Intelligent Route Control and QoS provisioning at the Internet eXchange Point

1. ABSTRACT

Internet eXchange Points (IXPs) of today are infrastructure providers for Autonomous Systems (ASes) that peer to exchange traffic. Thus, so far, participating ASes have been vested with the responsibility to administer efficient utilization and smart management of their traffic over the peering links. Lacking a holistic and dynamic view of the local peering network, and limited by the inadequate features of Border Gateway Protocol (BGP), participants are severely handicapped in taking full advantage of the immensely dense peering network of today's IXPs. In this paper, we present a new operational model for IXPs that leverages the Software Defined eXchange (SDX) IXP platform to deliver an intelligent, quality-aware route control service to participants. This model passively monitors traffic and link utilization, and actively measures performance of advertised routes to deliver a constraint-based QoS routing and network aware load balancing to IXP participants. We demonstrate this model with two experiments done on a one-switch IXP simulation: **a)** TCP bandwidth tests showing that intelligent route control avoids overburdening of a single link, thus improving average end-to-end TCP throughput and **b)** video streaming over the IXP showing that QoS routing of this model reduces frame corruption.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms

Design, Performance, Measurement

Keywords

SDN, IXP, QoS

2. INTRODUCTION

Over the past twenty years, Internet eXchange Points (IXPs) have evolved from providing just the bare necessities for interconnection among participating ASes (e.g. physical space, power, cooling, security) to providing a variety of services (e.g. Route Server, remote peering, support for IXP resellers) possible only through innovative use of advancements in networking technology. Due to explosive growth in the internet for the past two decades, the largest IXPs today, such as AMS-IX and DE-CIX have each grown to host more than 500 participants and carry similar amounts of traffic as some of the largest ISPs. With more than 300 IXPs worldwide [10], they have become key players in the modern Internet landscape.

As such, IXPs have been attracting much attention from the research community in recent years. The operational models and anatomical details of some of the largest IXPs have been studied [10, 7] and the traffic through them thoroughly analyzed [9]. Proposals for improving Border Gateway Protocol (BGP) communication between participating ASes through Route Server have also been seen [18]. Migrating IXPs to SDN to take full advantage of the data and control plane separation has also been suggested [8].

The scale of a large, modern IXP operation is evident in statistics from AMS-IX and DE-CIX [1, 3], the two largest European IXPs. There are more than 500 participant ASes and 50,000+ active peerings at just one; e.g. [7]. Though Route Servers enable AS administrators to implement sophisticated bilateral and multilateral peering policies [28], without sufficient quality information and automation, these policies are rigid and unnecessarily complex. Moreover, BGP, the interdomain routing protocol at IXPs makes dynamic policy implementations a very cumbersome task: e.g., to account for diurnal cycle fluctuations, network administrators write scripts that log into individual routers and indirectly change configuration [8]. With the formation of a working public peering taking many days, it is impossible for ASes to frequently fine-tune their routing policies and fully exploit the available peering opportunities.

Moreover, BGP is agnostic to performance or QoS metrics. The metrics used by the best path selection algorithm of BGP do not corrleate closely with actual QoS metrics. Experimental results, obtained using a real Internet topology and RTT data, showed that the AS_PATH length metric achieved only a 50% success rate of the trials performed to identify the destinations with smaller round trip time [17]. In fact, 40% to 50% route selections are made using the lowest router ID rule of BGP path selection algorithm [11]. Many solutions have been proposed to the interdomain QoS problem of the internet involving extensions to BGP, overlay networks and other mechanisms but none have achieved widespread adoption due to reasons ranging from impractical cooperation required from independent ASes, to the lack of economic incentive for ASes to deploy such proposals [12].

2.1 Main Contributions

To address these issues, we propose a QoS abstraction for SDX [16] where path selection for the next hop is managed by the IXP to achieve better utilization of available resources and where constraint-based QoS is provided to participants without requiring tedious configuration or cooperation between independent ASes. This paper makes the following major contributions:

- † Propose an intelligent route control mechanism similar to those discussed in [13] at the IXP and show the advantages to using IXP as a vantage point for such a system.
- † Propose a measurement and constraint-based QoS provision system for IXP participants and discuss its merits and limitations.
- † Propose a policy definition mechanism to enable participants to write next-hop agnostic, QoS-aware policies.

