Multiple Classes of Service Provisioning with Bandwidth and Delay Guarantees in Dynamic Circuit Network

Tananun Orawiwattanakul, Hideki Otsuki, Eiji Kawai, Shinji Shimojo

National Institute of Information and Communications Technology
{tananun, eiji, eiji-ka, sshinji}@nict.go.jp

***Abstract*— A dynamic circuit network (DCN) is a production advance bandwidth reservation service. The majority of DCN operators currently provide a single class of service (CoS), e.g., either a guarantee of bandwidth or no guarantee of quality of service. Although single CoS provisioning is viable and expedient, multiple classes of service (multi-CoS) provisioning offers greatly superior practice, including quality-based charging for a commercial service and efficient resource management. This paper proposes a multi-CoS architecture with bandwidth and delay guarantees in a DCN, more specifically, the deployment of On-Demand Secure Circuits and Advance Reservation System (OSCARS) over multiprotocol label switching (MPLS) networks. The main contribution of this paper is a guaranteed bandwidth-and-delay class where the system can compute the path which can satisfy the given bandwidth and delay constraints. Our system was tested on practical routers.**

***Keywords*— dynamic circuit network (DCN), bandwidth on demand (BoD), bandwidth and delay guarantees**

I. INTRODUCTION

The emergence of software-defined network solutions challenges network operators to leverage revenues by introducing service innovation and new business models. A dynamic circuit network (DCN) is an advance bandwidth provisioning service in which the network has a tool for users, either human or software applications, to automate the creation of virtual circuits (VCs) in advance. In most production DCN services, multiprotocol label switching (MPLS) is used as a transport technology instead of Openflow. This is because providing a DCN service on MPLS requires only the additional control plane (a DCN controller), and it retains the networks that have been invested and widely deployed. In addition, MPLS benefits include carrier-grade quality of service (QoS) provisioning, better security and survivability.

On-Demand Secure Circuits and Advance Reservation System (OSCARS) software [1] is an open-source product developed by the Energy Science Network (ESnet) [2]. Many academic network providers, e.g., ESnet and JGN-X [3], currently use OSCARS to provide a L2/L3 (layers-2 and -3) DCN service over MPLS networks.

JGN-X is a Japanese network testbed for new-generation network developments. Although most of the network operators, including JGN-X, overprovision network resources, many of them may observe high bursts of up to 80%–90% of link capacities because of a few high-rate flows from scientific computing applications. Several JGN-X services, i.e., DCN, Research Infrastructure for large-Scale network Experiments (RISE) (Openflow Networks) [4], and General Integrated Network Engineering Workbox (GINEW) (a VC provisioning system) [5], seamlessly use resources in the L2/L3 backbone networks in a best-effort manner, and the infrastructure resources can be oversubscribed.

The link utilization report for a 10 Gbps interface in the JGN-X core routers at Sendai connecting to Tokyo shows that the average 5 min incoming and outgoing traffic reached 7.362 Gbps and 9.662 Gbps, respectively, in May 2014. The data transmitted by JGN-X include both research experiments and in-service traffic. When network congestion occurs, in-service multi-media flows such as video streaming may experience high delays beyond the delay tolerance of applications. Multiple classes of service (Multi-CoS) provisioning is required for more efficient resource management and enriched services, such as, pay for QoS in a commercial service.

This paper proposes a multi-CoS architecture in OSCARS v.6. We select OSCARS and MPLS for our development, because they are widely deployed in academic communities. In addition, OSCARS has the extensive set of capabilities, e.g., an authentication system, for a production stage. The major contribution of this paper is a guaranteed bandwidth-and-delay class. This paper uses a combination of call admission control (CAC), hard-policing and scheduling disciplines to guarantee bandwidth and delay. The experiments were conducted on MPLS-based routers to estimate the queuing delay. If the propagation delay of links can be known, then, the upper average delay bound of links can be estimated in advance and used as a traffic-metric contained in topology data. We enhanced the path computation elements (PCEs) in OSCARS to compute the minimum delay path and determine whether the upper bound on the average delay of the end-to-end link can satisfy the user's delay constraint. The experiments were performed to observe propagation delays in JGN-X as preliminary results.

Section II presents related work. Section III shows the architecture of our extended OSCARS for multi-CoS provisioning (the control layer). Section IV describes the QoS mechanisms in an infrastructure layer and the experiments. Section V presents the conclusion and future work.

II. Related Work

Several techniques have been proposed for QoS provisioning in bandwidth reservation systems. In the Science Information NETwork (SINET), the layer 1 bandwidth-on-demand (BoD) service is provided on an optical network, and the estimated delay can be computed and notified to users [6]. However, most DCN providers supply L2/L3 on-demand circuits.

In [7], the Application-Layer Traffic Optimization (ALTO)–based virtual private network (VPN) topology manager gathers bandwidth information from Intermediate System to Intermediate System (ISIS)-Traffic Engineering (TE) and Resource Reservation Protocol (RSVP)-TE, and delay information by using the Operation and Maintenance (OAM) ping measurements through a network management system. This real-time information is used to find a path which can satisfy constraints given by the user. The real-time mechanism is efficient for immediate requests, but it cannot commit QoS guarantees of data transmission in the future time.

A resource-pooling mechanism in MPLS-transport profile (TP) networks was proposed in [8] to provide an on-demand VC service. The resource pool composed of label-switched paths (LSPs) and pseudo wires was set in advance, and the unused pooled resource that meets the user's demand is assigned to the request. Nevertheless, the study in [8] does not include multi-Cos provisioning and it may not be flexible for providing multiple path computation algorithms.

The Hybrid Network Traffic Engineering System was proposed in [9]. Traffic data in ESnet routers were collected and analyzed to identify IP addresses of high-rate flows, and then the ingress routers were configured to route these high-rate flows to TE QoS-controlled path. Since OSCARS performs TE, it is used to compute and create the QoS-controlled path.

