# [基于 SDN 与机器学习的业务流 Qos 保障系统]

# 设计文档

所在赛道与赛项：B-EP1

## 一、 目标问题与意义价值

随着网络技术的快速发展，网络规模的增大和应用数量的逐渐增加，使得用户对网络服务质量的保障提出了新的要求，需要高效的路由算法来保障业务流的服务质量(QoS)需求。但是传统的网络架构过于复杂，难以获取全局视图，受到了路由算法的设计和应用的限制，无法提供理想的 QoS 服务。

软件定义网络(Software Defined Network, SDN)架构的提出解决了控制平面和数据平面的耦合问题，为 QoS 路由提供了新的思路。使用 SDN 架构，可以根据 QoS 策略在控制平面实现对应的路由算法，并通过 OpenFlow 协议在数据平面上安装相应的流表。同时，机器学习算法的广泛应用也为 SDN 网络中的 QoS 路由优化带来了新的研究方向。

我们提出“基于 SDN 和机器学习的业务流 Qos 保障系统”，其核心是在 SDN 网络基础上利用机器学习实现流量预测和分类，进而实现 Qos 路由和差异化限速。该系统可以检测和排除异常流量，提供先见性的流量预测，并实现对路由的自动调整，从而为网络中的业务流传输提供保障，同时提供更加理想的 Qos 服务。

本系统结合了 SDN 和机器学习技术，将二者有机地结合，从而实现了对网络中业务流的精细化控制，为网络中的各种应用提供了更加理想的 QoS 服务。其次，该系统利用机器学习技术实现了对异常流量的检测和排除，提高了网络的可靠性和稳定性，同时提升了网络的安全性能。最后，该系统实现了对路由的自动调整，可以根据流量预测和分类结果对网络路由进行动态调整，从而进一步提高网络的质量和性能，为用户提供更加优质的服务。

## 二、 设计思路与方案

### 2.1 设计思路

基于 SDN 和机器学习的业务流 Qos 保障系统是基于软件定义网络（SDN）和机器学习技术的一种业务流质量保障系统。该系统可以对网络拓扑、网络流量、链路状态信息进行采集和存储，实现流量预测和流量分类，并利用预测数据计算未来路由，以实现路由自动调整。同时，该系统还可以对异常流量进行检测和排除，并实现差异化限速和交互友好的前台系统，方便用户进行数据可视化和下发业务策略

## 2.2 系统架构

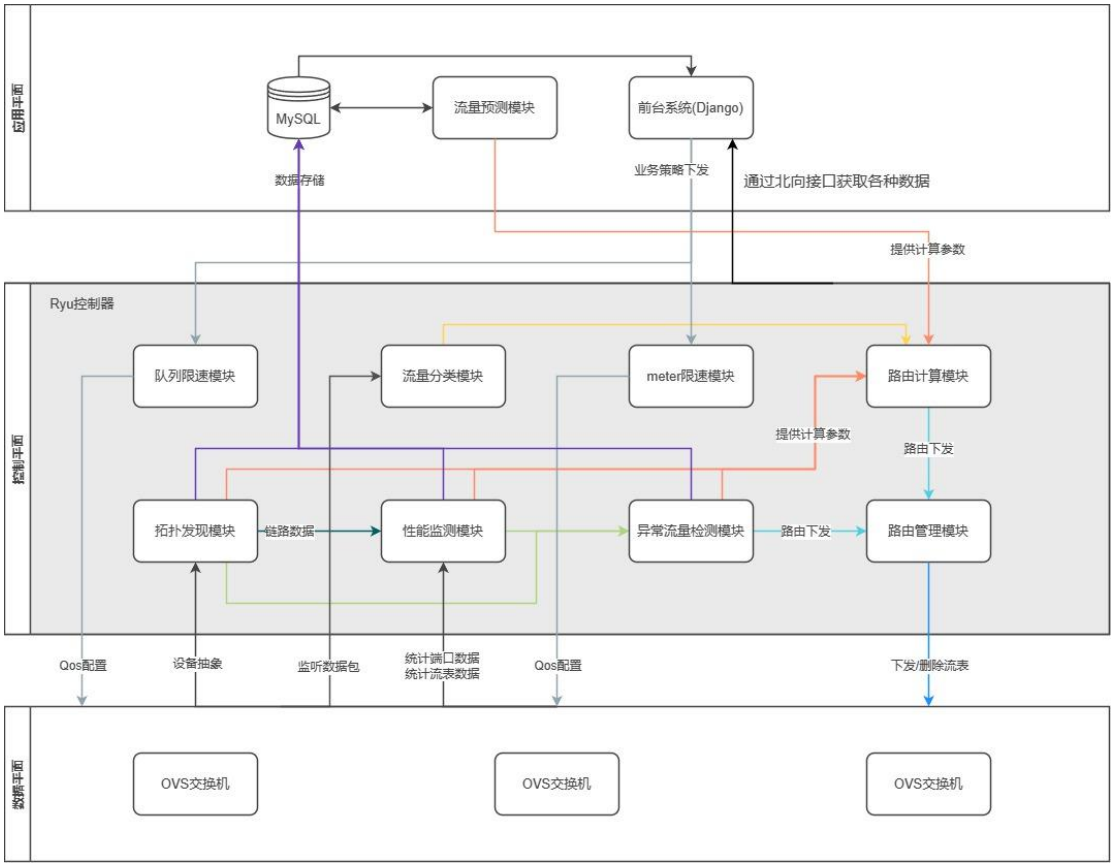



**图 1 系统架构图**

拓扑发现模块和性能监测模块随系统启动而启动，并每隔 30 秒采集一次数据。拓扑发现模块将链路信息输出至性能监测模块得到每条链路的状态信息，如吞吐量、时延、抖动、丢包率等。随后拓扑发现模块和性能监测模块将数据存储至数据库中，同时输出至异常流量监测模块。

当异常流量监测模块识别到 DDOS 攻击后将追溯流量路径，并阻断异常流量。

流量预测模块定时从数据库中获取采集到的链路信息进行预测，并将结果保存至数据库中。

当未知请求发起时会将数据包通过 Packet-In 上传至控制器进行分析，并将分类结果反馈给路由计算模块，路由计算模块通过多种计算参数计算路由并下发给路由管理模块，由路由管理模块进行流表的下发和删除。

路由下发后持续对路径进行预测和监测，以实现路由的自动调整。

前台系统可以从数据库和 Ryu 北向接口得到详细的拓扑信息、各类流量信息。并且可以进行流表管理和业务策略下发。业务策略下发给队列限速模块和 meter 限速模块来实现差异化服务。

