

# 基于SDN的服务驱动组播的设计与实现

陈红,刘宁春,郜帅,周华春  
(北京交通大学 电子信息工程学院,北京 100044)

**摘要:**为了使组播路由与组播用户的服务需求相匹配,提升用户的组播使用体验,结合软件定义网络(sdn)独有的路由与转发分离特点,设计并实现了基于SDN的服务驱动组播路由方案。该方案通过对Dijkstra算法进行改进,将组播用户的服务需求加入到组播路由的计算过程中,控制器可以根据用户需求实时更新组播路由并下发符合需求的新流表到交换机,实现根据用户需求动态调整组播路由的基于SDN的服务驱动组播路由机制。该机制拥有完整的组播流程,集组播建立、组播接收者加入/离开、组播管理、根据用户需求动态调整组播路由等功能为一体,创新性地根据用户的不同服务需求,随时变更并下发符合用户需求的组播路径。使用Mininet平台等仿真工具对提出的方案进行了功能测试和性能测试,结果表明,基于SDN的服务驱动组播在满足用户服务需求的情况下,不会对原有组播性能造成较大影响。

**关键词:**组播路由;服务驱动;软件定义网络;受控组播;组播优化

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2020)12-0130-06

doi:10.3969/j.issn.1673-629X.2020.12.023

## Design and Implementation of Service Driven Multicast for Software Defined Network

CHEN Hong,LIU Ning-chun,GAO Shuai,ZHOU Hua-chun

(School of Electronic and Information Engineering,Beijing Jiaotong University,Beijing 100044,China)

**Abstract:**In order to match the multicast routing with the service requirements of multicast users and improve the user's multicast experience,a service driven multicast routing scheme based on SDN is designed and implemented in combination with the unique separation of routing and forwarding features of SDN. In this scheme,the service requirements of multicast users are added to the calculation process of multicast routing by improving Dijkstra algorithm. The controller can update the multicast routing in real time according to the needs of users and send the new flow table that meets the needs to the switch,thus realize the SDN based service driven multicast routing mechanism that dynamically adjusts the multicast routing according to the needs of users. The mechanism has a complete multicast process,which integrates the functions of multicast establishment,multicast receiver joining / leaving,multicast management,dynamic adjustment of multicast routing according to the user's needs,innovatively changes and issues the multicast path that meets the user's needs at any time according to the user's different service needs. With Mininet platform and other simulation tools to test the function and performance of the proposed scheme,the performance test results show that the SDN based service driven multicast will not have a significant impact on the performance of the original multicast when it meets the service needs of users.

**Key words:**multicast routing;service driven;software defined network;controlled multicast;multicast optimization

## 0 引言

传统IP组播由两个部分组成,组管理协议IGMP运行在接入路由器和组播接收者之间,主要功能是提供转发组播数据包到目的地的最后阶段所需的信息,实现主机与路由器之间的组播组信息交互;组播路由协议运行在组播路由器之间,用于在路由器之间共享

组信息,为组播数据报的分发提供路由,确保每一个组播路由器都能收到数据包<sup>[1]</sup>。传统互联网中的路由计算和路由转发功能都由路由器来完成,组播的实现需要客户端、边缘路由器和经过的所有路由器的共同参与,组播的建立和维护需要复杂的组播路由算法和组播组管理机制<sup>[2]</sup>。

收稿日期:2020-01-22

修回日期:2020-05-22

基金项目:国家重点研发计划(2019YFB1802503);装备预先研究项目(JZX2017-1000/Y312-01)

作者简介:陈红(1995-),女,硕士研究生,研究方向为下一代互联网、SDN等;郜帅,博士,教授,研究方向为未来网络架构、移动互联网、无线传感网络等;周华春,博士,教授,研究方向为未来网络架构、移动互联网等。