The effects of different scheduling and bandwidth-policing schemes were studied in [10] to achieve high throughput of high-rate Transmission Control Protocol (TCP) flows while to reduce delay and jitter of real-time sensitive flows. Two soft-policing schemes for handling the traffic over the user's requested bandwidth *BW* were studied, i.e., reclassifying excess packets to a (third) scavenger-service (SS) queue and marking excess packets (modifying a packet's packet loss priority) to influence drop behavior by a weighted random early detection (WRED) mechanism. The latter policing scheme results in a better throughput when compared to the former one, because redirecting packet to different queues results in out of packet sequence at the receiver.

A current service contract of a DCN service in JGN-X is based on hard-policing (excess traffic is immediately dropped). The hard-policing scheme was not studied in [10], but it can be expected that the throughput of TCP flows based on hard-policing will be lower than that based on soft-policing, because more packets are dropped in hard-policing, and this triggers TCP's fast retransmit/fast recovery scheme. However, soft-policing is not suitable for strict guarantee of delay, because packets may gain long delay in the lower priority SS queue, and the buffer occupancy builds up in the case of marking/WRED. In contrast, the queuing delay when implementing CAC with hard-policing is expected to be lower than that with soft-policing because packets are routed to a single queue and their input rate can be controlled. Hard-policing is used in this paper. Our proposal commits a QoS level for a request-flow at the time a user requests a circuit, even if the actual transmission is scheduled for the future.

III. Multi-CoS Provisioning in the Control Layer

This section describes the high-level architecture of our extended OSCARS shown in Fig. 1. The latest version and details of conventional OSCARS can be found in [1]. OSCARS consists of several web service modules, and this paper classifies them into five main groups: User Manager (UM), Computing Resource Manager (CRM), Device Driver, Coordinator and database. The UM performs authentication and authorization functions. The Coordinator handles workflow process of a request between different modules. The CRM is responsible for path computation, and it consists of several modules, e.g., resource manager, topology manger, and PCEs. Note that the PCE stack consists of several sub-PCEs. For simplicity, we use the term CRM to describe functions of those modules. The database keeps user information and tracks used resources, e.g., the reserved bandwidth and time and the assigned virtual local area network identifiers (VLAN ids) in all the links, so that the CRM knows the remaining resources at any given time. The Device Driver communicates with routers to set up and tear down VCs.

A. CoS Definition

In this paper, the CoS policies are a set of QoS levels of data transmission defined by an administrator as follows:

(1) No guarantee of QoS (*NG-QoS*): Resources in the network are shared in a best-effort manner. The path with the least hop count is assigned.

(2) Guaranteed bandwidth (*GBw*): The user's requested bandwidth (*BW*) is guaranteed, and the path with the least hop count is assigned. *BW* refers to the average sending rate (see section IV for more details).

(3) Guaranteed bandwidth with minimum delay (loose) (*GBw-Dl-L*): The *BW* is guaranteed, and the path with the minimum network delay is computed.

(4) Guaranteed bandwidth with minimum delay (strict) (*GBw-Dl-S)*: The *BW* is guaranteed. The user supplies a network delay constraint, denoted as *Dl*. If the minimum delay path can satisfy *Dl*, a request is committed. Otherwise, it is denied.

Note that the term "guaranteed-QoS (*G-QoS*) request" refers to a request with CoS = "*GBw,*" "*GBw-Dl-L*," or "*GBw-Dl-S*."

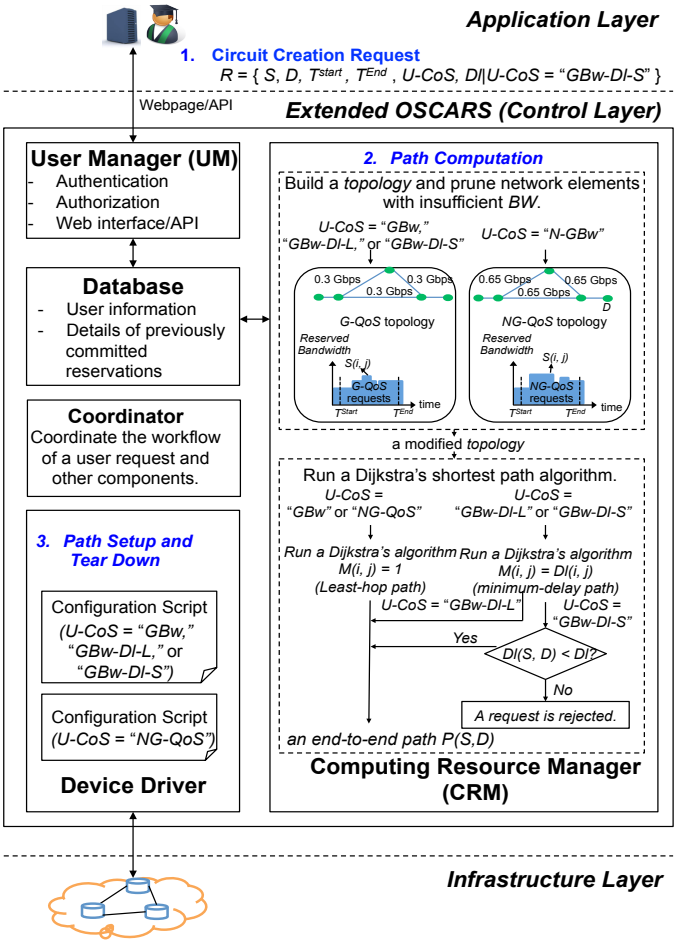

Fig. 1. Extended architecture of OSCARS v.6.

B. Circuit Request Process in Multi-CoS Provisioning

The three main steps in the process of circuit creation (Fig. 1) in our proposal are as follows.

1. Circuit Creation Request: An authenticated and authorized user selects a demanded CoS when providing circuit creation request information through either a webpage or an application programming interface (API). Let *U-CoS* denote the user's selected CoS. The request R can be described as R = (source (S), destination (D), amount of bandwidth (BW), start time (t^{Start}), end time (t^{End}), *U-CoS, Dl* (if *U-CoS* = *"GBw-Dl-S"*)). *U-CoS* is added to the *OptionalConstraint* attributes (an optional field for adding new fields to a request.)