## 2.3 系统流程

路由管理模块是该 Qos 路由方案的核心模块，在本系统中，控制器会采用一种新的 Qos 路由算法来执行业务的路由计算。

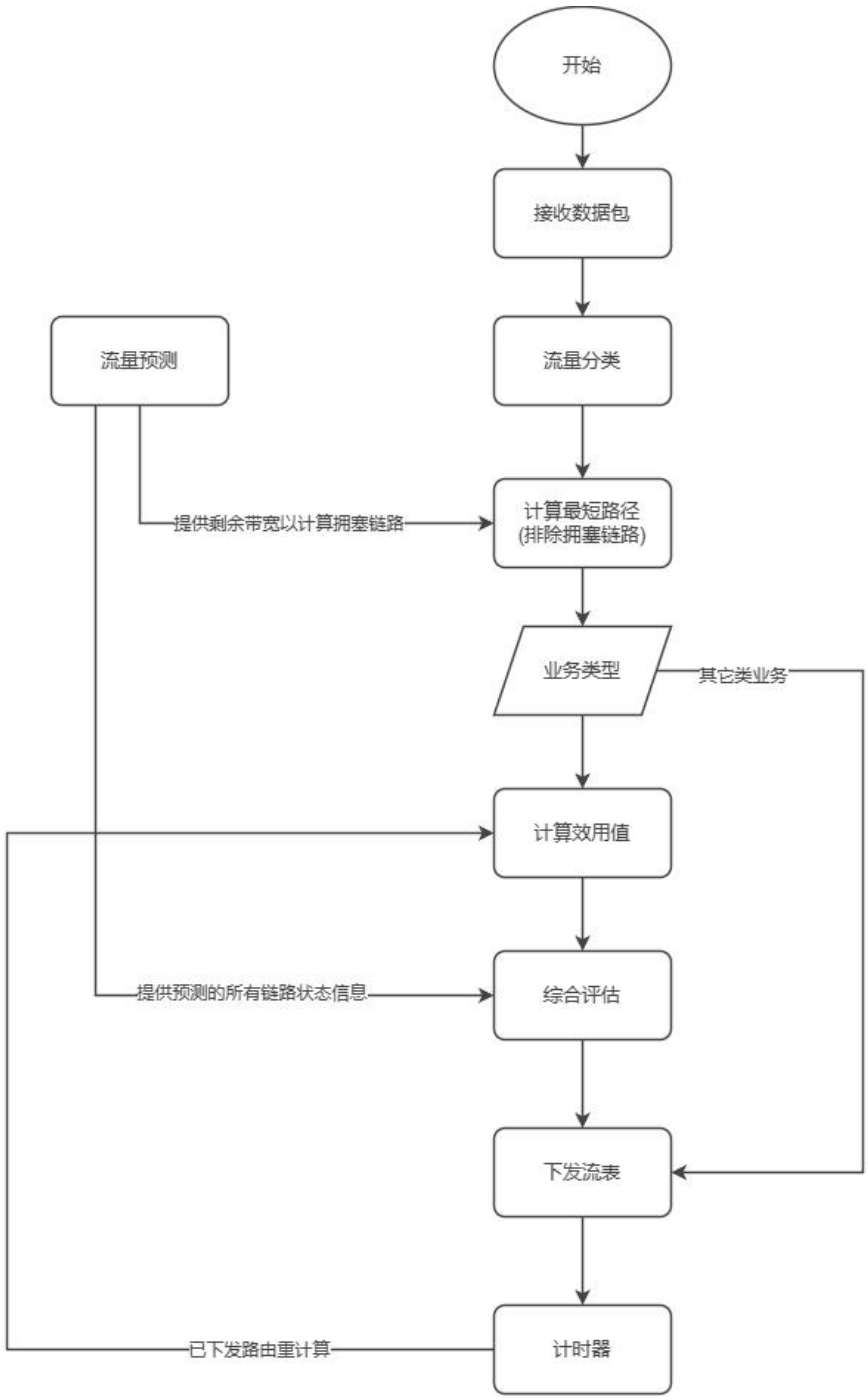



**图 2 路由算法**

该路由算法首先对收到的数据包进行识别，然后计算出源到目的的 K 条最短路径（K-Shortest Path，KSP）作为备选路径。若是其它类业务则直接下发最短路径路由，其余的进行 Qos 路由计算。Qos 路由结合当前与预测的网络状态和不同类型业务流的 QoS 需求对每条备选路径的效用值进行计算，作为路径度量。然后，以效用值最大的路径作为基准，选取一组效用值在容限范围内的优选路径进行下发。在流表下发后对当前路由信息

进行记录，并定时重新计算最优路径和自动调整。

## 2.4 业务流量设计

流量分类是对未知的数据流进行识别、分类，故设计业务流量是分类的前提。

网络传输中数据流的要求通常可以用四种参数来说明，分别是带宽、时延、时延抖动和丢包率。不同业务类型对 QoS 有不同的需求。而随着业务规模的不断扩大，对业务流的合理分类变成了提高网络资源利用率、满足业务 QoS 需求的主要任务。

本系统将业务分为会话类、流媒体类、交互类、下载类、其它类。

（1）会话类业务

实时的双向通信业务，即会话类型业务，有着实时且较低的端到端时延、上下行业务量的基本对称特征。在此类业务中，QoS 中的时延具有首要的关注指标，其次则是对于时延抖动方面的需求，这是由于时延抖动过大也会对用户的会话产生不良影响。此外，由于耳朵的感受上限，语音通话中可接受在一定程度内的信号丢失，所以在会话类型业务中丢包率的要求也相对宽松。该类业务的带宽需求不大，但是在考虑通话质量的同时，其优先级往往会被设置为最高。语音电话、视频会议等都是该类业务的代表性应用场景。

（2）流媒体业务

流媒体业务是一种单向实时传输业务。与会话类业务相比，流媒体业务不需要立即与用户进行互动，因此时间延迟的要求可相应放宽。尽管时延抖动仍会对流媒体业务的服务质量产生一定的影响，但具体的抖动容忍度会因用户接收设备的不同而异。由于视觉暂留现象的存在，流媒体业务对一定程度的数据丢失也很容忍。对于流媒体数据的格式、质量不同，对于带宽需求也会有所差异，不过在大多数情况下，流媒体业务所占用的带宽都相对较高。在线视频、在线音乐等都是流媒体业务的典型应用场景。

（3）交互类业务

互动类业务是指终端用户与远程设备进行双向在线数据交互的业务，为了确保业务的准确性，互动类业务首先需要关注的 QoS 指标是丢包率。互动类业务带宽占用相对较低，典型的业务应用包括网络游戏、网页浏览等

（4）下载类业务

下载类业务是一种单向传输业务，涉及数据的向终端用户的传输。与交互类业务和会话类业务不同，下载类业务不需要实时响应，因此时间延迟的要求相对较低。对于这种业务来说，QoS 中的关键指标是传输速率以及数据完整性。由于数据的完整性对于下载类业务至关重要，所以丢包率的要求相对较高。此外，带宽需求也较高，因为下载类业务通常涉及大文件、大数据量的传输。下载软件、电影、音乐等都是该类型业务的常见应用。