基于 SDN<sup>[3-4]</sup> 的组播路由借助 SDN 设计思想,将网络的路由计算与转发工作解耦,由控制器进行组播路由的计算和分发,交换机根据控制器分发的流表进行转发<sup>[5]</sup>。随着 SDN 的发展成熟,一些利用 SDN 提升组播性能的组播路由算法也相继出现<sup>[6-7]</sup>,由 Marcondes 等人提出的 CastFlow 组播路由算法<sup>[8]</sup>结合 SDN 网络架构,由控制器集中管控组播接收者变动和组播树的生成,节省组播响应时间;由 Ananta 等人提出的 EDSPA (extended Dijkstra's shortest path algorithm)<sup>[9]</sup>将节点处理信息的时延与链路状态结合,生成新的链路权重,用于计算从组播源到组播接收者间的最短路径树;由 Lin 等人提出的 LAMA (locality-aware multicast approach)<sup>[10]</sup>通过将组播组集成组播集群的方式,减少组播路由的计算时间和交换机上的路由条目,提高组播的可扩展性<sup>[11]</sup>。然而,上述基于 SDN 的组播路由机制均为被动地收集组播链路的状态信息,并基于此生成组播路由策略<sup>[12]</sup>并没有考虑特定组播服务的需求,如组播源希望其发送的组播数据包不经过某些特定的交换机;组播业务对链路带宽敏感等。

因此,该文提出了一种基于 SDN 的服务驱动组播机制,该机制可根据特定组播服务的需求(不经过特定交换机以及链路带宽敏感)动态调整网络中运行的组播策略,实现 SDN 组播的可管可控。

## 1 服务驱动组播路由

### 1.1 服务驱动组播路由算法

在组播实现的过程中,组播用户会有定制化服务的需求,例如绕过某个工作性能不是很好的交换机;选择一些高带宽低时延的链路等。该文所提出的组播路由实现了这样一种功能,为每个组播路由都加了一个条件限制参数,用来存储组播用户对于该组播组的服务需求。当组播建立完成后,用户可以在任何时间提出自己的要求,通过控制器的命令行输入具体指令,系统会将读取到的服务需求进行处理,重新规划组播的路径。具体的实现流程如图 1 所示。

服务驱动组播路由算法建立在特定源组播<sup>[13]</sup>的基础上,用户必须指定要添加服务需求的组播组的源地址和组播地址;目前可供选择的服务类型包括“障碍”需求和带宽需求,分别代表不允许经过某个交换机、限制链路带宽;条件参数则根据服务类型补充具体的条件。

组播应用接收到用户的服务请求后,判断该组播路由是否存在,若存在,将用户输入的过滤条件类型和条件参数加入到该组播路由中,根据服务驱动组播路由算法计算路径并下发到交换机。

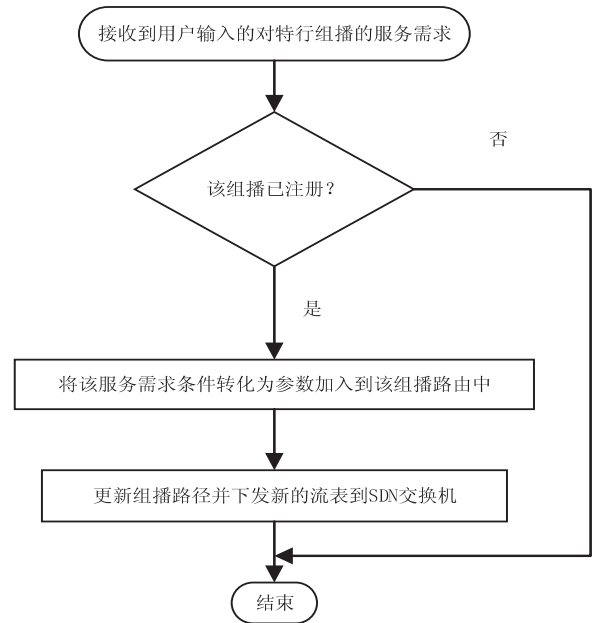


图 1 服务需求处理流程

服务驱动组播路由算法在计算路径时,首先需要指定起点  $s$  (即从顶点  $s$  开始计算)。此外,引进两个集合  $S$  和  $U$ 。 $S$  记录已求出最短路径的顶点(以及相应的最短路径长度),而  $U$  记录还未求出最短路径的顶点(以及该顶点到起点  $s$  的距离)。服务驱动组播路由算法求解组播路径的完整步骤为:(1)初始态,  $S$  只包含起点  $s$ ;  $U$  包含除  $s$  外的其他顶点,且  $U$  中顶点的距离为“起点  $s$  到该顶点的距离”,不与  $s$  直接相连的点的距离记为无穷大。(2)从  $U$  中选出“与起点  $s$  距离最短的点  $k$ ”,并将顶点  $k$  加入到  $S$  中;同时,从  $U$  中移除顶点  $k$ 。(3)更新  $U$  中剩余点与  $s$  的距离,更新方法为在满足组播用户的服务需求条件下,比较当前该点记录的到  $s$  的距离和该点经过  $k$  到  $s$  的距离大小,取其中的较小值更新为该点与  $s$  的距离,如算法 1 所示。(4)重复步骤(2)和(3),直到遍历完所有顶点。

