

Design and Validation of a Meter Band Rate in OpenFlow and OpenDaylight for Optimizing QoS

Mohamed Saleh Al Breiki^{*1}, Suiping Zhou¹, Yuan Roger Luo¹

¹Department of Computer Science, Middlesex University, NW4 4BT, United Kingdom

ARTICLE INFO

Article history:

Received: 15 October, 2019

Accepted: 01 February, 2020

Online: 09 March, 2020

Keywords:

Cloud Computing

Meter Band Rate

Network Function Virtualization

OpenDaylight

OpenFlow

Quality of Service

Software Defined Networks

ABSTRACT

Technological developments in the Internet and communications have created a vastly complex and dynamic context with diverse heterogeneous networks and fast growth of mobile devices and multimedia. As the Internet becomes the primary mode of communication for many organisations there is requirement to enhance quality of service (QoS) from heterogeneous systems and networks. Traditional networks such as TETRA have become increasingly incapable of addressing the demand for media rich, bandwidth intensive traffic flows and applications. Mission-critical multimedia over new generation mobile networks face QoS constraints. This research explores a novel solution for quality of service performance for streaming mission-critical video data in OpenFlow SDN networks. A Meter Band Rate Evaluation (MBE) mechanism is advanced that improves the native QoS capability of OpenFlow and OpenDaylight. The MBE is a physical component added to the OpenFlow meter table to evaluate and dynamically adjust traffic rates and allows the traffic volume to be specified relative to other traffic in the network. Its design and development are presented, and the mechanism is verified through a simulated experiment in an SDN testbed. The results identified that QoS performance experienced a significant percentage increase when the MBE was active. These findings contribute a novel Meter Band Rate Evaluation mechanism that extends the native capability of OpenFlow and OpenDaylight to enhance the efficiency of QoS provision.

1. Introduction

Software Defined Networking (SDN) has received increasing attention as a new approach implementing new and innovative cloud-based architectures capable of integrating quality of service (QoS) enhancing end-to-end video performance. QoS dimensions of SDN and Network Function Virtualization (NFV) have the potential to address reliability and the quality of mission-critical applications over hybrid networks including 4G, 5G, Terrestrial Trunked Radio (TETRA) [1]. SDN is a novel technology designed to be dynamic and flexible to modern computing contexts by offering improved network management features [2] and enhanced network control for cloud computing [3].

The context of this research is UAE law enforcement in which there has been a growing dependence on mobile real-time video communications. TETRA networks have become increasingly incapable of addressing the demand for media rich, bandwidth intensive traffic flows and applications [4]. Modern policing like all areas of society is engaged in intensive sharing of different forms of real-time data and media across a diverse range of devices

and platforms that form a critical part of its operations [5]. This context underlines the need to enlarge policing capabilities through the adoption of commercial mobile broadband and wireless networks and leveraging of cloud computing and SDN and NFV technologies to map multimedia applications over heterogeneous underlying networks. The programmability and the abstraction of the control plane from the forward and data plane is a key aspect of SDN's design [6,7]. This enables a single software control programme to control the data plane of numerous network elements [8] and enables greater and more flexible control and management of network components.

This paper contributes a novel Meter Band Rate Evaluation mechanism that extends the native capability of OpenFlow and OpenDaylight to enhance the efficiency of QoS provision. This research experiments with and analyses the implementation of an integrated set of technologies (SDN OpenDaylight controller, OpenFlow protocol and OpenQoS) over a centralized cloud platform using real-time video to 4G mobile devices. The primary goal of this research is to evaluate and validate the application of SDN and OpenFlow in realizing QoS for real-time video data transfer in a hybrid cloud computing context. A key objective is to

^{*}Mohamed Saleh Al Breiki, +1 646 580 6284, ma2634@live.mdx.ac.uk

monitor QoS parameters and evaluate the SDN-based architecture in optimizing QoS performance. This paper presents a novel QoS mechanism that expands capabilities by incorporating a new meter band rate component. The experiment results in this paper verify the performance of this mechanism. This contributes a new mechanism and demonstrates the potential to expand the features of OpenDaylight SDN controller to realize more optimized QoS for real-time video traffic over heterogeneous networks. Thus this research represents a novel QoS mechanism that advances existing QoS capabilities of OpenFlow and OpenDaylight.

2. Related Work

Multiple studies have focused on improving quality of service (QoS) by means of leveraging the network control components of SDN by focusing software solutions at the switch level. This allows for transparency of network resource interfaces in alignment with application requirements.

Quality of service concerns the extent to which applications receive expected service quality relative to their level of mission-criticalness [1,9]. Hakiri and Berthou [1] indicate that QoS provision lacks this responsiveness to guarantee the required service level. The design of the OpenFlow protocol allows the SDN controller to access and modify flow data at the switch level for all devices thereby connecting the control and data planes [10]. OpenFlow flexibility allows for data to be automatically directed with minimal disruption achieved by decoupling the forwarding layer from the control layers [11] which manages data flows and creates or updates rules for devices in the network [12].

Changes in hardware are not required by employing OpenFlow and vendors do not need to open systems to support it provided a small number of widely used interfaces are available to the control plane [13]. Any system that integrates the protocol is capable of controlling the switch behaviour of any devices in the Openflow network. Directives allow ISPs to deploy SDN to guarantee quality of service of streaming services alongside traditional network services [13].

The significant potential to manage QoS within the OpenFlow protocol has generated multiple studies focused on addressing QoS issues at network level. A key attribute of OpenFlow is the focus of QoS to address specifically different parameters such as delay, throughput and jitter [14] as a result of the control of packet and flow level activity. This granularity permits users of OpenFlow to specify how individual flows are to be managed in accordance with Integrated Services (IntServ) and open standards definitions [15]. Individual flows can also be amalgamated into classes by users. OpenFlow grants access to a set of programming tools which enable the generation and recycling of virtual flows or slices. Thus the user can stipulate how network resources including routers, switches and queues are allocated to different slices with different priorities [15].