（5）其它业务

其它类型的业务通常涉及到一些自动进行的后端任务。背景类业务占用的带宽往往较少，系统将使用最大的传输带宽，但当网络拥塞时，系统将选择性地抛弃该类业务。常见的其它类业务包括电子邮件、短信等应用。

| 业务类型 | 带宽要求 | 时延要求 | 抖动要求 | 丢包率要求 | 优先级 |
|---|---|---|---|---|---|
| 会话类 | 中 | 高 | 高 | 低 | 高 |
| 流媒体类 | 高 | 中 | 中 | 中 | 中 |
| 交互类 | 低 | 中 | 低 | 高 | 中 |
| 下载类 | 高 | 低 | 低 | 高 | 低 |
| 其它类 | 中 | 无 | 无 | 高 | 低 |

**表 1 各业务的 Qos 指标**

业务流量识别依靠 IP 报文头部的 Tos 域。本系统的 Tos 值同等于优先级。

| 业务类型 | Tos | 优先级 |
|---|---|---|
| 会话类 | 16 | 高 |
| 流媒体类 | 12 | 中 |
| 交互类 | 8 | 中 |
| 下载类 | 4 | 低 |
| 其它类 | 0 | 低 |

**表 2 业务 Tos 定义**

## 2.5 效用值设计

本系统将业务 QoS 需求与路径的满足程度量化为路径的效用值，并将该值作为路径度量标准，进而进行路由选择。

网络中的路由问题可以视作图论中的寻路问题，整个网络拓扑记为图 $G=(V,E)$，$V$ 为图 $G$ 的顶点集，代表 SDN 网络中的 OpenFlow 交换机，$E$ 为图 $G$ 的边集，代表 SDN 网络中的链路，每条链路用 e 表示，则 $e \in E$，记链路的带宽为$b_e$、时延为$d_e$、时延抖动为$j_e$、丢包率为$l_e$。一条源到目的的有效转发路径记为 $p$，记路径的带宽为 $b_p$、时延为$d_p$、时延抖动为$j_p$、丢包率为$l_p$，则路径的 QoS 参数计算公式如下：

（1）路径带宽：

$$b_p = \min_{e \in p} b_e$$

（2）路径时延：

$$d_p = \sum_{e=p} d_e$$

（3）路径时延抖动：

$$l_p = 1 - \prod_{e \in p} (1 - l_e)$$

（4）路径丢包率：

$$1_p = 1 - \prod_{e \in p} (1 - l_e)$$

基于计算出的路径的 QoS 参数，该系统使用效用值这一软性指标来评估不同路径对业务 QoS 需求的满足程度。该系统假设各个 QoS 指标相互独立，并通过为不同的 QoS 指标定义不同的效用函数，分别计算效用值，再通过加权求和的方法来计算一条路由的效用值。因此，设一条从源到目的的有效转发路径为 p，该路径的效用值为 U(p)，则该路径的效用值计算公式如下:

$$U_p = \delta_b u_b + \delta_d u_d + \delta_j u_j + \delta_l u_l$$

其中，$u_b, u_d, u_j, u_l$分表代表带宽、时延、时延抖动、丢包率的效用值。

$$\delta_b + \delta_d + \delta_j + \delta_l = 1$$

$\delta_b, \delta_d, \delta_j, \delta_l$分表代表带宽、时延、时延抖动、丢包率的权重，权重之和为 1。单个 QoS 指标的效用函数定义如下:

（1）带宽效用函数:

$$u_b = \frac{100}{1 + \beta e^{-\alpha x + c}} \quad \beta > 0, \alpha > 0$$

其中， $x$代表整条路径的最小剩余带宽。常见的业务可以用阶跃型、指数型和延迟自适应性等函数来表征带宽的效用函数。本系统采取 sigmoid 函数，可以通过调节 α,β两个参数，来呈现上述不同的函数特征。

（2）时延效用函数:

$$u_d = \begin{cases} 100 - \gamma_1 x, x < c_1 \\ b_1 \tanh\left(\beta(x - b_2)\right) + b_3, c_1 \le x \le c_2 \\ \delta - \gamma_2 x, x > c_2 \end{cases}$$

时延的效用函数设计为分段函数， $x$为整条路径的总时延。 $\gamma_1, \beta, \gamma_2$ 用于表征效用值随时延增加下降的陡峭程度。$c_1, c_2$.为上阈值和下阈值。 $b_1, b_2, b_3, \delta$ 是可调参数，用来确保分段函数的连续性。

（3）时延抖动效用函数:

$$u_j = b_1 \tanh\left[\beta(x - b_2)\right] + b_3$$

抖动的效用函数设计为双曲正切函数，因为可以双曲正切函数可以方便地根据时延抖动的上下阈值进行连续函数的建模。 $x$代表整条路径的时延抖动。

（4）丢包率效用函数:

$$u_l = b_1 - b_2 \log\left(b_3 + \beta x\right)$$

其中，函数中$x$代表整条路径的丢包百分比，$\beta$用于表征丢包率增加时效用值下降 的陡峭程度。

对于不同类型的业务，根据业务的实际 QoS 需求设计业务对应的效用函数。

各类型业务的效用参数如下:

| 业务类型 | | 会话类 | 流媒体类 | 交互类 | 下载类 |
|---|---|---|---|---|---|
| 带宽权重 | | 0.15 | 0.5 | 0.2 | 0.6 |
| 函数参数 | α | 0.2 | 0.004 | 0.25 | 0.05 |
| | β | 1 | 1 | 1 | 1 |
| | c | 6 | 14 | 6 | 12 |

**表 3 带宽效用参数**

| 业务类型 | | 会话类 | 流媒体类 | 交互类 | 下载类 |
|---|---|---|---|---|---|
| 时延权重 | | 0.4 | 0.1 | 0.4 | 0 |
| 函数参数 | y1 | 1/15 | 0.004 | / | / |
| | β | 0.04 | 1/200 | 0.015 | / |
| | y2 | 0.016 | 0.006 | / | / |
| | b1 | -25 | -25 | -45 | / |
| | b2 | 205 | 625 | 1.35 | / |
| | b3 | 65 | 75 | 60 | / |
| | c1 | 150 | 200 | / | / |
| | c2 | 300 | 1000 | / | / |

**表 4 时延效用参数**

| 业务类型 | | 会话类 | 流媒体类 | 交互类 | 下载类 |
|---|---|---|---|---|---|
| 抖动权重 | | 0.3 | 0.1 | 0 | 0 |
| 函数参数 | β | 1/15 | 0.005 | / | / |
| | b1 | -30 | -20 | / | / |
| | b2 | 50 | 3 | / | / |
| | b3 | 70 | 80 | / | / |

**表 5 抖动效用参数**

| 业务类型 | | 会话类 | 流媒体类 | 交互类 | 下载类 |
|---|---|---|---|---|---|
| 丢包率权重 | | 0.15 | 0.3 | 0.4 | 0.4 |
| 函数参数 | β | 3 | 12.5 | 1 | 1 |
| | b1 | 882 | 250 | 100 | 80 |
| | b2 | 200 | 50 | 50 | 45 |
| | b3 | 50 | 20 | 1 | 1 |