算法 1:更新节点  $s$  与节点  $x$  之间的最短距离。

$s$  /\* 源节点  $s$  \*/

$k$  /\* 中间节点  $k$  \*/

$x$  /\* 待更新节点  $x$  \*/

$D(s, k)$  /\*  $s$  与  $k$  之间的最短距离 \*/

$D(k, x)$  /\*  $k$  与  $x$  之间的最短距离 \*/

$D(s, x)$  /\*  $s$  与  $x$  之间的最短距离 \*/

path /\* 从某个点到另一个点的路径,以数组表示,如:  $a$  经过  $b$  到  $c$  的路径记为  $[a, b, c]$  \*/

$F(\text{path})$  /\* 判断某条路径是否符合预设条件,若符合预设条件,结果为 true \*/

/\*  $s$  与  $x$  经过中间节点  $k$  的距离小于原本记录的距离,并且路径  $[s, k, x]$  符合预设条件,则更新  $s$  与  $x$  之间的最短距离,否则不更新 \*/

if ((  $D(s, k) + D(k, x) < D(s, x)$  ) and  $F([s, k, x]) = \text{true}$  )

then  $D(s, x) \leftarrow D(s, k) + D(k, x)$

算法 1 中利用  $F$  函数来判断当前路径是否符合服务需求,不同的服务需求对应不同的  $F$  函数。根据用户提出的服务需求不同,可以大致分为指定服务方式的需求和指定服务质量的需求两类。下面分别以具体场景举例说明这两种需求的服务场景和对应的  $F$  函数的设计。

## 1.2 指定服务方式的需求举例

### 1.2.1 服务场景

使用最短路径算法生成的组播路径只会考虑组播源到组播接收者之间的距离最短的路径,然而在一些情况下,网络中的 SDN 交换机并不是完全可信的,组播用户希望组播算法生成的组播路径可以绕开不可信的 SDN 交换机,保证组播消息在可靠的 SDN 交换机之间传输,这也就是该文所提出的“障碍”需求服务场景,用户以控制器命令行的方式指明特定组播路由不允许经过的交换机,服务驱动组播路由算法根据用户需求重新生成组播路由并下发表。

### 1.2.2 判断函数

该需求服务场景的判断函数的入参为待判断路径  $[s, k, x]$ ,其中  $s$  为组播源,  $x$  为其中一个接收者,要判断的是,如果组播源  $s$  到接收者  $x$  之间的路径经过  $k$ ,是否符合用户提出的需求。

判断方法:读取用户提出的关于“障碍”需求的参数,若参数中包含交换机节点  $k$ ,则返回 false;否则返回 true。

## 1.3 指定服务质量的需求举例

### 1.3.1 服务场景

由最短路径算法生成的组播路径不会对带宽有明确的规定,也无法预测用户将要传送的组播消息占用带宽的大小。当用户明确要传送的组播消息占用带宽很大(如视频文件),在低带宽的链路中容易造成较大的丢包和较长的时延,这时用户可以在控制器命令行中明确提出使用的组播路径的最小带宽,服务驱动组播路由算法根据用户的需求筛选合适的路径,重新下发表。

### 1.3.2 判断函数

该需求服务场景的判断函数的入参为待判断路径  $[s, k, x]$ ,其中  $s$  为组播源,  $x$  为其中一个接收者,要判断的是,如果组播源  $s$  到接收者  $x$  之间的路径经过  $k$ ,是否符合用户提出的需求。

判断方法:读取用户提出的带宽需求;从控制器中取出节点  $s$  到节点  $k$  之间的链路的带宽,以及节点  $k$  到节点  $x$  之间的链路带宽;当  $s$  与  $k$  之间的带宽以及  $k$  与  $x$  之间的带宽均大于用户的带宽需求时认为这是一条满足用户需求的路径,返回 true;否则返回 false。