† Propose a centralized interface for IXP participants to access route collector functions, check currently implemented forwarding path for their traffic, receive policy conflict and failure updates, and access IXP-wide performance measurements.

2.2 Overview

In the following, we discuss related work in §3, the limitations of the current IXP model in practice in §4, the proposed IXP operational model, measurement techniques and QoS-based resource allocation mechanism in detail in §5, the design and architecture of our pilot implementation, issues such as self-load and path oscillation, and corrective techniques in §6, demonstration of our system in Mininet with a one-switch IXP simulation in §7, potential expansion areas, their impact and future work in §8, aspects of our work that encourage adoption by the network operators community and IXPs in §9 concluding in §10.

3. RELATED WORK

The concepts of abstracting routing decisions out of individual routers and implementing network aware load balancing are discussed in [30]. However, authors do not discuss bringing the function to interdomain routing at the IXP. Similarly, concepts like routing as a service and routing outsourcing with or without SDN are discussed in [19, 22], but the authors do not discuss QoS provision. Intelligent route control mechanisms in multihomed networks, i.e. networks with multiple egress links to the Internet are discussed in [13, 15] but these ideas are not ported to the Internet eXchange Point where they could be more beneficial as we have shown in this work.

QoS extensions to BGP have been proposed in [32, 33]. However, they do not address the issues of advertised QoS being inaccurate or the extensive interdomain cooperation required to implement these mechanisms which is often absent. OverQoS, an overlay network based solution for internet QoS has also been proposed [29], but this solution does not consider path selection in case of multihomed networks and is limited to prioritizing traffic within a single path.

The concept and implementation of Software Defined Networking at IXPs is discussed in [16]. The authors present an IXP architecture where multiple participant ASes can express their policies in the Pyretic programming language [24] along with the traditional BGP. They present a sequential composition technique for combining different policies and also a virtual switch abstraction to prevent policy and traffic interference of different participants. However, authors do not discuss intelligent path selection in the control plane or QoS provision to IXP participants. Control eXchange Point work in [20] proposes pathlet stitching by a central exchange point to provide end-to-end QoS to customers bringing the clear separation of control and forwarding planes in SDN over to interdomain QoS routing. This work also suffers from the issues of pathlet QoS guarantees given by ISPs not being met and the willingness of a large number of domains to participate in this system.

Our work is, to the best of our knowledge, the first to port the concept of intelligent route control to the IXP and to utilize the large number of alternate paths to a given destination usually available at a large IXP to provide QoS routing to its participants.

4. SHORTCOMINGS OF THE CURRENT IXP MODEL

In this section, we discuss the shortcomings of the current IXP model that lead to AS operators facing difficulties in policy implementation and management, and non-optimal utilization of the available resources.

- **No QoS provision or traffic prioritization:** Participants of an IXP have no way of demanding QoS or prioritization of their traffic that goes through an IXP. To achieve desired QoS, they have to establish special peering relationships and Service Level Agreements (SLAs) with other ASes. Although SLAs might be seen as a more reliable way to demand QoS guarantees for certain traffic, they have a significant management overhead in terms of establishing terms of the SLA and continuous monitoring to make sure that they are met.

- **Quality of available paths not known:** There is no mechanism in the current IXP model for a network operator to stay informed of the quality parameters of the links provided by other ASes. The responsibility of path selection is delegated to BGP which only considers some relatively static parameters such as AS_PATH length and Multi-Exit Discriminator (MED). More volatile and indicative parameters of link quality, such as latency, jitter and packet loss rate are ignored. This can lead to, in case of multilateral peering, non-optimal decisions by the Route Server and, in case of bilateral peering, excessive management overhead to stay informed of quality parameters.

- **Traffic engineering not easy:** At a typical IXP, participants maintain a mixture of bilateral and multilateral peering arrangements. With many edge routers installed at the IXP premises, multiple private peers, and the ISP effectively peering with all other participants by connecting to the Route Server, traffic engineering is notoriously difficult. Network operators have to resort to arcane techniques, such as BGP AS_PATH prepending, to implement trivial policy tasks such as inbound traffic engineering across multiple edge routers. As a result, it is hard to modify policies frequently and thus they become rigid and stagnant.