**表 6 丢包效用参数**

# 三、 方案实现

## 3.1 拓扑发现模块

拓扑发现模块用于定期将网络设备抽象化，然后存储和用于其它模块调用。同时利用全局拓扑视图构造每台交换机间的 k 条最短路径。

在该模块中记录以下内容:

(1) 交换机对:每一对相连的 ovs 的交换机 id，同时记录相连的端口。格式:(src_dpid,dst_dpid)->(src_port,dst_port)

(2) 交换机端口表:每一台 ovs 交换机对应的所有端口。格式:dpip->port_no

(3) 终端连接表:交换机 id 和交换机端口与相连的终端 IP 与 MAC 地址。格式:

(dpid,port) :[host1_ip,host_mac]

(4) 边界端口表:ovs 交换机与终端相连的端口。格式:dpid->port_no

(5) 内部端口表:ovs 交换机与之间相连的端口。格式:dpid->port_no

模块运行过程如下图:

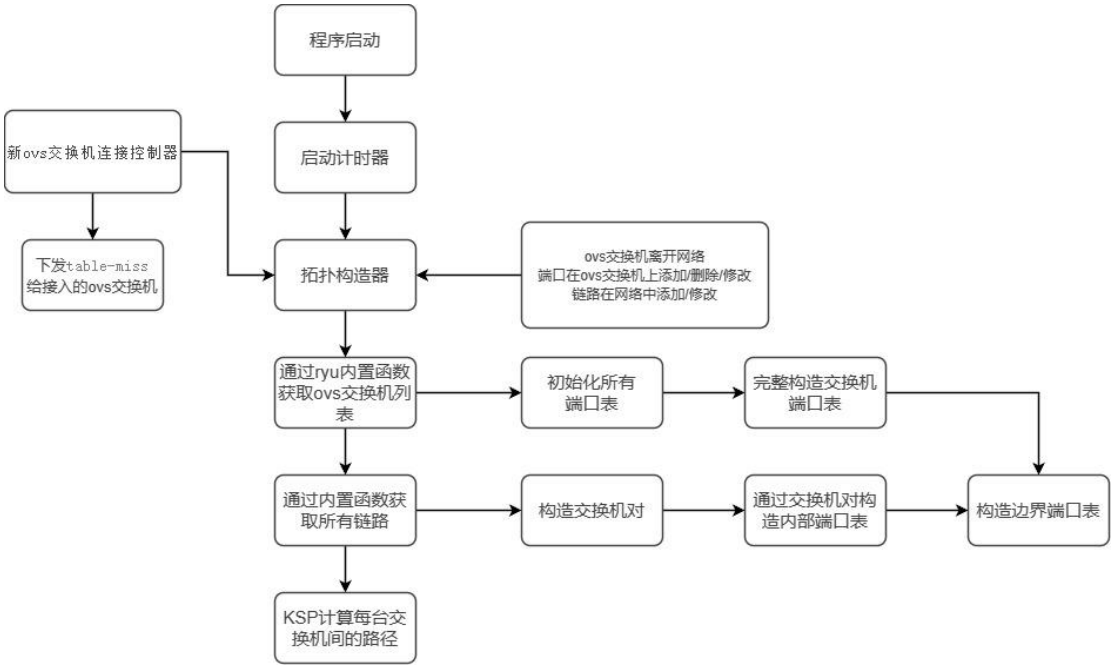



**图 3 拓扑发现模块流程图**

## 3.2 性能监测模块

QoS 路由方案需要根据网络中路径 QoS 参数计算备选路径的 QoS 效用值。因此，实时获取全网链路的性能参数是 QoS 路由方案的必要准备。本系统在 Ryu 控制器中设计了性能监测模块，监测 SDN 网络中每条链路的实时性能，以便进行 QoS 路由方案的应用。性能监测模块主要功能是获取链路的吞吐量、时延、时延抖动和丢包率。

(1)吞吐量

在 SDN 网络中，通常使用 OpenFlow 协议来获取链路带宽数据。一条链路的带宽取决于链路源端口和目的端口之间的流量。因此，可以通过获取两个端口的收发字节数来确定链路的带宽。在 OpenFlow 协议中，可以通过发送 STATS_REQUEST 报文至 SDN 控制器下的 OpenFlow 交换机来获取端口的统计信息。在 OpenFlow 交换机回复的 STATS_REPLY 报文中，可以获取端口接收和发送的字节数。

基于此原理本系统每 t 秒采集一次吞吐量，设第一次收到 STATS_REPLY 时源端口 p 发送的字节数为 b1，第 2 此为 b2。则吞吐量为(单位:bps,比特/秒):

$$Throughput=(b1-b2)*8/t$$

通过在前台系统输入的链路额定带宽我们可以得到链路剩余带宽。

(2) 时延

OpenFlow 协议是链路时延监测的基础，通过传输 PACKET_IN、PACKET_OUT 报文以及

ECHO_REQUEST 和 ECHO_REPLY 报文在 ONOS 控制器和数据平面之间实现链路时延监测。

设交换机 S1 和 S2 及其相应的端口 p1、p2。

本系统利用了 Ryu 自带的 Switches 模块的数据，从 Packet_in 中解析 LLDP 数据包，获得源 dpid，源端口，然后获取 LLDP 数据发送时的时间戳，获得该包时的时间戳进行相减得到单向的时延。通过上述方法获取 p1 和 p2 间的双向时延 t1、t2，此时的 t1 和 f2 还包括控制器到 OVS 交换机之间的 echo 往返时延，故还需要得到控制器到 S1、S2 之间的 echo 往返时延 t3、t4。

$$Delay=(t1+t2-t3-t4)/2$$

(3) 时延抖动

监测链路时延抖动是在进行链路时延监测的基础上实现的。设上次得到的链路时延为 d1,当前得到链路时延为 d2，采样时间间隔为 t:

$$Jitter=|d1-d2|$$

(4) 丢包率

OVS 交换机间会定期发送 LLDP，设一定时间内交换机 S1 向交换机 S2 发送 x 个 LLDP 包,交换机 S2 收到 y 个包。

$$Loss=(x-y)/x$$

## 3.3 异常流量监测模块

异常流量监测模块用于实现对 DDOS 的检测、溯源与排除。当检测到 DDOS 攻击后，该模块会从拓扑发现模块中得到全局拓扑视图，并根据性能监测模块提供的链路数据对异常流量进行溯源，当找到始发 OVS 交换机后，在异常流量源端口下发流表阻断目的地址为当前受害者 IP 的数据包。

熵是信息论的重要组成部分。熵可以度量进入网络的数据包的随机性，这是将熵用于 DDoS 检测的主要原因。熵随随机性的增大而增大，随随机性的减小而减小。公共熵包括信息熵、平均能量、平均 TeagerKaiser 能量、Shannon 小波熵和对数能量熵。本系统考虑了使用目的 IP 地址的概率，对于只有一个变量的情况下，本系统主要利用信息熵和对数能量熵，通过融合利用互补性达到提高检测效果的目的。