2. Path Computation: The CRM is responsible for computing a path that satisfies *U-CoS*.

2.1 The CRM builds a *Topology*. Topology data contain the topology of the network, the traffic metric of links, the link capacities and a range of VLAN ids. Two class-based virtual topologies, i.e., *G-QoS* and *NG-QoS*, are predefined by the administrator, and they are selected as a *Topology* for *G-QoS* and *NG-QoS* requests, respectively. In this paper, the topology layouts of both classes are the same, but the amounts of link bandwidth differ. Let $l(i,j)$ and $C(i,j)$ denote the link connecting nodes i and j and its bandwidth capacity, respectively.

2.2 The CRM performs CAC by pruning the links that have insufficient bandwidth for BW during the requested transmission period (from t^{Start} to t^{End}) from the *Topology*. The available bandwidth and maximum previously reserved bandwidth of $l(i,j)$ during the requested transmission period are denoted as $A(i,j)$ and $S(i,j)$, respectively, and $A(i,j) = C(i,j) - S(i,j)$. $C(i,j)$ are in the *Topology* data. The previously reserved bandwidth of $l(i,j)$ is time dependent, and $S(i,j)$ can be calculated from the details of the assigned paths and the reserved bandwidth of the connections listed in the database. Next, $l(i,j)$ is pruned from the *Topology* if $A(i,j) < BW$. As shown in Fig. 1, path computation of *G-QoS* and *NG-QoS* requests is independent because of using different virtual topologies. In addition, the previously committed *G-QoS* and *NG-QoS* requests are determined for the calculation of $S(i,j)$ for new arrival *G-QoS* and *NG-QoS* requests, respectively.

2.3 The links with insufficient VLAN ID are also pruned out from the *Topology*. At this stage, all the links in the *Topology* have sufficient BW. Because the QoS is committed at the circuit requesting time for future data transmission, an offline path computation based on the maximum network delay is used to find the path that satisfies the delay constraint. Let $Dl(i,j)$ denote the maximum of the average delay of $l(i,j)$. Let $M(i,j)$ denote a traffic metric of $l(i,j)$ used to calculate the Dijkstra algorithm. $M(i,j)$ is contained in topology data, and it can be the hop count, link utilization, path speed, or delay. The multicost Dijkstra PCE is proposed, and multiple traffic metrics, i.e., $M_{\text{R}}(i,j)$ (a routing metric) and $Dl(i,j)$ (a delay metric), are used, as follows.

2.4 If *U-CoS* is:

- Either "*GBw*" or "*NG-QoS*," the multicost Dijkstra PCE computes a path using a routing-metric [$M(i,j) = M_{\text{R}}(i,j)$] and running a Dijkstra algorithm. Because the current DCN service in JGN-X is based on the least-hop path computation, a hop count metric is used in this paper, i.e., $M_{\text{R}}(i,j) = 1$.
- "*GBw-Dl-L*," the multicost Dijkstra PCE runs a Dijkstra algorithm using $M(i,j) = Dl(i,j)$. The computed path is that with the minimum delay.
- "*GBw-Dl-S*," the process of path computation is the same as that for *U-CoS* = "*GBw-Dl-L*," coupled with delay guarantee checking. Let $P(S, D)$ denote a set consisting of the end-to-end path from S to D, and $Dl(S,D)$ denote the end-to-end delay of $P(S, D)$. The call will be rejected if $Dl(S,D) > Dl$.

3. Path Setup and Tear Down: The device driver provides several modules for supporting different technologies/platforms, e.g., MPLS or Openflow. The EoMPLSPSS module is used in this paper, and it supports L2/L3 equipment, for example, Cisco, Juniper and Dell routers. The administrator predefined configuration scripts, and the EoMPLSPSS uses these scripts to set up and tear down an MPLS LSP before t^{Start} and at t^{End}, respectively. RSVP is used to set up the MPLS LSP based on the PCEs' calculated path. In our proposal, the traffic for a *G-QoS* request is routed to a *G-QoS* queue in the routers, whereas that for an *NG-QoS* request is routed to an *NG-QoS* queue.

Fig. 2. Single-link network

The packets with certain IP addresses are injected in to a corresponding LSP in an L3 DCN service in OSCARS. Consequently, QoS mechanisms used in this paper are for both L2 and L3 DCN services. Section IV gives more details about the infrastructure setting.

OSCARS can interoperate with the InterDomain Controller Protocol (IDCP) [11] and Network Service Interface (NSI) [12] for interdomain communications. JGN-X currently uses IDCP for interdomain DCN provisioning and plans to deploy NSI in the future. Interdomain QoS provisioning can be achieved by cooperation among providers and standardization efforts. There is currently no interdomain QoS agreement for a DCN service among JGN-X and other providers. Consequently, our proposal focuses mainly on intradomain provisioning. The extended OSCARS automatically assigns one of four CoSs to all arriving interdomain requests according to the administrator configuration. For an interdomain request, which is initiated from our domain, the *GBw-Dl-S* class is not provided, while other classes can be provided only in our intradomain network.

C. Network Delay

In the offline multicost Dijkstra algorithm described in the previous subsection, the maximum of the average delay of links in a network must be estimated in advance for path computation of the *GBw-Dl-L* and *GBw-Dl-S* classes. In this paper, a network delay is defined as the time that a packet travels from an ingress router to an egress router (excluding a delay due to an application, a delay from a transmitter to a router, and a delay from a router to a receiver.) The maximum delay of $l(i,j)$, $Dl(i,j)$, equals the sum of delays associated with processing $[T_p(i, j)]$, queuing $[T_q(i, j)]$, serialization $[T_s(i, j)]$, and propagation $[T_l(i, j)]$:

$$Dl(i,j) = T_p(i,j) + T_q(i,j) + T_s(i,j) + T_l(i,j) \quad [1].$$

T_p is very low in high-speed routers, e.g., in the 4–20 µs range in routers with a hardware-assisted switch [13]. The authors suggested in [13] that the most reasonable T_p in practice should be 25 µs per hop, and we used this value in this paper. T_s equals the packet/link rate. We use the 1514 byte media maximum transfer unit size of a gigabit Ethernet interface for a Juniper MX-80 as the packet size for T_s computation. Therefore, the T_s value for a gigabit Ethernet interface is $(1514\times8)/10^9\times10^{-6} = 12$ µs per hop. The method of determining the queuing delay T_q is presented in Section IV.B. The T_l value of links in our experiments in Section IV was assumed to equal 0 ($T_l = 0$) because the routers were connected by a short optical fiber. The observed T_l for links in the JGN-X network is presented in Section IV.D.