In [2], the author proposes an SDN-based solution to ensure network-wide QoS based on multi-paths routing that utilizes OpenFlow's queue support to maximize the efficiency of resources in line with the network user requirements and minimizing excess provisioning. The solution focuses guaranteeing end-to-end QoS for individual data flows and the efficient management of open virtual switches (OVSs). The OVS configuration is recalculated in

response to new network demands or as users progress on to new nodes in order to minimize the utilization of OVSs in the network.

In an attempt to guarantee bandwidth end-to-end Tomovic et al. [16] advance an explicit QoS guarantee as in IntServ by incorporating controls for admission and reserving bandwidth in OpenFlow. This commences by transmitting the bandwidth requirements of a QoS flow initiating a Constrained Shortest Path computation by the controller in which the required bandwidth is used as a limiter. The technique uses the available unreserved bandwidth as the link weight ensuring that the reserved flows are allocated to bandwidth that has been reserved. The flow is rejected where the bandwidth is not available for end-end path. Furthermore, there is intermittent checking by the controller of all links associated with the QoS flow, with utilization greater than 80% resulting in the re-routing of best-effort flow to an alternative path with lower utilization.

To maintain QoS in real-time video streaming in SDN Liang and Shen [17] introduce a new SDN-based routing approach that is capable of dynamically routing deployment across layers of Scalable Video Coding (SVC) accounting for bandwidth and different QoS parameters: delay, and packet loss. SDN research by Celenioglu and Mantar [18] advanced a routing and resource control approach for enhancing QoS that takes into account pre-defined paths and reservation of resources between network segments. By improving routing scalability and decreasing admission time the solution provides QoS guarantees to stationary nodes.

Research into Openflow by Egilmez et al. [11] contributed an OpenQos controlled solution that concentrated on achieving end-to-end QoS which secures an optimal path for the service delivery of application traffic that prioritize multimedia traffic [19]. Flows for data, video and voice are categorized and subject to different algorithms either shortest routing for data or dynamic routing to guarantee QoS.

In a new approach [8] multiple issues are identified: the requirement for each node device to configure the same information; need for manual configuration of all ports of each forwarding device which can be error-prone and time-consuming; and the incapacity of legacy networks to make a distinction between the different resource requirements of voice and data network traffic. Data traffic requires more bandwidth while voice is associated with small packets and delay sensitivity. Achieving QoS in SDN is predicated on two key steps of marking packets for the data plane and management at the control plane. The network is made up of multiple virtual networks whose resources and requirements are autonomously monitored and managed [8].

To enhance QoS some research has focused on an extension of OpenFlow. The purpose of research by Lu et al. [20] was to provide a solution for QoS using conventional flow control techniques and maximizing flow. A QoS slice method and forwarding mechanism is advanced nevertheless the study does not address the issue of supporting QoS in OpenFlow natively. Other studies undertaken in this field do not examine the constraints associated with QoS by OpenFlow [8, 21, 22, 23]. Bhattacharya and Das [24] propose that a module is embedded to minimize any communication latency between switch and controller by means of caching flows nevertheless a QoS mechanism for the data plane is not advanced.

Examining the utilization of DiffServ and multipath QoS routing in OpenFlow Jinyao et al. [25] propose HiQoS as a method to guarantee QoS. In HiQoS traffic is classified into three groups of best effort data stream, low delay interactive multimedia and high throughput video streaming. The switches forward the different categories of traffic into three preconfigured queues with fixed rates that effectively divides bandwidth. Flow paths are determined by employing existing bandwidth usage as weight and flows from the same category of traffic are able to utilize different paths. While the method distinguishes low delay traffic as a distinct category it fails to account for delays in relay-time and calculates routes using existing bandwidth information. The assumption is that greater utilization in terms of bandwidth and queue leads to greater delay. Therefore HiQoS aims to guarantee bandwidth for categories of traffic rather than providing guarantees at flow level [25].

3. Problem Formulation

While cloud networking solutions based on SDN have specific performance, scalability and flexibility advantages, they are nevertheless limited to supporting specific kinds of network environments and lack more precise control of network paths for attaining objectives such as traffic prioritization or rapid failover [11]. QoS mechanisms including Differentiated Services (DiffServ), IntServ and Multiprotocol Label Switching (MPLS), are subject to numerous limitations in respect of level of flexibility and integration in different systems. The potential of SDN is associated with the ability to extend the diversity and targeting of resource provision so that it better aligns with specific application of QoS requirements [26]. One component of this is a suite of tools that facilitate distributed and collaborative management to achieve better classification of data flows in accordance with traffic characteristics.

The OpenDaylight SDN controller based on the OpenFlow protocol is widely used and has been embraced by vendors worldwide. OpenFlow is a key component within SDN as it controls switches through signalling between switches, guarantees the provision of QoS and is significant in respect of the best effort paradigm. The major benefit of the protocol is backwards compatibility and its ability to be integrated within current network infrastructures and legacy flow-enabled network devices with minimal changes in design. Outlining the present capabilities of OpenFlow's support for QoS in SDN assists understanding of the problem. Currently support is provided using two approaches of either DiffServ or per-flow QoS. Using Per-port Queue, a queue for a flow in a flow-entry can be specified within OpenFlow so that in the event of a match decision is allocated for each queue. Every single queue is configurable with properties, governing the treatment of packets within the queue. OpenFlow's per-flow queue attribute is utilized for the definition of per-flow QoS by means of a per-flow meter table in which single or multiple flows can be attached. The role of meters is to determine packet rate and control that rate. There is also the possibility to attach single flows to multiple meter tables on the condition that different tables are utilized. Within a single table each flow may be assigned to only one meter table. The combination of per-flow queues and meter tables can enable the implementation of DiffServ.