要计算信息熵，首先是计算目标 IP 地址的概率。变量 x 用于定义数据包的目的 IP 地址，x 的概率由等式(1)计算。将窗口中的数据包数设置为 n。窗口中每个元素的概率定义为 P。

$$P_i = \frac{x_i}{\sum_{i=1}^{n} x_i}$$

在等式中，i={1,2,3·.n}，0<TT<1

然后计算信息熵，其计算公式如下。H 是出现在特定数据包中的目标 ip 地址的信息熵。

$$H1 = -\sum_{i=1}^{n} p_i \log p_i$$

对数能量熵作为熵的另一种形式，其计算公式如下所示。n 和 pi 仍然表示数据包的数量和目标 IP 地址的概率。

$$H2 = -\sum_{i=1}^{n} \log {p_i}^2$$

研究发现，当攻击发生时，信息熵的熵值会明显降低，但不能快速检测到攻击而对数能

量能快速检测到攻击，但熵值不如信息熵明显。

考虑通过加权融合的两个熵，以达到互补的效果。由于 p 的范围从 0 到 1，根据对数函数的数学性质，对数能量熵将得到一个负值。 为了更好地与信息熵相结合，我们将日志能量熵乘以负 1，并用信息熵进行加权。这一变化体现在上述方程中。权值的选择是根据两种熵在攻击发生时熵下降的变化率来进行的。融合熵有效地实现了信息熵和对数能量熵的优势互补，既能快速检测出攻击又具有较高的熵下降率。

当在一个特定的窗口中，在同一台主机或交换机端口上接收到多个数据包，并且数据包的数量超过阈值时，就会检测到 DDoS 攻击。在攻击过程中，如果指定窗口的计算熵持续下降到阈值以下，则指定交换机上的目标端口将被阻塞。

## 3.4 Meter 限速模块

Meter 表是 OpenFlow 中的一种流量计量模块，在实际网络中可以用来控制业务流量的速率，从而避免网络拥塞和流量过载，提高网络的稳定性和可靠性。Meter 表可以根据流量类型设置不同的限速规则，并且可以在不同的时间段内自动切换限速策略，灵活地应对不同的业务需求。Meter 表还可以根据不同的业务流量进行优化带宽使用，从而提高网络带宽的利用率。

基于业务流量分类模块，流表可以通过匹配不同的流量标记来区分业务流量，在进行网络限速时，需要将 meter 表与流表绑定使用，因此可以实现 meter 表对一类业务流量进行限速。在网络中，某些流量如果过大，可能会对网络造成拥塞，进而导致丢包严重等问题。限速就是通过控制发往特定网络接口上的流量来防止这种拥塞情况的发生。Meter 表基于各个流的数据包计量，可以按照不同的流量需求进行限速，并在达到限制时进行措施，比如丢弃部分或全部数据包，或者将数据包排队等待后续处理，从而提高网络的可靠性和稳定性。在 Meter 表中，可以设置不同的 Bands，每个 Bands 对应一个速率和一个动作，动作可以是丢弃该流量、放行该流量、或将该流量标记上不同的 QoS 等级;在速率上，传统方法是通过数值进行速率的调整，由于 meter 表是与流表进行绑定的，每条链路的速率可能存在差异，传统方法的数值调整存在灵活性较差的表现，因此，我们采用 OpenFlow 协议中的 OFPMF_BAND_PERCENTAGE 标志来对 Meter 表进行配置，采用百分比的方式对速率进行调整，通过限定速率百分比的方式，通过将不同的流量流经不同的 Bands，可以对流量进行分类和控制，以达到限制流量速率、减少拥塞等效果。

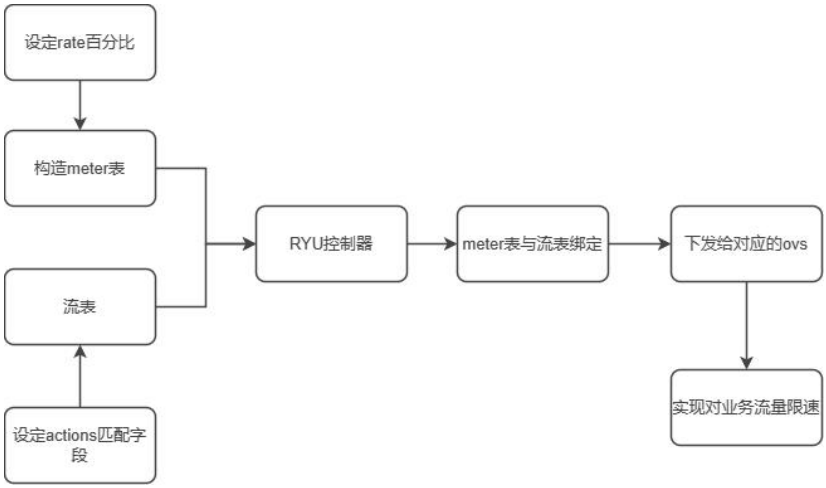



**图 4 Meter 限速**

## 3.5 队列限速模块

除了根据不同业务的 QoS 需求区分外，QoS 路由算法还需要考虑交换机具备区分业务优先级的能力，以实现网络流量的优先转发。在网络拥塞时，会对高优先级业务进行优先处理。本系统设计了队列配置模块，利用 OVSDB 协议向数据平面的 OVS 下发队列配置命令，配置优先级队列，实现高优先级业务的优先转发。

## 3.6 流量预测模块

在流量预测模块当中，使用到的算法主要是 LSTM（Long Short-Term Memory，长短期记忆）算法，是循环神经网络的其中一种变体，相比传统的 RNN，LSTM 能够更好地解决长期依赖问题，适用于需要捕捉长期上下文信息的任务。

LSTM（长短时记忆网络）是一种基于循环神经网络（RNN）的模型，用于解决长期依赖性问题。它通过增加记忆单元和控制器来扩展 RNN 模型。

LSTM 网络的关键组件是一个称为门（gate）的控制器，用于控制输入、输出和记忆单元之间的信息流动。LSTM 网络包括输入门、输出门和遗忘门，这三个门通过使用 sigmoid 函数和点乘运算来控制数据的流动。

具体的数学原理如下：

1. 输入门（Input Gate）：

输入门用于控制输入数据对于记忆单元的影响力，包括当前输入和前一时刻的记忆单元。输入门的输出结果是一个 0 到 1 之间的数值，表示输入数据的重要程度。

公式：$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$

其中，$i_t$表示输入门的输出结果，$h_{t-1}$表示前一时刻的记忆单元，$x_t$表示当前时刻的输入数据，$W_i$和$b_i$分别表示输入门的权重矩阵和偏置向量，$\sigma$表示 sigmoid 函数。

