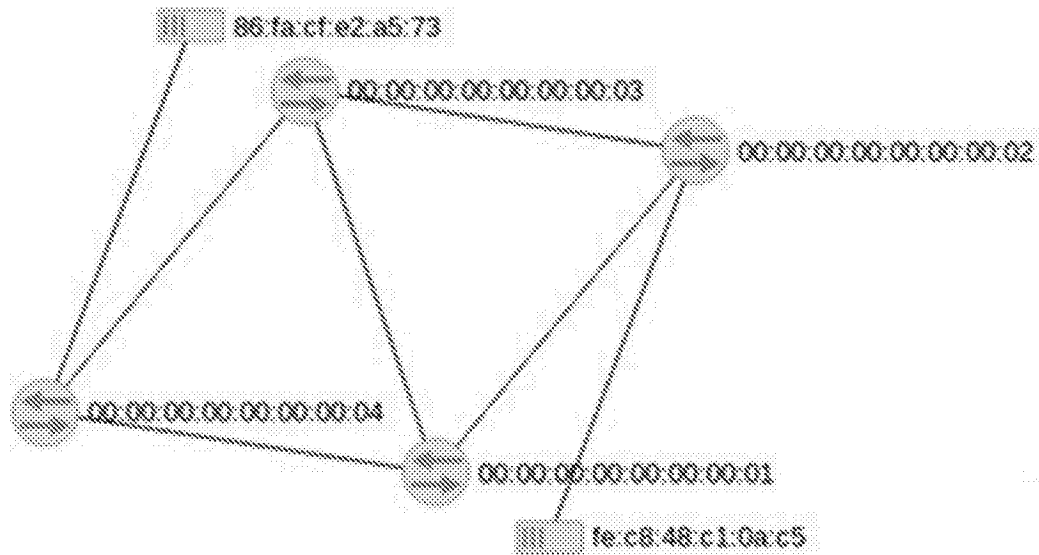


图3