IV. Multi-CoS Provisioning in the Infrastructure Layer

This section presents QoS mechanisms in the infrastructure layer for bandwidth and delay guarantees, a determination of the queuing delay in the *G-QoS* queue, and an observation of delay in the JGN-X network. QoS mechanisms matter only when the networks become congested, and it is difficult to perform a test on a service network. Therefore, we conducted experiments on a Juniper MX-80 router running Junos v.12.3. The logical system capability of an MX-80 router enables a single router to be partitioned into multiple virtual devices with independent processing tasks, and it was used to build a single-link network as shown in Fig. 2. A group of physical ports (gigabit Ethernet interfaces) were assigned to each virtual router, and two virtual routers were connected by a direct optical fiber.

Although different router vendors may use different implementations, today's routers can support similar features to achieve the QoS mechanisms used in the routers in our experiments, as follows.

At an egress queue of an interface, three virtual queues were defined: *G-QoS*, *NG-QoS*, and network control (*NC*). An *NC* queue is used to transmit networking protocols such as routing protocols and hello or keepalive messages. We describe a queue as a set of an assigned transmit rate, a percentage of the buffer size, and a priority. For example, a *G-QoS* queue can be presented as *G-QoS*{0.3 Gbps, 30% of buffer, "strict-high"}. Note that this paper treats the *NG-QoS* class in a best-effort manner, and the *NG-QoS* queue is shared between the *NG-QoS* DCN and other traffic in the network. The *NG-QoS* class can be treated as the lowest priority by redirecting its traffic to the isolated low resources queue, e.g., the SS queue.

A combination of a priority queue (PQ) and a weighted round robin (WRR) scheduler is implemented at an egress queue of an interface. A queue is considered to be *in-profile* when the rate at which packets are drained from it ($Rate_{output}$) is below its allocated transmit rate. The PQ scheduler traverses the sets in descending order of priority. Within a given priority, the in-profile queues are served in a WRR fashion. The next lower priority level is serviced only when all the queues at the current and higher priority levels are empty (when configuration is in a work-conserving mode) or have reached their transmit rate (are out-of-profile.) Note that a strict-high priority is exceptional. The scheduler serves the strict-high queue immediately when it has a packet. In all the experiments, the transmit rate was shared among virtual queues in work-conserving mode; therefore, the out-of-profile queues can use the leftover bandwidth of the other queues. The buffer space was strictly allocated.

The classification of the ingress traffic is a mechanism that assigns which virtual queues packets should be forwarded to. In general, the administrator configures what classifications are used for a certain port, e.g., Ethernet level 802.1p or MPLS experimental bits (EXP)/traffic class (TC) bits for layer 2 ports, or Differentiated Services Code Point (DSCP), Internet

Protocol (IP) precedence or 802.1p for layer 3 ports. A firewall filter and bandwidth policing can be used to overwrite the general classification and classify packets to a certain queue. A firewall filter subjects the traffic of a request-flow to the bandwidth policer. The bandwidth policing based on single-rate two-color (SrTc) is used in this paper to ensure that the traffic of a DCN flow will not exceed its requested bandwidth *BW*, and it over-writes general classification of packets to assign proper queues. The token rate and depth of the bucket in bytes are the same as the settings in JGN-X DCN, i.e., *BW* and (0.1 × *BW* (bits/s)) (bytes), respectively. The policer routes all traffic under *BW* for a *G-QoS* request to a *G-QoS* queue in the egress interfaces, whereas that for an *NG-QoS* request is routed to an *NG-QoS* queue. Traffic over *BW* is dropped, and this policy is called hard-policing. Note that a bandwidth guarantee in this paper refers to a commitment to deliver the average sending rate *BW* measured by the SrTC (packets that pass through SrTc to an egress queue of an interface) to a destination.

The queues and their resource allocation were directly preconfigured in the routers by an administrator, whereas the bandwidth-policing and firewall were established on a request-flow basis by OSCARS through the administrator's prewritten configuration scripts. In all the experiments, the requested transmission times (t^{Start} and t^{End}) of an OSCARS request covered the entire experimental period, and they are not mentioned for simplicity. A ping application was used to measure the queuing delay (in the *G-QoS* and *NG-QoS* queues). The 10 Mbps LSP was created for each ping flow in Sections IV.A, IV.B and IV.C. These LSPs have no policing, and a firewall filter is used to redirect all ping packets to certain queues. The application in our experiments was Iperf v.2.0.5, and it generated user datagram protocol (UDP) data with a 1470 byte datagram size. In Fig. 2, Let S_i and D_i denote the pair of source and destination hosts and s_i and d_i denote their corresponding ports.

A. Buffer size

Objective: To determine whether using a ping application is feasible to estimate the queuing delay T_q by comparing the observed queuing delay when a virtual queue is full with the default value of an interface.

Network scenario: We used a topology as shown in Fig. 2. Two requests were made to OSCARS for UDP traffic flows: (s_1, d_1, 750 Mbps, "*NG-QOS*") and (s_2, d_2, 750 Mbps, "*NG-QOS*"). The 10 Mbps LSP for a ping application from s_3 to d_3, and the traffic of this ping flow was route to the *N-GQoS* queue. In this experiment, a single virtual queue *NG-QoS* was configured on the egress queue of the interfaces from s_4 to d_4, and back. The 100% of link capacity (1 Gbps) was allocated to this queue, while the buffer allocation was strictly partitioned. First, we allocated 100% buffer allocation to the virtual queue. While no other traffic load, the S_3-D_3 ping application sent 60 ping messages, and their average round-trip times (RTT) was 0.281 ms.