2. 遗忘门（Forget Gate）：

遗忘门用于控制前一时刻的记忆单元对于当前时刻记忆单元的影响力。和输入门类似，遗忘门的输出结果也是一个 0 到 1 之间的数值，表示前一时刻的记忆单元的重要程度。

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

其中，$f_t$表示遗忘门的输出结果，$W_f$和$b_f$分别表示遗忘门的权重矩阵和偏置向量。

3. 记忆单元（Memory Cell）：

记忆单元用于存储 LSTM 网络中的长期信息。当前时刻的记忆单元是由前一时刻的记忆单元和当前时刻的输入数据通过输入门和遗忘门计算得到。

$$C_t = f_t * C_{t-1} + i_t * tanh(W_C[h_{t-1}, x_t] + b_C)$$

其中，$C_t$当前时刻的记忆单元，tanh 表示双曲正切函数，$W_C$和$b_C$分别表示记忆单元的权重矩阵和偏置向量。

4. 输出门（Output Gate）：

输出门用于控制当前时刻的记忆单元对于输出结果的影响力。和输入门类似，输出门的输出结果也是一个 0 到 1 之间的数值，表示当前时刻记忆单元的重要程度。

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

其中，$o_t$表示输出门的输出结果，$W_o$和$b_o$分别表示输出门的权重矩阵和偏置向量。

5. 输出结果：

输出结果是由当前时刻的记忆单元和输出门计算得到的。

$$h_t = o_t * tanh(C_t)$$

其中，$h_t$表示当前时刻的输出结果。

综上所述，LSTM 通过输入门、遗忘门、记忆单元、输出门和输出结果构成一个完整的网络结构，实现长期依赖性的建模和预测。

该模块主要的工作内容就是：为网络当中链路的吞吐量、时延、抖动和丢包率进行预测，为最后的 Qos 路由选择提供数据依据。对流量数据的四个特征进行预测处理，设计出以下方案实施。

（1）数据收集：通过网络检测工具和流量统计器等相关工具获取到链路上的数据，并且保存到数据库当中；

（2）数据预处理：对收集到的数据进行清洗和处理。这包括去除异常值、处理缺失数据、平滑数据等操作，以确保数据的准确性和一致性，尽可能大地把数据真实的准确性提高；

（3）提取数据特征：在获取到的数据当中提取相关有用的特征，可以根据链路的带宽、处理延迟以及正常数据项的正常波动范围等作为特征；

（4）模型选择：在本作品当中选择的模型为 LSTM（Long Short-Term Memory，长短期记忆）算法模型；

（5）训练：使用历史数据对选定的模型进行训练。将数据集分为训练集和验证集，通过调整模型的参数和超参数来优化模型的性能。利用历史数据循环训练加强模型对数据集特征的学习，提高模型预测的准确性；

（6）预测：将训练好的模型初步用于对未来数据的预测，然后与验证数据集进行对比，如果准确度高进行下一步的算法评估，进而考虑是否能用于真实链路中未来流量预测，反之则继续循环训练；

（7）评估：评估预测结果的准确性和性能。可以使用常见的评估指标，均方根误差（RMSE）、平均绝对误差（MAE）来评估模型的预测能力；

（8）更新：因为真实链路当中的流量不是稳定的，所以要根据实际的情况来更新算法的相关参数或者使用最新的训练模型代替旧模型。

## 3.7 流量分类模块

该系统的主要目的是保障网络的 QoS，因此需要能够准确地对网络中的流量进行分类和管理。它需要能够以高精度、高效率地对流量进行分类，以便后续的流量管理。

为了实现这一目标，我们采用了基于机器学习的流量分类技术。具体而言，我们通过收集和处理网络中的流量特征数据，然后利用机器学习算法对这些数据进行分析和训练，最终得到一个高精度的流量分类模型。在该系统中，我们采用了 SDN（软件定义网络）技术作为底层网络架构，这可以帮助我们更好地管理和控制网络中的流量。在 SDN 架构中，流量分类模块主要由控制器和数据平面组成。控制器的主要作用是收集和分析网络中的流量数据，并将其传送给数据平面。数据平面则用于进行实际的流量分类和管理。

朴素贝叶斯是基于贝叶斯定理的一种分类算法，其数学原理为。

假设有 N 个样本，每个样本有 M 个特征，设第 i 个样本的特征为$x_{i1}, x_{i2}, \ldots, x_{iM}$，其对应的分类为$y_i$，则对于新的样本$(x_1, x_2, \ldots, x_M)$，我们需要求得其分类 y 的概率。根据贝叶斯定理，我们有：

$$P(y|x_1, x_2, \ldots, x_M) = \frac{P(x_1, x_2, \ldots, x_M|y)P(y)}{P(x_1, x_2, \ldots, x_M)}$$

其中，$P(y)$为先验概率，表示在没有任何特征信息的情况下，一个样本属于某一类的概率；$P(x1, x2, \ldots, xM|y)$表示样本在给定分类 y 的情况下，各个特征出现的条件概率；$P(x1, x2, \ldots, xM)$表示样本在任意分类情况下出现的概率。因此，我们只需要计算后两项的

值，并比较各个类别的后验概率，即可得到样本的分类。

由于朴素贝叶斯算法在计算$P(x1,x2,\ldots,xM|y)$时假设所有特征相互独立，即：

$$P(x_1,x_2,\ldots,x_M|y)=\prod_{i=1}^{M}P(x_i|y)$$

在这一过程中，我们采用了 OpenFlow 协议来实现控制器和数据平面的交互，以便更好地控制和管理网络中的流量。流量分类模块主要方案实施由以下几个步骤：

（1）数据采集：在该系统中，当未知请求发起时会将数据包通过 Packet-In 上传至控制器进行分析。通过网络流量并收集相关特征数据，我们可以更好地对流量进行分类和管理。

（2）特征提取：在收集到流量数据之后，我们需要进行特征提取以便进行后续的机器学习训练。我们可以通过提取网络流量的应用、协议、源 IP、目的 IP、源端口、目的端口等特征来描述流量，并建立特征向量进行下一步的分类训练。

（3）分类训练：在特征提取之后，我们使用机器学习算法进行训练，以确定对应的流量分类。在分类训练中，采用多种机器学习算法，支持向量机（SVM）、朴素贝叶斯（NB）等。

（4）流量分类：在完成分类训练之后，我们可以将训练好的模型应用到实时的流量数据分类中。通过对流量进行分类，我们可以更好地管理和控制网络中的流量，以保障网络的 Qos。

## 3.8 路由管理模块

路由管理模块负责路由到流表的映射。当其它模块向路由管理模块下发路由时，路由管理模块会判断是否存在冲突路由，并决定是丢弃路由还是替换路由。当完成冲突检测后路由管理模块会解释路由信息，并下发或删除流表。

## 3.9 路由计算模块

QoS 路由方案运用响应式体系结构，让控制器处于被动式请求模式。QoS 路由方案和 Ryu 控制器同时被启动。

路由计算模块分成 5 个部分:ARP 处理机、备选路径计算、效用值计算、最优路径选择、路由自调整。

（1）ARP 处理机

当 OpenFlow 交换机收到业务报文时，如果不能匹配已经安装的流表项，该报文将被上发到控制器，调动数据报文处理器进行处理。控制器在接受数据报文后，将从中识别以太网帧和源端口，如果此时以太网 Type 字段为 ARP 类型则会触发 ARP 处理机。ARP 处理机从拓扑发现模块获取终端列表，若未发现目的 IP 则进行泛洪。当其它 OVS 交换机收到泛洪包时 Packet-In 到控制器，若同一台交换机收到两次相同的 ARP 包则丢弃。若终端列表中存在目的 IP 则有控制器进行 ARP 代理。