There are a number of drawbacks and areas in which QoS in OpenFlow (versions 1.1-1.5.1.) could be improved. Simple queuing mechanisms such as min-rate and max-rate provide limited support for QoS and queues cannot be configured inside OpenFlow. Only rudimentary support for DiffServ (for example, DSCP Remark) is available. Moreover when using queue, QoS is solely permitted at egress. Lastly, most switches do not support optional meters. The simplicity of the queuing mechanisms engenders two key problems for QoS. Firstly the per-port queue allows the available bandwidth to be sliced into a number of queues and to specify certain properties for each queue of min-rate, max-rate and others. However this action fails to guarantee that the bandwidth available is efficiently utilized. Furthermore over-provisioning to avoid low QoS during congestion may be needed, resulting from the specification of minimum bandwidth for traffic. Behavior is reliant on the particular hardware implemented and is therefore not able to be predicted. A second QoS issue is the restricted scalability entailed by the simple queuing mechanisms. For all traffic, there is a flow-entry requirement to specify which it belongs to however switches have a limitation in the number of flow-entries. For a large network, every switch needs flow-entry and queue configuration for all traffic in the network. Furthermore, OpenFlow's inability to allow for queues to be configured implies that dynamic changes cannot be made. This results from the nature of the SDN controller which is not able to be utilized to configure queues and therefore other mechanisms including OVSDB, OF-Config, or cli must be deployed. These however do not work in the fast path. Consequently end-to-end QoS implementation is challenging using this approach. The basic support provided for DiffServ such as DSCP Remark is a further key issue undermining QoS. Fields are adjustable by means of meter tables nevertheless meters complement rather than replace queues and are unable to surmount the limitations associated with them. Within a packet meters have the capacity to alter certain fields but do not provide support for the comprehensive operation of all kinds of per-hop-behavior (PHB). Illustrating this point, different queuing mechanisms such as strict priority queuing (SPQ) or weighted fair queuing (WFQ) are not able to be applied as solutions for a PHB. In addition there are feasibility issues in applying mechanisms such as Call Admission Control (CAC) when using meters. By permitting QoS at the single point of egress only the use of queues is problematic as there is a distinct advantage in the ability to drop traffic at the point of ingress that safeguards switch resources, when possible, for other traffic. Meters can support this however possess a limited capability for QoS as previously outlined. The prevailing absence of switch support for meters underlines a lack of guarantee that support will be provided by OpenFlow switches.

4. Experimental Study

This research applied an experimental approach in order to test and analyse video streaming quality of service performance under different experimental conditions. This comprised the establishment of an SDN based network testbed based on OpenDaylight SDN controller and OpenFlow 1.3 and utilizing the Mininet emulator to undertake several tests. The testbed was implemented in line with the system parameter and mechanisms indicated and detailed in Figure 1 and Table 1 that specify the key software versions utilized during the implementation.

Table 1: Software versions utilized in implementation

Software	Description
Python 2.7.14	Programming language
Linux	Host OS
Open vSwitch 2.5	Virtual multilayer switch with OpenFlow support
Ubuntu 16.04	Virtual machines
Yang UI	User interface for OpenDaylight
Mobile Phone Samsung 4/4G Client smartphone device	
Networks	WiFi (WLAN), 3G, 4G, SDN Controlled
OpenDaylight Controller	SDN framework
Mininet 2.2.1	Network topology environment

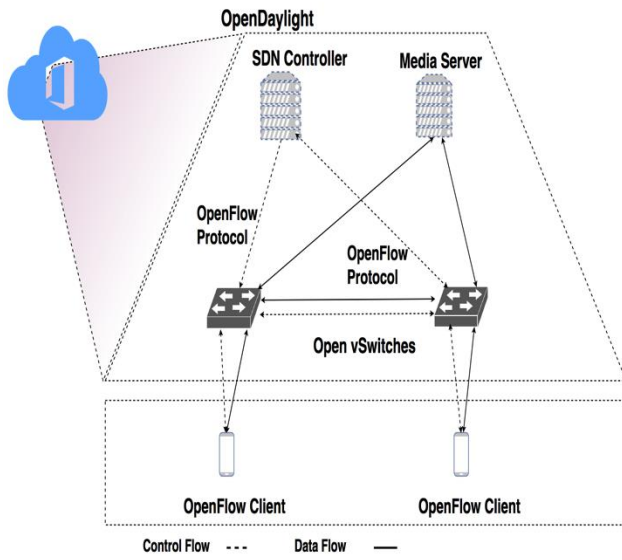


Figure 1: OpenFlow-based SDN network testbed

OpenDaylight Controller (ODL) was utilized as the implementation framework subsequent to assessment of different controllers. Key benefits are associated with ODL in terms of: regular updating relative to other controllers; support for all versions of OpenFlow; necessary features to allow for the implementation of a traffic engineering platform in SDN, and possession of several existing modules able to be used with customized applications. Firstly a series of experiments were undertaken to assess QoS performance for existing SDN/OpenFlow capabilities in a high traffic environment utilizing two distinct sets of experimental conditions. The experiment allowed for detection and analysis of issues associated with QoS and the constraints of SDN/OpenFlow that can be addressed in the following stage of research. The tests provided benchmarks for different QoS metrics of delay, packet delay variance (jitter), packet loss and throughput.