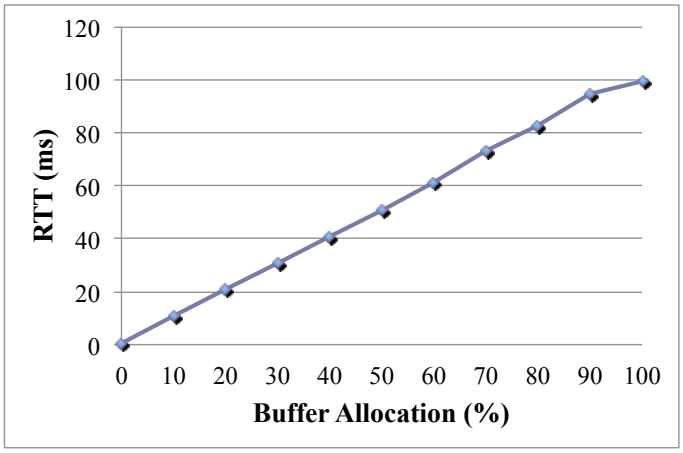

Fig. 3. RTT in single virtual queue assignment.

Next, we varied the buffer allocation from 0% to 100%. The S_1-D_1 and S_2-D_2 UDP flows transmitted data at the rate 750 Mbps per flow, therefore, the whole buffer of a virtual queue was full. The S_3-D_3 ping application sent 60 ping messages.

Results and discussion: Figure 3 shows the average RTT reported by the ping application. Although 0% of buffer-space was configured, in fact, a router allocated a small amount of buffer space to a virtual queue which was full in this experiment, and the RTT equals 0.426 ms. At the 100% buffer allocation, the RTT 99.194 ms could be observed.

Since a Ping packet is small, this paper assumes that the approximated queuing delay T_q at a certain input load equals to the RTT at a certain input load (RTT_T) minus the RTT at when no other input load (RTT_0):

$$T_q = RTT_T - RTT_0 \quad [2].$$

From Eq. [2], our approximated T_q of the full buffer space is 98.913 ms (99.194 – 0.281).

The default delay buffer of the gigabit interface for a Juniper MX-80 equals 100 ms [14], and its corresponding size is 125 MB. Consequently, the first conclusion is a router used in this experiment uses a default buffer size. Next, using the ping application and calculation in Eq. [2] to estimate T_q would give around 1% error.

B. Maximum of Average Queuing Delay in an Excessive Traffic Load Scenario

Objective: In our proposal, the *GBw-Dl-L* and *GBw-Dl-S* classes consider the delay, and the maximum delay of each link must be known in advance, as described in Section III.B. The experiments in this subsection were conducted to estimate and observe the upper bound of the average queuing delay $T_{q\text{-}MAX}$ in the *G-QoS* queue with different queue priority assignments in the over-traffic load scenario.

Network scenario: Figure 4 shows our experimental setup. Resources in the infrastructure layer are associated with those in the virtual topology *G-QoS* in OSCARS, and resource allocation for non-guaranteed services depends on the service policy and network design of an operator.

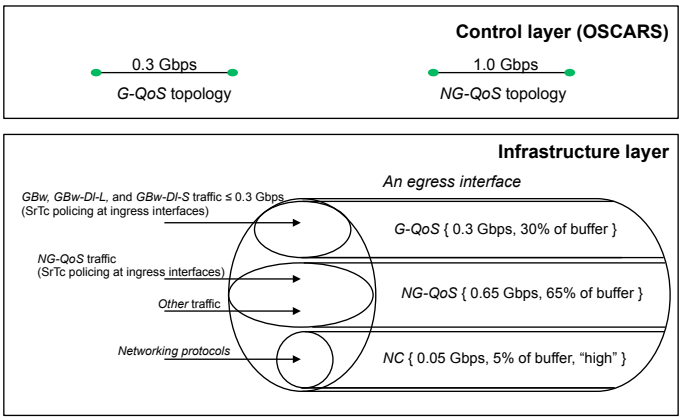

Fig. 4. Virtual topology in OSCARS and queue assignment in router.

In this experiment, the *G-QoS*{0.3 Gbps, 30% of buffer size}, *NG-QoS*{0.65 Gbps, 65% of buffer size} and *NC*{0.5 Gbps, 5% of buffer size, "high"} queues were configured at the egress queue of the interfaces from s_4 to d_4 and back, as shown in Fig. 4. A single link in the *G-QoS* virtual topology in OSCARS (the control plane) has a capacity of 0.3 Gbps (Fig. 4). Therefore, 0.3 Gbps transmit rate was assigned to the *G-QoS* queue. Since we used OSCARS to create *NG-QoS* circuits up to the link capacity in the experiments, we perform overbooking by setting the *NG-QoS* topology in OSCARS to have a 1 Gbps capacity. Only 0.65 Gbps transmit rate was assigned to the *NG-QoS* queue.

In this subsection, we focus on three different scheduling priority assignments of the *G-QoS* and *NG-QoS* queues: 1. strict-high (*G-QoS*) and medium-high (*NG-QoS*), 2. medium-high (*G-QoS*) and low (*NG-QoS*), and 3. low (*NG-QoS*), and low (*G-QoS*) and low (*NG-QoS*). Two requests were made to OSCARS for the following pairs of sources and destinations: (s_1, d_1, 0.3 Gbps, "*GBw*") and (s_2, d_2, 1 Gbps, "*NG-QoS*"). Since our topology was small, there was no difference among *GBw*, *GBw-Dl-L*, and *GBw-Dl-S* classes. Two additional 10 Mbps LSPs for ping applications from s_3 to d_3 were created, and the traffic from these ping flows was routed to the *G-QoS* and *NG-QoS* queues, respectively. First, when there was no other traffic, the RTTs of the s_3-d_3 (*G-QoS*) and s_3-d_3 (*NG-QoS*) ping flows were almost the same, 0.251 ms.