## 2 仿真测试

### 2.1 仿真环境

仿真是在由 Mininet<sup>[14]</sup> 搭建的 SDN 网络上<sup>[15]</sup> 进行的。Mininet 是一个轻量级软件定义网络和测试平台;它采用轻量级的虚拟化技术使一个单一的系统看起来像一个完整的网络运行相关的内核系统和用户代码,也可简单理解为 SDN 网络系统中的一种基于进程虚拟化平台,它支持 OpenFlow 等协议。在仿真环境中,使用一个控制器来控制网络内的交换机和主机,网络的基本配置信息如表 1 所示。

表 1 仿真环境配置

参数	设置
控制器个数	1
组播源数目	1
控制器配置	ONOS1.5.0
OpenFlow 交换机	OpenvSwitch 1.3
测试工具	Iperf
测试数据速率	100 Mbps
链路带宽配置	s3 与 s8 之间的链路带宽设置为 100 M, 其他链路带宽为 1 000 M
每个用例测试时长	30 s

### 2.2 基于 SDN 的服务驱动组播机制构建

#### 2.2.1 组播流程

基于 SDN 的服务驱动组播将组播接收者管理工作集中在 SDN 控制器内,支持多个组播同时独立进行组播业务,支持组播接收者在任意时间的加入或退出,同时支持用户在组播运行过程中,随时根据服务需要,为组播组添加路由过滤条件,控制器根据用户添加的条件及时更新组播路由。

组播的工作流程如下:

(1) 接收到需要处理的组播数据包(符合目的 IP 地址为组播地址的 IPv4 数据包)。

(2) 对组播数据包进行筛选,避免浪费应用资源进行不必要的处理工作。在这一过程中,去除掉一些不符合条件的数据包,如已经被别的应用处理过的、内容为空的以及 IPv6 数据包等。

(3) 解析数据包,得到数据包的源地址、目的地址(组播地址)、使用的网络层协议(用来判断该数据包是组播源发出的组播数据包还是由组播接收者发出的加入/离开组播请求)。

(4) 如果解析到的数据包协议为 IGMP 协议,则执行组播接收者加入/退出流程;否则认为该数据包是由组播源发送的,执行组播建立过程。



### 2.2.2 组播建立

组播应用接收到来自组播源的数据包,首先解析数据包得到该数据包的源地址与目的地址,数据包的源地址即为组播源的 IP 地址,数据包的目的地址即为该组播组地址,根据源地址与组播地址,查找是否含有相同组播地址和源地址的组播组存在,如果有,则认为该组播组已经注册过,不再对该数据包进行后续处理;如果没有,则将新的组播路由添加到组播管理应用中,并且根据数据包的源地址信息将该条组播路由条目的与组播源相连的交换机端口信息补充完整,然后将组播地址和源地址的对应关系添加到源和组播地址对应表中。

至此,一条新的组播路由建立完成,它包含组播地址、源地址和组播源相连的交换机端口,由于当前组播还没有成员加入,控制器不会生成组播路由并下发路径到交换机。当有组播接收者加入的时候,控制器才能获得该组播组的接受者所连的交换机端口集合信息,并将组播路由转化为相应的组播路径下发到交换机,在没有组播用户提出的特殊服务需求时,组播路由采用 Dijkstra 算法<sup>[16]</sup>;在有服务需求时,采用服务驱动组播路由算法。成员加入的流程具体见 2.2.3。

### 2.2.3 组播接收者加入/离开

组播应用接收到来自组播接收者的数据包 (IGMP 数据包),首先解析数据包得到该数据包的有效载荷,根据有效载荷中提供的信息判断该数据包是加入组播组请求还是离开组播组请求,分别对应着不同的处理。

以加入组播组的请求为例,根据成员想要加入的组播组地址,查询组播源与组播地址对应表中是否有对应的组播源地址,如果没有,则表示该成员想要加入的组播组尚且没有在组播应用中建立,无需处理入组请求。如果有,则表明该组播组已经在组播应用中注册,此时要根据要加入的组播组地址和查找出的对应的组播源地址,获取该组播的成员列表,如果成员列表中不包含该成员,那么就将该成员加入,更新组播路由。