5. Meter Band Rate Design

The design of the Meter Band Rate algorithm is focused on enhancing the native capability of OpenFlow at the switch level. The design of the proposed mechanism is outlined in this section and an overview of MBR model and components is represented in

Figure 2. The evaluator represents the novel component of this design based on a band rate description language (BRDL) for analysing required dynamic states for the mechanism. The BRDL facilitates classification of network traffic. For instance, if there are $i = 1 \dots c$ types of classes in a network with priority of types as follows, $1 > 2 > \dots > c$ such that the network is sliced into c classes. The BRDL syntax enables the specification of slices relative to others within the network:

$$\text{rate}(x) = \{\text{MAX}-\} \{ \text{rate}(y)- \} \dots \{ \text{rate_const} \} \quad (1)$$

Where, MAX is the total available capacity, $\text{rate}(x)$ specifies rate of this traffic class, $\text{rate}(y)$ specifies rate of any other traffic class and rate_const is any constant. Bandwidth reservation is controlled by rate_const and avoids the need for explicit assignments of services to meters. In the case of different types of traffic, for instance voice or data flows, the data traffic needs only to be configured with meter and assigned the necessary bandwidth for voice traffic to rate_const . To exemplify the value of this consider two types of flows in the network of voice traffic and iSCSI traffic. Voice flow has a fixed rate and therefore can be assigned a constant rate. Therefore, only rate_const part will be applied:

$$\text{rate}(\text{voice}) = \text{rate_const_voice} \quad (2)$$

Then, iSCSI traffic is assigned $\text{rate}(y) = \text{rate}(\text{voice})$ is set and the rate_const component is not required. Therefore,

$$\text{rate}(\text{iSCSI}) = \text{MAX} - \text{rate}(\text{voice}) \quad (3)$$

Using this approach, the MBR invocation is either DROP or ALLOW. The benefit of the BRDL is in overcoming the limitation of meters and per-port queues by adding constant max-rate and min-rate limiting features. Further this optimizes traffic flow in terms of throughput and simultaneously guaranteeing QoS requirements of each individual slice. The SDN controller assigns all flows with OpenFlow meter at the install stage with unique meter_id identifier associated with local switch for each flow entry. This action is dependent on the supported meter band type:

1. OFPMBT_DROP
2. OFPMBT_DSCP_REMARK
3. OFPMBT_EXPERIMENTER

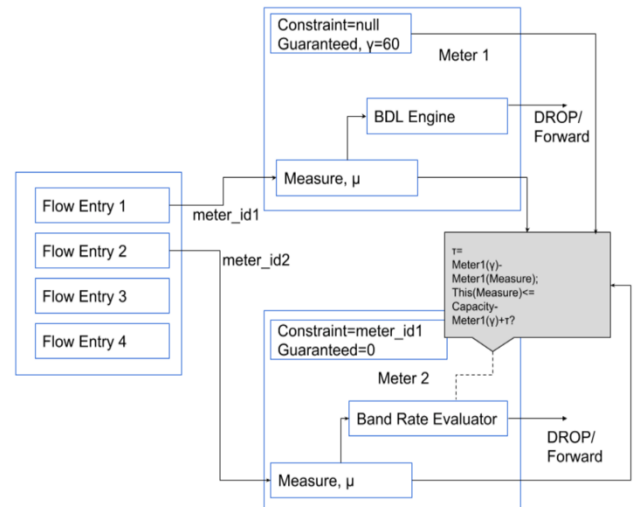


Figure 2: Meter Band Rate model

The OFPMBT_EXPERIMENTER indicates the band rate evaluator in this model. OpenFlow meter bands comprise three bands: type, rate, burst and counters. *Constraint* and *guaranteed rate* extends the meter band to enforce priorities. The *constraint* records the priority of meter_id's supplied by band description language given in Equation (1). Thus the *constraint* list of "band rate evaluator" is determined from $rate(y)$ and $rate_const$ of BRDL. The *guaranteed rate* specifies the bandwidth guarantee for this meter. The *measure* field reflects the measured data rate by the meter. The above values can be specified in KBPS or packets per second as set in meter flags.

Figure 2 depicts the flow-entries associated with meters. Flow 1 is associated with Meter 1 having meter id "meter_id1" and Flow 2 is associated with Meter 2 having meter id "meter_id2". The *band rate evaluator logic* is given in the following section. Thus all incoming packets are associated with a meter flow entry at the ingress port. Figure 3 depicts meter evaluation to determine the band rate to drop the packet, so that where the evaluated rate is higher than the existing rate, the packet is passed.

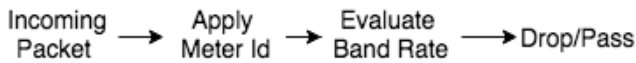


Figure 3: Applying Meter Band Rate Evaluator to all packets

Figure 4 illustrates the evaluation process that treats guaranteed and measure components as *superior constraints*. The evaluator meter uses the *guaranteed* and *measure* parameters of the meters specified in the *constraint* field. Parameters of "higher priority" or "superior" meters, are considered *superior constraints*. The constraint meter list is used to identify superior constraints. Step 1 - the measure field is subtracted from the guaranteed field of each meter in the constraint list, which yields the available bandwidth of each constraint meter; Step 2 - available bandwidths are totalled; Step 3 - the band rate evaluator adds the guaranteed fields and deducts the result from the total port capacity; Step 4 - the results from step 2 and step 3 are totalled. The result from the step 4 is compared with the measure field; if higher, the packet is allowed to pass. In respect of compilation of the band description language into the constraint list used by band rate evaluator three types of traffic are specified: voice, video and data. The data rates of these traffic types in band description language can be expressed as:

$$\begin{aligned}
 \text{rate(voice)} &= 32\text{Kbps (const)} \\
 \text{rate(video)} &= 1\text{Mbps} \\
 \text{rate(data)} &= \text{MAX} - \text{rate(video)} - 32\text{Kbps}
 \end{aligned}$$

The $rate_const$ (=32Kbps) is used in expressing rate (data). Meters will not be required for voice traffic. Thus, there are two meters: *meter_video* and *meter_data*. The *Constraint* and *Guaranteed* fields of these meters are as follows:

$$\begin{aligned}
 \text{meter_video: } &\{ \text{constraint} = \text{null, guaranteed} = 1\text{Mbps} \} \\
 \text{meter_data: } &\{ \text{constraint} = \text{meter_video, 32Kbps} \}
 \end{aligned}$$

The proposed design focuses on the limitations of OpenFlow to provide a flexible and optimum QoS solution. Currently there are two instruments inside OpenFlow to harness traffic behavior, namely per-port-queues and meters. Both queues and meters have their own limitations in terms of versatility and efficiency.