Let $Rate\text{-}In_i$ and $Rate\text{-}Out_i$ denote the rate at which packets enter and are drained from queue i, respectively. If $Rate\text{-}In_i > Rate\text{-}Out_i$, the buffer of queue i finally becomes full. In this case, the maximum of the average queuing delay ($T_{q\text{-}MAX}$) equals the buffer size B divided by $Rate\text{-}Out_i$:

$$T_{q\text{-}MAX} = B/Rate\text{-}Out_i \quad [3].$$

The default buffer size of the gigabit interface for a Juniper MX-80 equals 125 MB. From Eq. [3], the maxima of the average queuing delay $T_{q\text{-}MAX}$ for the *G-QoS* and *NG-QoS* queues are both 100 ms:

$T_{q\text{-}MAX}$(*G-QoS*) = 30%*125 MB/0.3 Gbps = 100 ms [4], and
$T_{q\text{-}MAX}$(*NG-QoS*) = 65%*125 MB/0.65 Gbps = 100 ms [5].

TABLE I.
AVERAGE RTT OF PING PACKET PASSING THROUGH *G-QoS* AND *NG-QoS* QUEUES IN EXCESSIVE TRAFFIC LOAD SCENARIO

Priority Assignment (*G-QoS*:*NG-QoS* queues)	Average RTT (*G-QoS*) (ms)	Average RTT (*NG-QoS*) (ms)
Strict-High:Medium-High	0.391	95.117
Medium-High:Low	2.274	95.692
Low:Low	2.218	95.723

However, the traffic entering the *G-QoS* was controlled by the CAC in OSCARS and SrTc at the ingress interfaces. To determine the actual $T_{q\text{-}MAX}$(*G-QoS*), applications were executed to simulate the excessive traffic load scenario as follows. S_1 ("*GBw*") sent UDP data at 0.5 Gbps to D_1, and only 0.3 Gbps were passed through the SrTc to the *G-QoS* queue. The S_2-D_2 UDP flows ("*NG-QoS*") sent data at 1 Gbps; therefore, the *NG-QoS* buffer was full. All the UDP flows transmitted data for 120 s. At 60 s, two ping applications on S_3 sent 60 packets to D_3 through each of the *G-QoS* and *NG-QoS* queues.

Results and discussion: Table I shows the average RTT of a ping application through the *G-QoS* and *NG-QoS* queues. In this scenario, the sending rate of the *G-QoS* flow equaled 0.5 Gbps, and only 0.3 Gbps passed through the SrTc to the *G-QoS* queue. However, the scheduler and SrTc use different algorithms to measure the input traffic. The 0.3 Gbps traffic that pass through the SrTC occasionally made the *G-QoS* queue become out-of-profile based on the PQ/WRR of the scheduler. The "strict-high" priority setting gave superior delay results (0.391 ms), because the PQ scheduler serviced the strict-high queue immediately when it had a packet. Implementing a strict-high priority for OSCARS guarantee service does not starve the lower-priority queue as long as the assigned transmit rate of the *G-QoS* queue is allocated in agreement with the *G-QoS topology* maintained by OSCARS.

Excluding the strict-high priority, when the *G-QoS* queue is assigned a higher priority than the *NG-QoS* queues, the PQ scheduler switched to the *NG-QoS* queue when the *G-QoS* queue became out-of-profile. However, the time when the status of the *G-QoS* queue became out-of-profile was not long enough to observe differences in the delays between the *G-QoS* and *NG-QoS* queues for different priorities: medium-high (*G-QoS*) and low (*NG-QoS*), and low (*G-QoS*) and low (*NG-QoS*).

From Eq. [2], the estimated $T_{q\text{-}MAX}$(*NG-QoS*) was equal to 95.692 – 0.251 ms = 95.441 ms. Although the input traffic load of the *NG-QoS* queue exceeds its allocated transmit rate, the *NG-QoS* queue can use the leftover bandwidth of the *NC* queue. Therefore, the average queuing delay of the *NG-QoS* queue when the buffer is full was smaller than the maximum delay (100 ms), as calculated in Eq. [5]. The estimated $T_{q\text{-}MAX}$(*G-QoS*) values with the priorities strict-high and low were 0.14 ms (0.391 – 0.251) and 1.967 ms (2.218 – 0. 251), respectively.

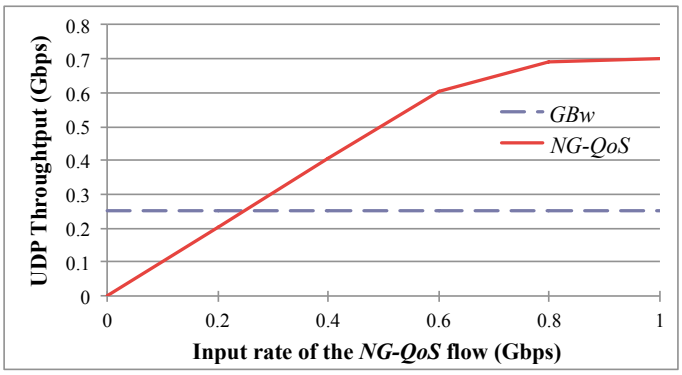

Fig. 5. UDP throughput of DCN flows with different CoSs.

TABLE II.
AVERAGE RTT OF PING PACKET PASSING THROUGH *G-QOS* AND *NG-QOS* QUEUES IN VARIOUS TRAFFIC LOAD SCENARIO

Input rate of *NG-QoS* flow (Gbps)	Average RTT of ping packets (ms)			
	G-QoS (strict-high)	*NG-QoS* (medium-high)	*G-QoS* (low)	*NG-QoS* (low)
0	0.25	0.246	0.238	0.252
0.2	0.278	0.27	0.269	0.275
0.4	0.322	0.388	0.33	0.394
0.6	0.36	0.764	0.361	0.653
0.8	0.376	90.092	0.395	90.002
1	0.387	90.042	0.398	90.086

According to Eq. [1], the maximum of the average delay of $l(i,j)$ for *G-QoS* flows is $Dl(i,j) = T_p(i,j) + T_{q\text{-}MAX}(i,j) + T_s(i,j) + T_l(i,j)$, where $T_p(i,j) = 25$ µs, $T_s(i,j) = 12$ µs. $T_{q\text{-}MAX}$ can be approximated from the experiments in the overload traffic scenario in this subsection as 0.14 ms (strict-high) and 1.967 ms (low). Although the estimated $T_{q\text{-}MAX}$ in our experiments would give around only 1% error (Section IV.A), $T_{q\text{-}MAX}$ should be rounded up in practical implementation to compensate other errors, e.g., when using different router models. $T_{q\text{-}MAX}$ between 5.0-10.0 ms is acceptable for using in JGN-X. Since $T_{q\text{-}MAX}$ was estimated in this sub-section, if the propagation delay $T_l(i,j)$ of links is known, $Dl(i,j)$ (a delay metric) can be computed in advance and added in topology data for path computation in the multicost Dijkstra PCE.