对离开组播组请求的处理与加入组播组请求处理过程大致相似,不同的是一个是将该成员加入到组播的成员列表中,另一个是从成员列表中移出该成员。

## 2.3 服务驱动路由功能测试

在组播用户没有提出特定服务需求的情况下,组播应用会按照 Dijkstra 算法生成从组播源到各个组播接收者之间的最短路径。当用户提出特定的服务需求时,组播路由会根据用户的需求决定是否更新。如图 2 所示的测试拓扑,由 11 个交换机和 16 个主机构成。组播源为 h1,四个组播接收者分别为 h5, h7,

h8, h9。

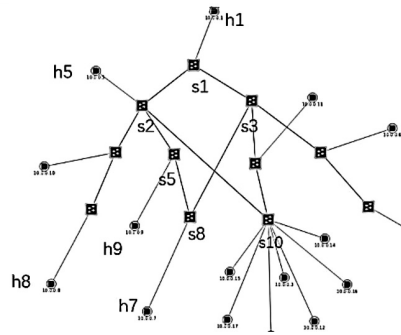


图 2 服务驱动路由功能测试

### 2.3.1 特定交换机需求测试

测试当用户添加不允许任何组播路径经过 s3 交换机的服务需求时,从 h1 到 h7 的路径是否会发生变化。

其中 h1 到 h7 有两条路径可以选择,分别是 [s1, s3, s8] 和 [s1, s2, s5, s8]。组播用户没有对组播路径提出具体需求,组播应用按照 Dijkstra 算法生成从组播源到各个组播接收者间的最短路径,可以看到此时 h1 到 h7 之间的路径选择的是 [s1, s3, s8], 如图 3 所示。

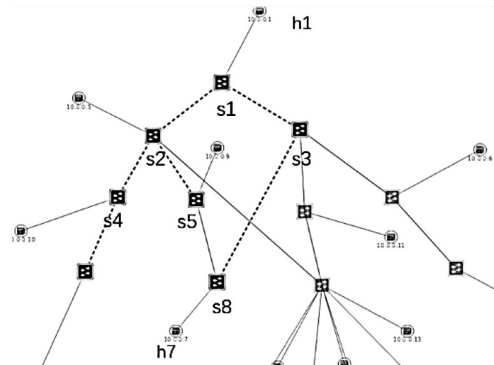


图 3 没有用户指定服务需求时的组播路由

之后,组播用户为该条组播添加特定的服务需求,需求为:所有组播路径不允许经过 s3,此时的组播路由变化如图 4 所示,可以看到此时从 h1 到 h7 的路径选择了 [s1, s2, s5, s8]。

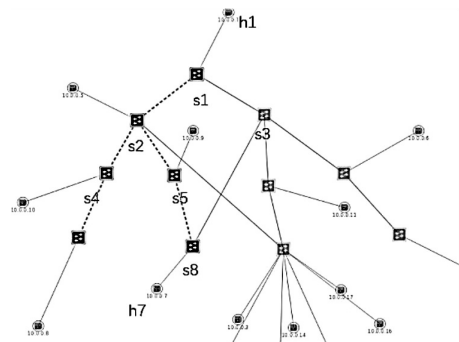


图 4 添加用户服务需求的路由

### 2.3.2 带宽需求测试

根据链路带宽设置,重新开启一个组播地址为

224.1.1.5 的组播路由,建立由 h1 到 h7 和 h8 的组播路由,由于 s3 与 s8 之间的链路带宽设置为 100 M,测试当加上所选路径带宽必须大于 300 M 的时候,h1 到 h7 之间的路径是否会发生变化。

在该组播路由没有带宽需求的时候,h1 到 h7 之间的路径如图 3 所示,为根据最短路径算法得到的最短路径。之后,组播用户为该条组播添加特定的服务需求,需求为:要求所有组播路径的带宽必须大于 300 M,添加带宽服务需求后的组播路径如图 5 所示。

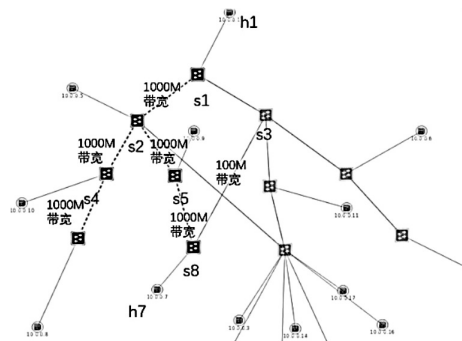


图 5 添加带宽需求后的组播路径

在添加需求后,h1 到 h7 之间选择了 [s1, s2, s5, s8],绕开了 [s3, s8] 这个不符合服务需求的路径。

## 2.4 服务驱动组播性能测试

### 2.4.1 端到端时延测试

消息从组播源发出到组播接收者成功接收的时间记为端到端时延,是组播用户非常敏感的一个参数,测试该文所提出的服务驱动组播在没有服务需求和有服务需求两种情况下,端到端时延的变化。端到端时延的测量方法为:在组播源发出的数据包中打上时间戳,接收者接收到数据包解析时间戳,并用接收时间减去时间戳时间,得到该组播源到该接收者之间的端到端时延。时延测试结果如图 6 所示。