To address these shortcomings the following components have been incorporated in the proposed solution:

- 1) Band description language (BDL)
- 2) Band rate evaluator (BRE)
- 3) Constraint list
- 4) Guaranteed rate

The band description language (BDL) is a logical component of the design which allows the network administrator to specify desired QoS behavior in a flexible and efficient manner. A major limitation of OpenFlow queues is that they only allow for simple QoS behavior such as max-rate limit. This means that rate limited traffic will face restriction despite the availability of free capacity in the network. In this solution, BDL allows the traffic volume to be specified relative to other traffic in the network. As a result, there is no unnecessary bandwidth restriction for any traffic.

The inability to be provisioned in the fast path is another limitation of the queue-based approach. Consequently, the SDN controller is unable to update the queues in real time to reflect network demand. The proposed design solves this issue firstly by making BDL capable of specifying per-flow bandwidth in terms of other flows. Therefore, flow rates are automatically adjusted to network demand. Secondly, OpenFlow meters are used to implement the solution. Meters in OpenFlow can be configured in the fast path, therefore the QoS settings can be updated in real time by the SDN controller in the solution.

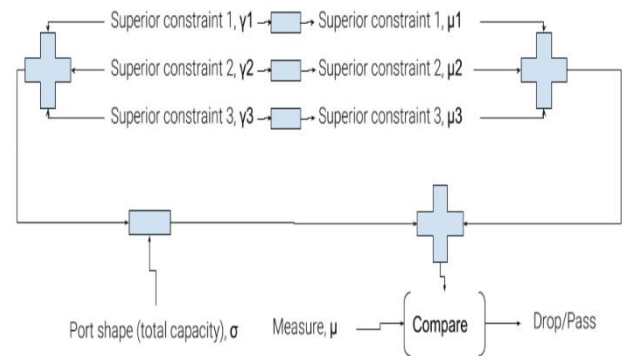


Figure 4: Evaluator process

The band rate evaluator (BRE) is a physical component added to the OpenFlow meter table. Presently, in OpenFlow, band rates can be specified as static constants. When the traffic rate reaches one of these band rates, OpenFlow meter will execute a certain action (e.g. dropping the packet, or modifying certain protocol header fields in the packet). This static meter bands feature can be useful for simple rate limiting but is unable to adjust to dynamic network demand. It is theoretically possible to periodically update these static band rates by the SDN controller. However, in practice the traffic rate is too fast for the SDN controller to be able to

continuously update meter band rates, even in a small network. In this solution therefore BRE is used to dynamically adjust traffic rates based on the BDL.

The constraint list is another physical component of the proposed design. As discussed, the BDL and BRE allow traffic rates to be specified relative to other flows. The constraint list is responsible for storing information in relation to which traffic rates are dependent upon which other traffic. The constraint list is designed to be understood by the BRE whereas the BDL is intended for use by humans and the SDN controller. The QoS rules specified in BDL are processed by the SDN controller to create the constraint list. Once the constraint list is generated, the SDN controller will install it into the SDN switch.

The guaranteed rate is the final physical component of the proposed solution. This allows reservation of a virtually dedicated amount of traffic for a flow. Use of the term ‘virtually dedicated’ signifies that this operation is not a dumb reservation as performed by min-rate limiters in OpenFlow queues. In OpenFlow queues, reserved bandwidth by a min-rate limiter will be unavailable to other traffic when the flow for which bandwidth is reserved has a lower rate than reserved bandwidth. This means that the reserved bandwidth will be wasted most of the time. In this design the BRE will mitigate this problem by allowing the bandwidth specified by the guaranteed rate for a certain flow to be re-used by other traffic as long as there is some free bandwidth unused by the owner of the guaranteed bandwidth. The guaranteed rate can be left unspecified for any flow as long as it is not mentioned in the constraint list of another flow. This can be useful for best effort traffic which requires least priority. For other traffic which has more priority compared to other traffic the guaranteed rate must be given.

6. Meter Band Rate Evaluator Implementation

The Meter Band Rate Evaluator outlined earlier was deployed in OpenFlow Software Switch 1.3. The module ofsoftswitch13 was modified to integrate the mechanism to implement the Meter Band Rate Evaluator. Table 2 depicts the key components and core functions of the Meter Band Rate Evaluator algorithm developed for this implementation. Several modifications were made to the OFLIB. All messages in OpenFlow share common header containing information message type where the default struct for each message allows for two actions: marshalling (packing) and unmarshalling (unpacking). These functions enable oflib-messages transmission of new modifications through wire format. In oflib-struct.h the meter_configuration was modified with statistical function adding to the number of fields to perform metering monitoring of parent free tokens. A flag in meter configuration was added to facilitate enabling or disabling of the band evaluation engine. oflib-messages were modified to allow for band evaluation engine to be set to enabled/disabled states. The central part of the algorithm is deployed in meter_entry.c of the udatapath component to enable parent monitoring support and to apply entries. The Meter Table is responsible for QoS operations and was adapted with fields to define new actions.

The meter table performs the main processing of packets in accordance with the band meter design. Meter_table.c was modified to activate the evaluation engine and increase the structure size from 16 to 20. Finally, the BEEFLAGS within the

utilities/dpctl component was modified to enable tracking and debugging and query of Flow Table state in OpenFlow switches which avoids the requirement for debugging at the controller level.