- **Uneven distribution of traffic:** Consider that two paths to reach a given prefix, $p1$ and $p2$ are announced to the Route Server by two participants. Also consider that $p1$ has a shorter AS_PATH length. Applying the BGP best path selection algorithm discussed in [27], the Route Server will select $p1$ as the best path to reach that prefix and announce this route to all other participants. This way, the participant that announced $p1$ will become the next hop for all traffic destined for that prefix. Since there is no check for overburdening of a path, $p1$ may very well get overloaded and congested while another path to the same prefix, $p2$, though not optimal, remains under utilized.

In the coming sections, we will see how our proposed model addresses these issues for constraint-based QoS provision to participants and even traffic distribution for better utilization of available paths.

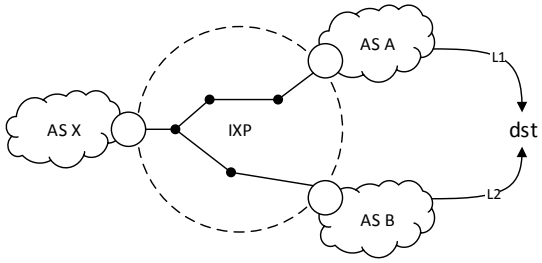

Figure 1: IXP Model

5. OPERATIONAL MODEL

The role of an IXP is to facilitate peering among participating autonomous systems. When two ASes establish a direct BGP session between each other and exchange routes privately, it is called bilateral or private peering. Multilateral peering, by contrast, is when more than two ASes connect to a central Route Server and the Route Server is responsible for route collection and announcement. The Route Server of a large IXP processes a huge number of routes often exceeding the typical 512K capacity of a CISCO router. Thus, we can safely assume that such Route Servers make many best path decisions for any given prefix weighing their merits against each other according to the protocol in action.

An example of an IXP facilitating interconnection between participating ASes is shown in Figure 1. In this example, traffic from AS X destined for dst can be routed through both AS A and AS B via respective links $L1$ and $L2$. With the current IXP model, the best path to reach dst is selected through the BGP path selection algorithm [27]. This selection considers route parameters such as AS_PATH length and Multi-Exit Discriminator which are not reflective of actual QoS parameters of a path as we have seen in §2 and are agnostic to available bandwidth on the path, congestion, and latency. Once a path for destination dst is chosen by the Route Server, say path through $L1$, AS A's edge router will be advertised as next hop to all other participants and all traffic destined to dst will pass through this path. If dst is a high traffic destination prefix, this will lead to uneven distribution of traffic as path through $L2$ will remain idle. It is a plausible assumption that such inefficiencies affect modern IXPs considering the large numbers of participants and the dense peering network.

In this model, we introduce a mechanism to override BGP path selection and introduce constraint-based QoS routing to the IXP by reacting to fluctuations in quality parameters such as link utilization and latency in addition to special participant policy constraints. We employ measurement, reactive path selection, and easier policy management to enable participants to make better peering decisions more easily. This section discusses in detail the operational model of the proposed system.

5.1 Design Goals

Here we present the goals in designing this system. These goals reflect in the choices we have made. We target a solution with the following properties:

1. **No hardware or protocol restrictions:** No requirement for participants of the IXP to install any new hardware or to implement some new or unknown protocol to utilize the new services we intend to introduce. If this is not the case, our work would suffer the same fate as previous BGP QoS extensions.

2. **No cooperation between ASes:** We did not want to require our ASes to cooperate mutually to achieve QoS routing. This cooperation could be in the form of Service Level Agreements or trust that promised QoS would be delivered.

3. **Minimal management or operational overhead:** Impose as minimal overhead on an AS that desires to utilize our route control QoS services as possible. Ideally, a participant, if it does not desire to utilize the new IXP services, should not have to face any overhead; and only minimal if they opt in for the new services.

4. **Incrementally deployable:** Our system should be designed such that an AS can incrementally deploy and test it and then phase out to full migration. An all or nothing solution would only invite fear of downtime from an IXP.

5.2 Measurement Techniques

To collect and dissipate the necessary information required for path management at the IXP, we:

- Query participant edge routers for link capacity (either wire speed or designated if the link is rate limited) using the SNMP ifSpeed Object Identifier [25]. Participants are expected to set these values and provide the IXP with read access to the relevant MIB.