C. Average Queuing Delay in Various Traffic Load Scenario

Objective: The goals of the experiments in this subsection are as follows: 1. To verify that the requested *BW* can be guaranteed for *G-QoS* DCN flows during network congestion and that their unused bandwidth can be used by *NG-QoS* DCN and other traffic flows. 2. To observe the average queuing delay $T_{q\text{-}MAX}$ in the *G-QoS* queue with different queue priority assignments under various traffic loads.

Network scenario: The topology, queue assignment, and circuits created by OSCARS in this subsection were the same as those in the previous subsection. S_1 ("*GBw*") sent UDP data at 0.25 Gbps to D_1 for all executions. The input traffic of the S_2-D_2 UDP flow ("*NG-QoS*") was varied from 0 to 1 Gbps for each execution. Both UDP flows transmitted data for 120 s. At 60 s, two ping applications on S_3 sent 60 packets to D_3 through each of the *G-QoS* and *NG-QoS* queues. In the previous subsection, the delay results (medium-high (*G-QoS*) and low (*NG-QoS*), and low (*G-QoS*) and low (*NG-QoS*)) were almost the same. Therefore, only two scheduling priority assignments: strict-high (*G-QoS*) and medium-high (*NG-QoS*), and low (*G-QoS*) and low (*NG-QoS*) were interested in this subsection.

Results and discussion: Figure 5 shows the UDP throughput of the DCN flows with different CoSs. Based on our scenarios, the different priority assignments had no effects on the throughput. Therefore, the UDP throughputs of *GBw* and those of *NG-QoS* with different queue priorities were the same. The throughputs of the *GBw* were at their sending rates because a delivery of traffic below their requested *BW* was guaranteed. When the sending rate of the *NG-QoS* flow was below the allocated transmit rate of the *NG-QoS* queue (0.65 Gbps), they received a throughput at their sending rate. When the sending rate of the *NG-QoS* flow equaled 0.8 and 1 Gbps, its throughput was approximately 0.7 Gbps because it could utilize the unused bandwidth of the *G-QoS* and *NC* queues. Note that the bandwidth report of Iperf is a UDP datagram rate that excludes overheard, e.g., IP, Ethernet, and MPLS headers. Therefore, when input rate of the *NG-QoS* flow greater than or equal to 0.8 Gbps in Fig. 5, the maximum throughput of all the flows per link equaled approximately 0.95 Gbps (not 1 Gbps).

Table II lists the average RTT of ping packets through the *G-QoS* and *NG-QoS* queues. The average RTTs of ping packets in the *G-QoS* queue with strict-high and low priorities in this subsection are almost the same. This is because $Rate\text{-}In_{G\text{-}QoS} < Rate\text{-}Out_{G\text{-}QoS}$ for the entire execution time. When the input rate of each *NG-QoS* flow was below the allocated transmit rate of the *NG-QoS* queue (0.65 Gbps), the average RTT of ping packets in this queue was low. When the buffer of the *NG-QoS* queue was full, the average RTT of ping packets through the queue was approximately 90 s. In this subsection, the *NG-QoS* queue could utilize the unused bandwidth of both the *NC* and *G-QoS* queues. Therefore, the average RTT of ping packets through the *NG-QoS* queue in this subsection was lower than that in the previous subsection.

Since the un-used bandwidth of the *G-QoS* class can be utilized by the *NG-QoS* flows, the QoS mechanisms in this paper efficiently use resources. In this subsection, the queuing delay is relatively low, because the input-rate of the *G-QoS* queue is lower than the queue's allocated transmit rate. In practice implementation, the $T_{q\text{-}MAX}$ results in the overload traffic environment should be used as described in the previous subsection.

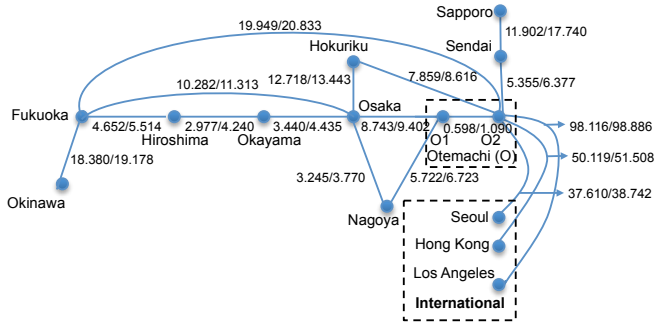

Fig. 6. Minimum/average RTT of ping packets in JGN-X networks.

D. Propagation Delay

Objective: The goal of this subsection is to observe the characteristics of delays in the JGN-X networks using a ping application as preliminary results and to discuss on a feasibility to implement a proposal on the JGN-X network.

Network scenario: Figure 6 shows part of the JGN-X domestic topology. Experiments were performed to simply observe T_l in the JGN-X network. 200-ping packets were generated by JGN-X routers (Juniper MX series) to their peer connecting routers (including international links) for each execution. The ping packets were routed through a best-effort queue which is used for data transmission, and processed by the routers. The executions were done once a day at 14:00–18:00 from July 16th–22nd, 2014 (7 records), and more executions were done at other times (13 records).