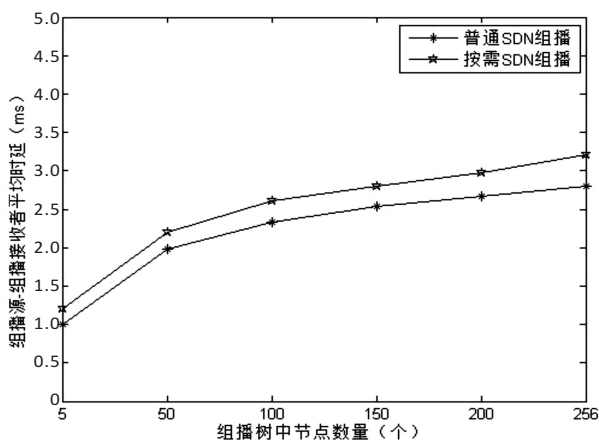


图 6 平均端到端时延的测量结果

根据时延测量结果,当添加了服务需求后,由于服务需求对流表的影响,使端到端时延产生了一定的增加,但是增加的范围很小,不会对时延性能造成太大

影响。

### 2.4.2 路由收敛速度测试

组播接收者的加入/退出、链路状态改变、用户添加服务需求等都会对组播策略产生影响,造成组播路径的重新规划和流表的重新下发等行为。该文测试了由于组播接收者加入/退出造成的路由重新规划,从成员发送 IGMP 包开始到新的策略下发到交换机的整个过程的时间。策略下发时间包含成员发送 IGMP 包到控制器时间 T1,控制器处理 IGMP 包并生成新的组播路由时间 T2,控制器将组播路由下发到交换机的时间 T3。由于控制器与交换机之间直接相连,T3 相比 T1 和 T2 来说很小,策略下发时间  $T \approx T1 + T2$ 。策略下发时间的测试结果如图 7 所示。

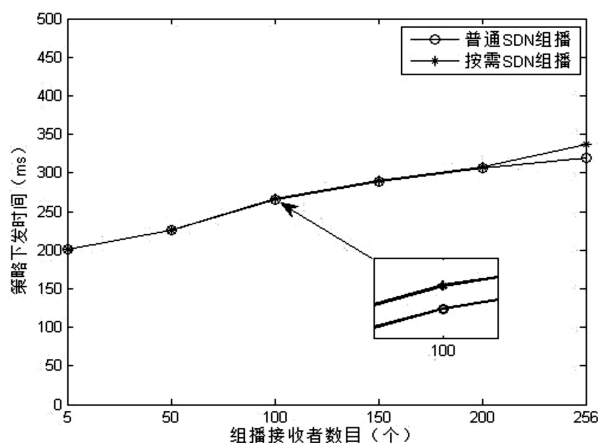


图 7 策略下发时间测量结果

根据测量结果显示,基于 SDN 的服务驱动组播的策略下发时间在几百毫秒之间,随着组播规模的增加,策略下发时间也会增加;添加服务需求后的组播路由的策略下发时间要比没有添加服务需求的下发时间长,会比没有服务需求的组播路由增加约 1%。

## 2.5 测试结论

经过本章的功能测试和性能测试,得出以下结论:

(1) 所提出的基于 SDN 服务驱动组播可以在 SDN 网络条件下,服务于有基础组播需求的用户和有特殊服务需求的组播用户;(2) 没有服务需求时,组播路由默认采用最短路径算法,为组播路由添加服务需求后,根据需求对最终生成的组播路径的影响程度,会轻微牺牲组播的性能表现,但是不会对性能产生严重影响。

## 3 结束语

从组播用户的角度提出了基于 SDN 的服务驱动组播的概念,设计并实现可以根据用户需求动态更改组播路由的组播算法,一方面可以实现用户对于组播的不同的定制化需求,另一方面依靠 SDN 路由与转发分离的特性,将传统互联网中的组播路由计算和根据