- Measure bytes per second output at each switch port using OpenFlow counters [6]. These counts are delivered periodically and stored in a database for future reference.

- Combine the above to measure link utilization [2] by:

$$Output\ util = \frac{\Delta ifOutBytes \times 100}{(number\ of\ seconds\ in\ \Delta) \times ifSpeed}$$

  High output utilization values such as 97% and above indicate congestion at the link caused either by rate limiting of outgoing traffic or by hardware switching speed limitation of the device. We have used 97% in our demo as congestion effects were not visible for lower values. However, this can be adjusted based on testing in a real scenario.

- Measure bytes per second sent from different participant networks to destination prefixes using OpenFlow counters. These give us the average traffic destined to each prefix from each AS. Later, this information is used to offload traffic to other paths to achieve even distribution of traffic and reduce congestion.

- Send ICMP probes to determine latency, jitter and loss to the most active IP addresses through each available path. The frequency of probes is fine tuned to achieve granularity while avoiding excessive measurement traffic.

– Compile temporal trends from the above data to inform network operators of diurnal cycles and characteristics of potential peers. Hourly averages of latency, loss, and jitter are computed and recorded.

Available throughput of a path to a destination prefix can be valuable to our system to avoid self-load effects in intelligent route control as discussed in §6.1. However, we have excluded this from our model as we have not found a reliable technique for measuring throughput without end-host cooperation and any resource reservation or information protocol would go against our goal of not imposing any new hardware or protocol requirements on participant ASes.

5.3 Policy Abstraction

As discussed in §2, it is very difficult for AS administrators, especially in case of multilateral peering, to keep a complete, updated view of their implemented policy, change it and implement the changed policy over any number of routers installed at the IXP premises. Though the SDX project allows participants to express policy in the Pyretic programming language and have have it implemented across the IXP, it has no provisions to allow QoS-aware policies and time of day routing. To address this, we enable a more abstract expression of participant policy by providing a JavaScript Object Notation (JSON) API to network operators to express next hop agnostic, QoS aware policies as simple key-value pairs. This API also allows them to specify quality constraints in terms of latency, loss and jitter and peering constraints in terms of whitelist and blacklist parameters that BGP does not allow; constraints that the operators had to employ convoluted techniques, prone to error, to implement. The system translates this policy and implements it across the participant's presence at the IXP. For example,

```
"outbound": {
  "120.0.0.0/16": [ {
    "from": "110.0.0.0/16",
    "srcport": "*",
    "dstport": "80",
    "time": "0, 12",
    "latency": "< 10ms",
    "jitter": "*",
    "loss": "< 0.2%",
    "whitelist": [],
    "blacklist": ["X"]
    }
  ]...
```

the above shows an example of outbound policy where the operator defines special behavior for traffic destined for subnet 120.0.0.0/16. Further filters can be applied so that routing is not done based only on destination IP. Here, the operator specifies that source IP be in the prefix 110.0.0.0/16, places no condition on the source transport layer port, and specifies that destination port be the web port 80 (for application specific peering). These filters are possible because OpenFlow-enabled switches can perform a match on many packet headers instead of just the destination IP.

The time constraint is for 0 to 1200 hours. This enables time of day routing, i.e. different behaviour during different times of day. Latency constraint has been set to $< 10ms$ and packet loss rate $< 0.2\%$. With an empty whitelist, the operator indicates that the IXP can choose any AS, except the blacklisted AS X, to route this traffic through while honoring the constraints. Vice versa, an empty blacklist and a non-empty whitelist would imply that the IXP can only chose from the whitelisted ASes. The full AS_PATH information present in BGP route advertisements allows our system to select paths such that blacklisted ASes are not traversed along the whole path. The operator is alerted if none of the available paths satisfy the given constraints so that they can define a more flexible policy.

Similar to the outbound policy shown above, participants also define inbound policy with whitelist and blacklist constraints. QoS constraints are not supported for inbound policy specification. In future, inbound policy can be used to express transit fees for incoming traffic for different source ASes and bundles. Besides providing a more expressive form for policy, this mechanism also addresses the loss of control and predictability associated with multilateral peering at IXPs by supporting extremely fine-grained control over traffic.