Table 2: MBE core algorithm

Packet Handling at Meter
Input: <i>packet, monitoringPeriod</i> Output: <i>decision</i>
WHILE <i>parent</i> ← <i>nextParent</i> DO IF <i>free(parent)</i> < <i>free</i> <i>free</i> ← <i>free</i> + <i>free(parent)</i> ENDIF ENDWHILE
<i>elapsedTime</i> ← <i>currentTime</i> - <i>startTime</i>
IF <i>elapsedTime</i> = <i>monitoringPeriod</i> <i>totalFree</i> ← <i>free</i> <i>startTime</i> ← <i>currentTime</i> ENDIF
IF <i>available(meter)</i> > <i>size(packet)</i> <i>decision</i> ← <i>ALLOW</i> <i>available(meter)</i> ← <i>available(meter)</i> - <i>size(packet)</i>
ELSIF <i>free</i> > <i>size(packet)</i> <i>decision</i> ← <i>ALLOW</i> <i>free</i> ← <i>free</i> - <i>size(packet)</i>
ELSE <i>decision</i> ← <i>DROP</i>
ENDIF

7. Results

Two rounds of testing were undertaken with and without the Meter Band Rate Evaluator (MBE). Initial tests conducted without the MBE provided a benchmark for performance and gathered and analysed QoS data for four parameters of delay, jitter, packet loss and throughput under conditions simulating a congested environment. The second stage of testing was undertaken after the MBE had been installed and configured in order to evaluate the capacity of the novel solution to improve QoS performance. The tests undertaken in the first experiment were replicated under the same environmental conditions for periods of 4.3 minutes.

For the four QoS parameters the results showed that there was a marked improvement when the MBE was applied. The figures chart the comparison between the benchmark tests and those for the MBE. Figure 5 reveals significant optimization in delay performance with a reduction in delay time over the period of 260 seconds. In the first round of tests delay ranged on average between 500-1100ms in comparison to results with the MBE enabled of between 80-360ms. Average delay with the MBE disabled was 546.7ms compared to 206.6ms with the MBE.

For the remaining three QoS performance measures the positive impact of the MBE was evident and provides support for the solution to optimize QoS performance within OpenFlow.

Analysis of the data for jitter performance as shown in Figure 6 indicates that in comparison application of the MBE significantly reduced jitter to a median average of 1430 compared to 1906.5 without the MBE. Similarly there was a substantial minimization of packet loss when the MBE was active as shown in Figure 7, indicating a consistent average of under 5% across all tests, compared to between 5-15% when the MBE was inactive. The median average PLR was 2.1% when the MBE was enabled compared to a figure of 8.3% when disabled.

Results also revealed significant optimization of throughput performance, a key QoS performance measure for ensuring the quality of video. As shown in Figure 8 when the MBE is active there is a significant increase in throughput of more than 4000 packets/sec and rising to 9000 packets/sec, a markedly higher figure than when the MBE is inactive. Results without the MBE show that packets/sec varied between 300-4000 packets/sec.

As Table 3 shows for all QoS parameters there was a significant percentage change in performance when the MBE was active. For delay, results reveal an average reduction of 94% reflecting the substantial impact of MBE on performance for this measure. A mean percentage reduction in excess of 800% for jitter is indicated, and when extreme values are accounted for a 32% median average decrease in jitter is shown.

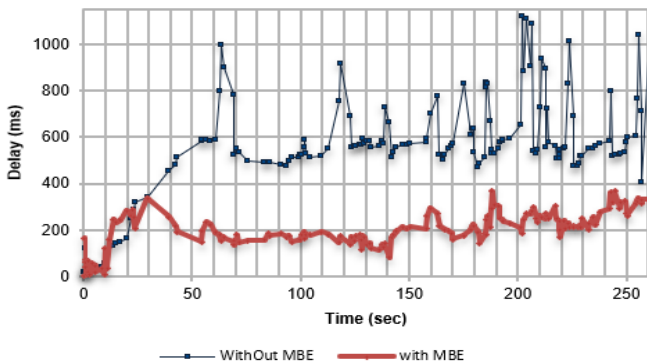


Figure 5: Delay performance with MBE

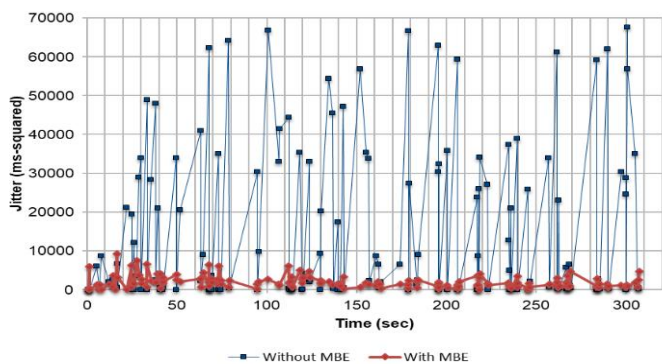


Figure 6: Jitter performance with MBE

Table 3: Percentage change in QoS measures with Meter Band Rate Evaluator

	Delay	Jitter	Packet Loss	Throughput
Mean % Change	-94.16	-866.45	-285.32	200.20
Median % Change	-96.27	-32.76	-325.83	189.49

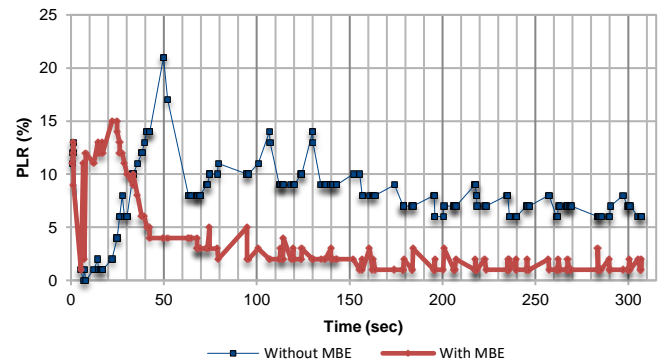


Figure 7: Packet loss ratio performance with MBE

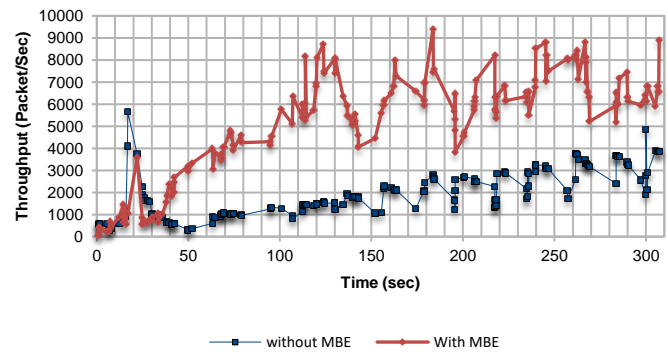


Figure 8: Throughput with MBE

In relation to ratio of packet loss the mean average decrease was 285% and median average decrease of 325%. Lastly, throughput under MBE increased by 200% for the mean average and 189% for the median. Based on the metrics tested these results provide substantial validation and support for the ability of the MBE to improve the native capability of OpenFlow QoS performance.