（2）备选路径计算

首先，以全局连通拓扑图、源主机所在的网络交换机和目标主机所在的网络交换机作为输入参数进入算法程序，通过 K 条最短路径算法计算出 K 条最短路径作为备用路径。与此同时，路由管理模块还可以实时获得流量预测模块提供的链路流量数据，对于流量预测模块认定为拥塞的链路，路由管理模块将其链路设置中断，此时在 KSP 路径计算结果中不会出现包含该拥塞链路的路径。

（3）效用值计算

根据性能监测模块获取的链路时延、波动和数据包丢失率，以及流量预测模块提供的链路流量预测信息，计算出每条备选路径的 QoS 指标。 根据根据业务先定义的效用函数和备选路径的 QoS 指标，计算出每条备选路径的 QoS 效用值。

（4）最优路径选择

当得到备用路径 Qos 效用值后，从流量预测模块中获取预测的链路状态数据重新计算路径 Qos 效用值。将当前时刻与未来时刻的 Qos 效用值结果做权衡，避免贪心算法。例如当前时刻备选路径的 Qos 效用值差距较小，但未来时刻 Qos 效用值差异较大或路径排名发生变化的时候，将选择未来时刻计算出的路径以应对即将到来的网络状态变化。

（5）路由自调整

当流表下发后，记录当前的路由信息，并定时重新计算 Qos 效用值。即重复执行第（4）步。若最优路径发生变化则自动调整。

## 3.10 前台系统

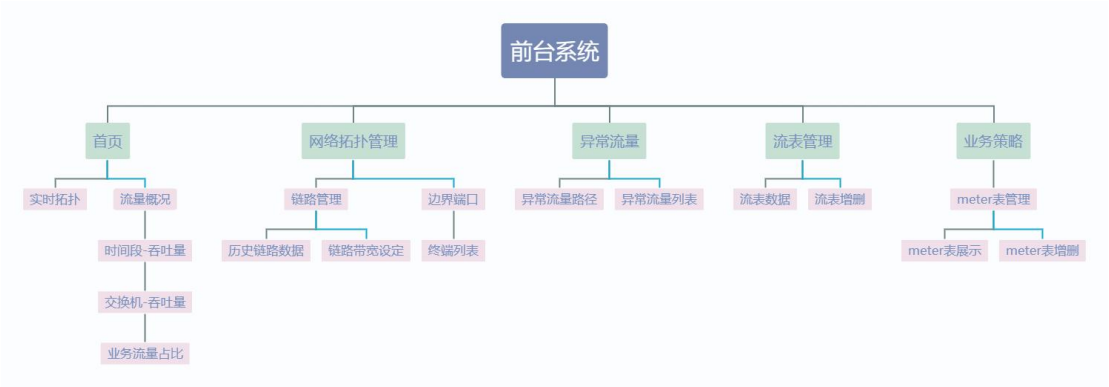



**图 5 前台系统功能展示**

# 四、 完成情况

1. 实现拓扑发现模块和性能监测模块
2. 实现流量预测模块
3. 实现异常流量监测模块
4. 实现 Meter 限速模块
5. 前台系统主体完成
6. 路由计算模块已实现业务流路由计算功能。
7. 路由管理模块可以实现基本的流表下发和删除