Results and discussion: Figure 6 shows the RTT (minimum/average) of ping packets on JGN-X networks and international links on July 16th, 2014 at 19:00. We consider the minimum delay because it is closest to the propagation delay of links. For simplicity, it is assumed that the propagation delay of the link $l(i, j)$ is $T_p(i, j)$ = (RTT/2). All the RTT results were analyzed, and there was no significant variation in the observed data. This is because JGN-X overprovisions its network. High burstiness of up to 80%–90% of link capacity occasionally occurred owing to research experiments and academic events, and a high traffic load was not observed during our observation period. Note that a method of measuring the one-way propagation delay of links is required for practical implementation in the future, for example, the One-Way Active Measurement Protocol (OWAMP) [15].

Finally, if the propagation delay of links can be known, an estimated delay metric can be computed and added in topology data for use in path computation. From the results in Section IV.B, although the strict-high priority assignment to the *G-QoS* queue gives superior delay results in the excessive traffic load, in our proposal, a strict-high priority is not recommended. This is because a combination of CAC and policing (SrTc) results in a low queuing delay even the priorities of the *G-QoS* and *NG-QoS* queues are the same, and implementing a strict-high queue may starve other queues if the rate limit of the queue is not well controlled by the administrator. Because the queuing delay of *G-QoS* flows can be controlled, the propagation delay of links plays an important role in delay guarantees for a *G-QoS* DCN. The propagation delay of links in JGN-X was low. The delay of the longest hop (Sapporo-Okinawa) was lower than the delay threshold of multimedia applications, e.g., 250-300 ms. The path computed on the basis of a hop count metric is the same as that computed on the basis of a delay metric in JGN-X, because a shorter hop results in a lower propagation delay in our observation.

Consequently, only three classes should be implemented in JGN-X: *NG-QoS*, *GBw-Dl-L*, and *GBw-Dl-S*. However, it is worth including the *G-BW* class in a proposal for other operators who use other routing metrics.

V. Conclusion

The goal of this paper is multi-CoS provisioning in an intradomain DCN service in which the user can specify QoS constraints, i.e., bandwidth and delay guarantees, and the QoS level is committed for a request-flow when a request is made for transmission in the future. Although guarantee of delay in a practice network is a challenge, the results in the paper show that if proper QoS mechanisms, e.g., a combination of CAC in a DCN controller, hard-policing, and the scheduling disciplines, are implemented, guarantee of bandwidth and delay can be achieved in an intra-domain L2/L3 DCN service. However, interdomain QoS provisioning requires an effort on standard developments and cooperation among operators in the future. Future plans include an extension of OSCARS to provide multiple bandwidth reservation services with independent virtual topologies, resource allocation, and different QoS levels. User authorization is used to control the eligibility to access a given service.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. They are also grateful to Mr. Yoshiyuki Ota for his developed tool for delay measurement, and Mr. Tomonori Wakabayashi, Mr. Nakamura Kazuhiko, Mr. Jin Tanaka and all members of the JGN-X network operations center for their kind help.

References

[1] On-Demand Secure Circuits and Advance Reservation System (OSCARS). Retrieved: 19.1.2015. [Online]. Available: http://www.es.net/engineering-services/oscars/

[2] Energy Science Network (ESnet). Retrieved: 19.1.2015. [Online]. Available: http://www.es.net/

[3] JGN-X. Retrieved: 19.1.2015. [Online]. Available: http://www.jgn.nict.go.jp/english/

[4] Research Infrastructure for large-Scale network Experiments (RISE). Retrieved: 19.1.2015. [Online]. Available: http://www.jgn.nict.go.jp/rise/english/index.html.

[5] General Integrated Network Engineering Workbox (GINEW). Retrieved: 19.1.2015. [Online]. Available: http://wp.ginew.net.

[6] S. Urushidani, K. Fukuda, Y. Ji, S. Abe, M. Koibuchi, M. Nakamura, S. Yamada, K. Shimizu, R. Hayashi, I. Inoue, and K. Shiomoto, "Resource Allocation and Provision for Bandwidth/Networks on Demand in SINET3," in Proceedings of the IEEE Network Operations and Management Symposium Workshops (NOMS), 2008, pp. 212 – 218.

[7] M. Scharf, V. Gurbani, T. Voith, M. Stein, W. Roome, G. Soprovich, and V. Hilt, "Dynamic VPN optimization by ALTO guidance," in Proceedings of the 2nd European Workshop on Soft-ware Defined Networks (EWSDN), 2013, pp. 13–18.

[8] T. Iijima, T. Suzuki, K. Sakamoto, H. Inouchi, and A. Takase, "Applying a Resource-pooling Mechanism to MPLS-TP Networks to Achieve Service Agility," in Proceedings of the 5th International Conference on Cloud Computing, GRIEs, and Virtualization (CLOUD COMPUTING), 2014, pp. 31-36.

[9] Z. Yan, C. Tracy, and M. Veeraraghavan, "A hybrid network traffic engineering system," in Proceedings of the IEEE 13th High Performance Switching and Routing (HPSR), 2012.

[10] Z. Yan, M. Veeraraghavan, C. Tracy, and C. Guok, "On How to Provision Virtual Circuits for Network-Redirected Large-Sized, High-Rate Flows," in Proceedings of the International Journal on Advances in Internet Technology, vol .6, no. 3&4, 2013.

[11] InterDomain Controller Protocol (IDCP). Retrieved: 19.1.2015. [Online]. Available: http://www.controlplane.net.

[12] Network Services Interface (NSI) working group, Open Grid Forum. Retrieved: 19.1.2015. [Online]. Available: https://redmine.ogf.org/projects/nsi-wg

[13] "Design Best Practices for Latency Optimization," Financial Services Technical Decision Maker White Paper, Cisco Systems, Inc.

[14] D. R. Hanks Jr. and H Reynolds, "Juniper MX Series, A Comprehensive Guide to Trio Technologies on the MX," California, O'Reilly, 1st Edition, October 2012.

[15] One-Way Active Measurement Protocol (OWAMP). Retrieved: 19.1.2015. [Online]. Available: http://software.internet2.edu/owamp/