8. Discussion of Findings

This research explored application of an SDN-based mechanism to enhance QoS for real-time end-to-end video transmission. The solution centres on the development of a Meter Band Rate Evaluator (MBE) mechanism employing a new band rate description language that extends the capabilities of OpenFlow and OpenDaylight to support QoS. This solution was designed and subject to experiments in a Mininet test environment which revealed significant improvements in level of QoS performance across multiple parameters: jitter, throughput, delay, packet loss. This contribution addresses the limitations of existing methodologies to guarantee QoS performance, either through marking data-plane packets and management at the control plane automatic manager for monitoring requirements of virtual networks [8] or slice and forward methods [20]. The findings of this study demonstrate potential to address the limitations of QoS in OpenFlow's native features and protocol to support QoS [8, 21, 22, 23]. Bhattacharya and Das [24] introduce an embedded module to reduce communication latency between switch and controller by caching flows, however a data plane QoS mechanism is not proposed.

The mechanism applies a hierarchical approach in classifying traffic classes providing higher positions of priority for mission-critical types of data. This provides flexibility for defining relative QoS relationship between different classes of traffic. The meter band rate determines the point of application using constant 32-bit values specified either in packets/sec or kb/s which can be updated by OFPT_METER_MOD message. This constant inhibits the application of a more dynamic and QoS responsive approach because it fails to account for the co-existence of flows within the wider networks. This solution addresses this issue by specifying band rates of slices relative to multiple slices in the wider network. The issue of constant rate limiting in existing meters and per-port queues is addressed. Furthermore, the results validate performance of this new mechanism to provide higher levels of QoS across the four parameters measured.

In comparison with other solutions in the literature this design offers some distinct contributions. Civanlar et al. [27] aim to provide QoS guarantee to video services similar to the proposed architecture in this solution, however their work is limited to selecting best route for QoS traffic. Minimal detail is provided on how to prioritize the QoS traffic in the SDN infrastructure layer, except for the assumption that QoS traffic can have the right to disrupt non-QoS traffic when they are placed on the same route. The primary focus of this work is on overall best path selection therefore how traffic is processed at each hop is not considered. Contrastingly the proposed solution in this paper focuses on this area and provides per-hop processing with the help of the four physical components.

In [28] video traffic is divided into two classes (lossless and lossy) accompanied by two types of routing algorithms generating different types of flows. The number of traffic classes is shown to have significant impact on overall video performance. From this it can be concluded that there is an advantage to flexibility in the number of classes that can be implemented. The proposed algorithm is flexible enough to accommodate any number of traffic classes as in BDL there is no limitation to the number of meters. Therefore, the network administrator is free to design the system with his or her preferred number of classes.

In advancing OpenQoS Egilmez et al. [29] extend the OpenFlow controller to provide QoS for multimedia traffic. In this architecture the call admission process is relied on to block incoming traffic when there is insufficient resource available in the network to guarantee the requested QoS. However, this could deny admission to voice traffic although possible to guarantee QoS for this type of traffic at the expense of quality for non-priority traffic. The protocol proposed here does not suffer from this limitation of OpenQoS as there is no call admission process. QoS is guaranteed by using OpenFlow meters and the band evaluation engine.

Certain limitations are acknowledged in relation to the proposed solution. The maintenance of network state information is increasingly complex under the dynamic character of the network typology and changing route behaviour. Established routing paths can be disrupted even while data is being transferred entailing, for QoS, necessary maintenance and development of paths in ad hoc networks that have minimal routing overhead and delay [30]. Mobile nodes may further be constrained by limited power supply in comparison with the nodes of wired networks.

Ensuring QoS could mean that, as a result of overheads, more power is used that rapidly drains that available to the node [30]. Scalability of this solution may also be impacted by the computational cost associated with the centralized algorithm in SDN. As networks become larger in size, they become more expensive and difficult to maintain requiring greater storage and more powerful infrastructure [31]. Finally increased computational time on systems in the network can impact negatively on the energy consumption of user devices [31].

9. Conclusion

These findings contribute a novel Meter Band Rate Evaluation mechanism that extends the native capability of OpenFlow and OpenDaylight to enhance the efficiency of QoS provision. The focus of this research was on addressing constant rate limiting with existing QoS mechanisms by extending the capability of OpenFlow to support efficient QoS provision. The introduction of a Meter Band Rate mechanism extends knowledge and suggests new possibilities in developing the existing capabilities of OpenFlow and OpenDaylight to support QoS. Experiment results indicated a significant increase in video streaming and multimedia traffic for multiple QoS parameters and point to the scope of the meter band rate mechanism for optimization of QoS in OpenFlow. The evaluator represents the novel component of the design based on a band rate description language (BRDL) for analysing required dynamic states. This allows specification of desired QoS behavior in a flexible and efficient manner and there is no unnecessary bandwidth restriction for any traffic. Further research in this area may subject the mechanisms to further testing and validation across more complex networking scenarios and hybrid networks. In addition, this may address a comprehensive range of optimization strategies to establish performance and validation in relation to load balancing, energy-consumption, controllability of network flow, safe updating or network security.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The research has been supported and sponsored by the Ministry of Interior in the United Arab Emirates as part of an international research programme.