# 五、 运行结果


**图 6 首页**

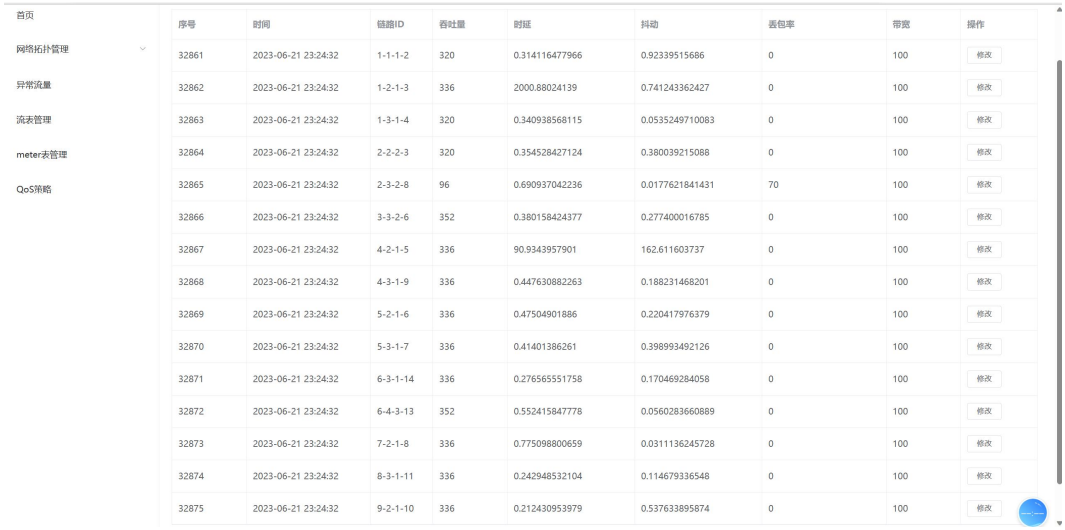



**图 7 链路状态信息查看**

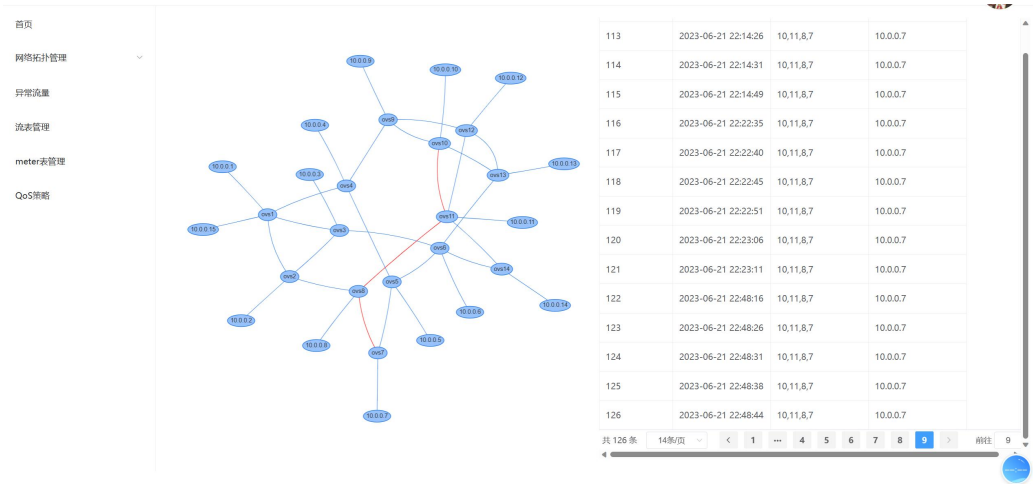



**图 8 异常流量检测**

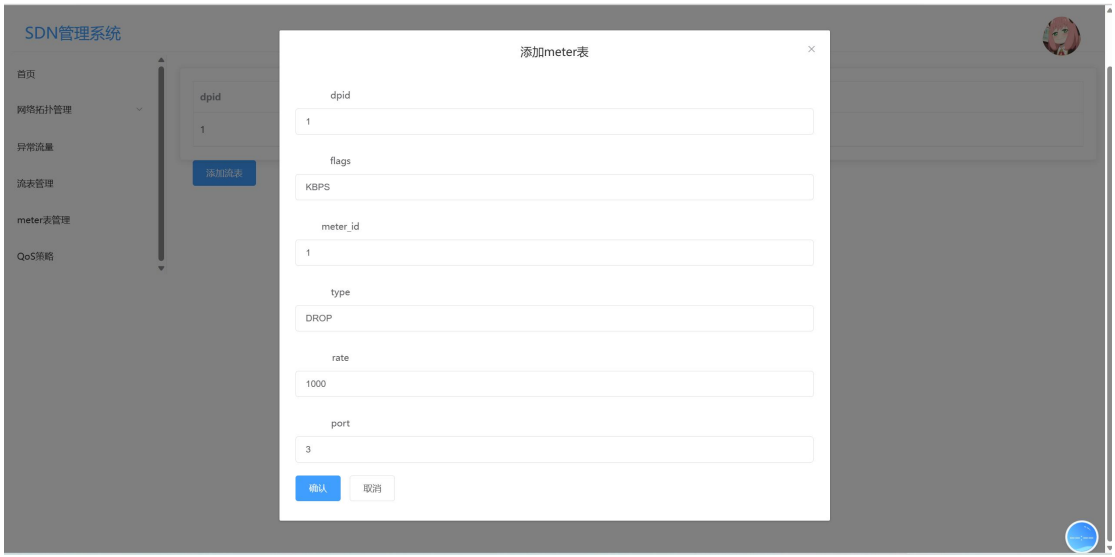



图 9 Meter 表下发

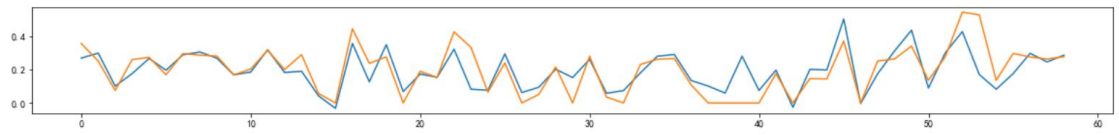
```
root@mininet-vm:/home/mininet/mininet/custom# iperf -c 10.0.0.1 --tos 16
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 65] local 10.0.0.3 port 52834 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 65]  0.0-10.0 sec   896 KBytes   733 Kbits/sec
root@mininet-vm:/home/mininet/mininet/custom#
root@mininet-vm:/home/mininet/mininet/custom# iperf -c 10.0.0.1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 65] local 10.0.0.3 port 52836 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 65]  0.0-24.0 sec   256 KBytes  87.3 Kbits/sec
```

图 10 Qos 路由测试

```
[('[3, 2, 1]', 99.61431249781977), ('[3, 6, 5, 4, 1]', 85.03471952439035), ('[3,
1]', 83.1122555555847)]
[PATH]10.0.0.3<-->10.0.0.1: [3, 2, 1]
[PATH]10.0.0.1<-->10.0.0.3: [1, 3]
```

图 11 备选路径 Qos 效用值

```
[<matplotlib.lines.Line2D at 0x224562c6af0>]
```



```
def MAPE(true, pred):
    diff = np.abs(np.array(true) - np.array(pred))
    return np.mean(diff / true)
def RMSE(predictions, targets):
    return np.sqrt(((predictions - targets) ** 2).mean())

print(f"根均方误差(RMSE)：{RMSE(Y_predict_real/(1024*1024), Y_test_real/(1024*1024))}")
print(f"平均绝对百分比误差(MAPE)：{MAPE(Y_predict, Y_test)}")
```

```
根均方误差(RMSE)：0.09023123862370759
平均绝对百分比误差(MAPE)：0.3395939304942712
```

**图 12 流量预测结果（黄线：预测值，蓝线：实际值）**

# 六、 创新与特色

1. 将 SDN 与机器学习相结合，不仅能够实现网络的灵活配置和优化，还能通过智能决策和自适应学习，快速应对不同网络环境下的挑战，为用户提供个性化的高效服务。
2. 在为业务流提供 Qos 服务前先对异常流进行检测和排除，为业务流传输提供保障且使 Qos 服务更加精确。
3. 使用流量预测对链路进行拥塞预测和剩余带宽、时延、抖动、丢包率等进行预测，并将其结果作为 Qos 路由计算的参数。且会根据未来一段时间网络状态的变化自主调整当前路由以优化网络状态。
4. 使用机器学习进行异常流量检测和流量分类具有处理流量数据时可以获得更高的准确性、可以有效适应新的网络和应用场景，从而具有更好的可扩展性、可以根据网络环境的变化和攻击手段的变化而自动适应以及适合在实时和动态环境中处理网络流量，可以迅速地检测和响应流量的变化和异常。

# 七、 参考文献

[1]薛晓宇,龙杰,方义成.基于 LSTM 算法的无线网络流量预测研究[J].长江信息通信,2021,34(10):4-6.

[2]潘成胜,王羽夫,杨力.基于改进 LSTM 算法的天地一体化信息网络流量预测[J].天地一体化信息网络,2020,1(02):57-65.

[3]姜梦雅. 基于 LSTM 的 SDN 网络流量预测研究[D].北京邮电大学,2021.DOI:10.26969/d.cnki.gbydu.2021.002727.

[4]郭佳丽,邢双云,栾昊等.基于改进的 LSTM 算法的时间序列流量预测[J].南京信息工程大学学报(自然科学版),2021,13(05):571-575.DOI:10.13878/j.cnki.jnuist.2021.05.009.

[5]王海宁,袁祥枫,杨明川.基于 LSTM 与传统神经网络的网络流量预测及应用[J].移动通信,2019,43(08):37-44.

[6]张梓强.基于 ONOS 的 SDN 网络 QoS 优化设计与实现[J].电子科技大学,2022,DOI:10.27005/d.cnki.gdzku.2021.001885

[7]Journal/Article: Fadil, A.; Riadi, I.; Aji, S. Review of Detection DDOS Attack Detection Using Naive Bayes Classifier for Network Forensics. Bull. Electr. Eng. Inform. 2017, 6

[8]Journal/Article: Dayanandam, G.; Reddy, E.S.; Babu, D.B. Regression algorithms for efficient

detection and prediction of DDoS attacks. In Proceedings of the 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Tumkur, India, 21–23 December 2017;