路由转发两个功能解耦出来,极大地减少了交换机负担,提高组播的服务质量。

目前的用户需求仅支持用户手动选择或者添加相应的服务特定需求。但是,组播用户对于组播路由的需求和业务场景是有密切联系的,是有规律可循的。可以利用组播用户大数据,探索用户需求与业务场景之间的关系,找出其中的联系规律,将部分需求封装在组播路由内部,由组播系统自动帮助用户选择。例如,用户传送视频等需要大带宽的文件时,自动为用户选择链路占用率小的链路,保证组播文件的有效传送。

在大数据与人工智能发展如此快速的今天,基于SDN的组播路由可以借助其力量,为组播用户提供更加便捷优质的服务。

#### 参考文献:

- [1] LATIF A, KATHAIL P, VISHWARUPE S, et al. Multicast optimization for CLOS fabric in media data centers[J]. IEEE Transactions on Network and Service Management, 2019, 16(4):1855-1868.
- [2] ZHANG Xinchang, YANG Meihong, ZHAO Yanling, et al. An SDN-based video multicast orchestration scheme for 5G ultra-dense networks[J]. IEEE Communications Magazine, 2017, 55(12):77-83.
- [3] NUNES B A A, MENDONCA M, NGUYEN X N, et al. A survey of software-defined networking: past, present, and future of programmable networks[J]. IEEE Communications Surveys & Tutorials, 2014, 16(3):1617-1634.
- [4] 张朝昆, 崔勇, 唐嵩祎, 等. 软件定义网络(SDN)研究进展[J]. 软件学报, 2015, 26(1):62-81.
- [5] RAWAT D B, REDDY S R. Software defined networking architecture, security and energy efficiency: a survey[J]. IEEE Communications Surveys & Tutorials, 2016, 19(1):325-346.
- [6] ISLAM S, MUSLIM N, ATWOOD J W. A survey on multicasting in software-defined networking[J]. IEEE Communications Surveys & Tutorials, 2017, 20(1):355-387.
- [7] ALSAEED Z, AHMAD I, HUSSAIN I. Multicasting in software defined networks: a comprehensive survey[J]. Journal of Network and Computer Applications, 2018, 104:61-77.
- [8] MARCONDES C A C, SANTOS T P C, GODOY A P, et al. CastFlow: clean-slate multicast approach using in-advance path processing in programmable networks[C]//2012 IEEE symposium on computers and communications (ISCC). Capadocia: IEEE, 2012:94-101.
- [9] ANANTA M T, JIANG J R, MUSLIM M A. Multicasting with the extended Dijkstra's shortest path algorithm for software defined networking[J]. International Journal of Applied Engineering Research, 2014, 9(23):21017-21030.
- [10] LIN Y D, LAI Y C, TENG H Y, et al. Scalable multicasting with multiple shared trees in software defined networking[J]. Journal of Network and Computer Applications, 2017, 78:125-133.
- [11] DALDOUL Y, MEDDOUR D E, AHMED T, et al. Performance and scalability evaluation of IEEE 802.11 v/aa multicast transport[J]. Wireless Communications and Mobile Computing, 2016, 16(14):1987-2000.
- [12] HUANG L, ZHI X, GAO Q, et al. Design and implementation of multicast routing system over sdn and sflow[C]//2016 8th IEEE international conference on communication software and networks (ICCSN). Beijing: IEEE, 2016:524-529.
- [13] KIM H, YUN S, KIM H, et al. A novel SDN multicast for large-scale IoT environments[C]//2017 international conference on information and communication technology convergence (ICTC). Jeju Island: IEEE, 2017:823-828.
- [14] MISIC M J, GAJIN S R. Simulation of software defined networks in Mininet environment[C]//Telecommunications forum telfor. Belgrade: IEEE, 2014:1055-1058.
- [15] 邹剑锋. 基于OpenFlow的SDN组网技术研究[D]. 北京: 北京邮电大学, 2014.
- [16] BROUMI S, BAKAL A, TALEA M, et al. Applying Dijkstra algorithm for solving neutrosophic shortest path problem[C]//2016 international conference on advanced mechatronic systems (ICAMEchS). Melbourne: IEEE, 2016:412-416.