References

- [1] A. Hakiri, and P. Berthou, "Leveraging SDN for the 5G networks: Trends, prospects and challenges," 2015, <http://arxiv.org/abs/1506.02876>
- [2] D.L.C Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end QoS based on multi-paths routing using SDN technology," in GLOBECOM IEEE Global Comms Conf., 2017, pp. 1-6.
- [3] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang, "Meridian: An SDN platform for cloud network services," *IEEE Comm. Magazine*, vol 51, pp. 120-127, 2013.
- [4] J. Dunlop, D. Girma, and J. Irvine, *Digital Mobile Communications and the TETRA System*, John Wiley & Sons, 2013.
- [5] R. Blom, P. de Bruin, J. Eman, M. Folke, H. Hannu, M. Näslund, M. Stålnacke, and P. Synnergren, "Public safety communication using commercial cellular technology," in *IEEE Next Gen Mobile Applications, Services and Technologies, NGMAST'08.*, 2008, pp. 291-296.
- [6] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol 72, pp.74-98, 2014.

- [7] O. Coker, and S. Azodolmolky, *Software-Defined Networking with OpenFlow*, Packt Publishing, 2017.
- [8] J. Shi, and S.H. Chung, "A traffic-aware quality-of-service control mechanism for software-defined networking-based virtualized networks," *Int. J. Dist. Sensor Networks*, vol 13, pp.1-13, 2017.
- [9] J. Huang, P.J. Wan, and D.Z. Du, "Criticality-and QoS-based multi-resource negotiation and adaptation," *Real-Time Systems*, vol 15, pp. 249-273, 1998.
- [10] T. Reza, *Learning OpenDaylight: A Gateway to SDN (Software Defined Networking) and NFV (Network Functions Virtualization) Ecosystem*, Packt Publishing Ltd, 2017.
- [11] H.E. Egilmez, S. Civanlar, and A.M Tekalp, "An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks," *IEEE Trans.on Multimedia*, vol 15, pp. 710–715, 2013.
- [12] M. Sisov, "Building a software-defined networking system with OpenDaylight controller", BSc Thesis, Inf. Techn., Helsinki Metro. Uni. of Applied Sciences, 2016.
- [13] I. Trajkovska, P. Aeschlimann, C. Marti, T.M. Bohnert, and J. Salvachúa, "SDN enabled QoS provision for online streaming services in residential ISP networks," in *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 33-34, 2014.
- [14] S. Jivorasetkul, M. Shimamura, and K. Iida, "End-to-end header compression over software-defined networks: A low latency network architecture," in *4th Int Conf on Intell. Networking and Collab. Sys., (INCoS)*, pp. 493–494, 2012.
- [15] F. Hu, Q. Hao, and K. Bao, K, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Comm. Sur. & Tutor.*, vol 16, pp. 2181-2206, 2014.
- [16] S. Tomovic, N. Prasad, and I. Radusinovic. "SDN control framework for QoS provisioning,". *Telecomms Forum Telfor (TELFOR)*, 22nd. IEEE. pp. 111–114, 2014.
- [17] W.E. Liang, and C.A. Shen, "A high performance media server and QoS routing for SVC streaming based on Software-Defined Networking," in *Int. Conf. on Comput., Networking and Comm., (ICNC)*, pp. 556-560, 2017.
- [18] M.R. Celenlioglu, and H.A. Mantar, "An SDN based intra-domain routing and resource management model," in *IEEE Int. Conf. on Cloud Eng.*, pp. 347-352, 2015.
- [19] A Mirchev, "Survey of concepts for QoS improvements via SDN," in *Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM)*, Summer Semester, Munich, Germany, pp. 33-39, 2015.
- [20] Y. Lu, B. Fu, X. Xi, Z. Zhang, and H. Wu, "An SDN-based flow control mechanism for guaranteeing QoS and maximizing throughput," *Wireless Personal Comm.*, vol 97, pp.417-442, 2017.
- [21] D. D'souza, K.P. Sundharan, S. Lokanath, V. Mittal, L. Perigo, and R. Hagens, *Improving QoS in a Software-Defined Network*, University of Colorado, 2016.
- [22] R. Thenmozhi, T. Preethi, S. Sadhana, V. Shruthi, and B. Yuvarani, "Integrating multimedia services over software defined networking," *Int. Res. J. Eng. and Tech (IRJET)*, vol. 4, 2017.
- [23] L. Han, S. Sun, B. Joo, X. Jin, and S. Han, "QoS-aware routing mechanism in OpenFlow-enabled wireless multimedia sensor networks," *Int. J. Dist. Sensor Networks*, vol 12, p.9378120, 2016.
- [24] B. Bhattacharya, and D. Das, "QoS enhancement using embedded agent in OpenFlow based SDN node," in *Elect., Comp. and Comm Techn. (ICECCT)*, *IEEE Int. Conf. on*, pp. 1-5, 2015.
- [25] Y. Jinyao, Z. Hailong, S. Qianjun, L. Bo, and G. Xiao, "HiQoS: An SDN-based multipath QoS solution," in *Comms, China*, pp. 123–133, 2015.
- [26] I.F. Akyildiz, S.C. Lin, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Computer Networks*, vol 93, pp.66-79, 2015.
- [27] S. Civanlar, M. Parlakisik, A.M. Tekalp, B. Gorkemli, B. Kaytaz, E Onem, "A qos-enabled openflow environment for scalable video streaming," in *IEEE Globecom Workshops*, pp. 351-356, 2010.
- [28] H.E. Egilmez, B. Gorkemli, A.M. Tekalp, and S. Civanlar, "Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing," in *2011 18th IEEE International Conference on Image Processing*, pp. 2241-2244, 2011.
- [29] H.E. Egilmez, S.T. Dane, K.T. Bageci, and A.M.. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks,". in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1-8, 2012.
- [30] N. Kumar, "Mobile ad hoc network: issue and challenges related to QoS and solutions," *Int. J. Eng. Technol. Sci. Res*, vol 4, pp.415-419, 2017.
- [31] M. Masip-Bruina, J. Yannuzzib, A. Domingo-Pascuala, M. Fonteb, E. Curadob, F. Monteirob, "Research challenges in QoS routing," *Comp. Comms*, vol 29, pp.563-581, 2006.