5.4 QoS-based Link Allocation

For delivering QoS to the participants we considered the following known interdomain QoS solutions:

1. Resource reSerVation Protocol (RSVP) and variants
2. Constraint-based routing
3. QoS-based resource allocation

The Resource reSerVation Protocol addresses the problem of reserving resources along an end-to-end path for guaranteed QoS delivery. The sender sends a PATH message to the receiver specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH message to the next hop and a RESV message propagates back to the sender indicating either success or failure of resource reservation. As RSVP requires cooperation from independent ASes for QoS delivery, we concluded that it is not suitable for our IXP QoS solution. The reader is referred to [34] for details on RSVP and related protocols.

Constraint-based routing relates to our problem in that we have similar latency, loss, jitter and potentially, cost constraints on path selection. Constraint-based routing is concerned with finding a full path making routing decisions at every next hop. Thus, the algorithms discussed in [21] consider weights and constraints at every link (u, v) along the path. However, our problem is to only select the immediate next hop adjacent to the IXP subject to given constraints. Thus the algorithms discussed in [21] are not relevant.

We have found that our problem maps more closely to the resource allocation problems for QoS applications as discussed in [26]. Solutions discussed in [26] are based on those to different variations to the knapsack problem. The knapsack problem and its variations have been studied in operations research for decades. We have found that our problem maps to the 0-1 multiple knapsack problem as explained in this section.

Initially, we restrict our problem to finding a link allocation scheme for only one destination prefix. We denote the interval between two successive allocations by *update interval*, whose duration is T_u seconds. At the beginning of the nth update interval, the transmission rate of each flow is averaged over T_u seconds of the $n-1$th interval, denoted by

b_k *for* $k = 1, 2, 3, ...K$. Here, K is the set of traffic flows. There are two types of flows in this system, defined as:

1. a BGP flow characterized by destination prefix and source AS
2. a special policy flow characterized by the filters in the JSON API as shown in the previous section

We define $K_{bgp} \subseteq K$ as the set of all BGP flows and $K_{js} \subset K$ as the set of all JSON policy flows. Each flow $k \in K_{js}$ has a set of quality constraints Q_k. Also, we have the set of egress links L that are candidate routes to the destination prefix with each link having a capacity $C_l \; \forall \; l \in L$ as determined by the SNMP queries described in §5.2. We assume that the average bandwidth of a flow over T_u seconds of a given interval remains relatively constant. Allocating such fixed bandwidth flows to multiple candidate fixed-capacity links maps to the 0-1 multiple knapsack problem where each knapsack or link has the capacity C_l and each item or flow has the weight b_k. Since we intend to maximize the bandwidth utilized, the problem becomes:

$$\text{maximize} \sum_{l=1}^{m} \sum_{k=1}^{n} b_k x_{lk}$$

$$\text{subject to} \sum_{k=1}^{n} b_k x_{lk} \leq C_l, \text{ for all } 1 \leq l \leq m$$

where x_{lk} is a binary variable such that $x_{lk} = 1$ if flow k is accommodated on link l and 0 otherwise. We also have the restriction that link l satisfies all $q \in Q_k$ quality constraints when $x_{lk} = 1$.

There are many exact and approximate solutions to the 0-1 multiple knapsack problem in the literature as it has been studied in the operations research domain for decades. A detailed analysis of each solution is beyond the scope of this work. We have used a heuristic algorithm as discussed in [23] for an approximate solution in our implementation as exact solutions are computationally expensive for large problems. Martello and Toth, the authors, use Lagrangian and surrogate relaxations as heuristics and convert the multiple knapsack problem into m single knapsack problems. The knapsacks need to be ordered so that:

$C_1 \leq C_2 \leq ... \leq C_m$where m is the number of knapsacks

The worst case complexity of this algorithm for finding an approximate solution is $O(n^2)$ as compared to exponential complexities of exact algorithms. The authors of [23] refer to this algorithm as $MTHM$. We also make the following considerations:

- Flows in K_{js} are given priority so that a separate optimization problem is solved first where only $k \in K_{js}$ are considered as items. Once all flows in K_{js} are allocated for which there is at least one link that satisfies all of its quality constraints, link capacities are updated as shown below and the next optimization problem is solved for K_{bgp}.
- If there are no links that satisfy quality constraints of a $k \in K_{js}$ flow, it is considered as a BGP flow and moved to the set K_{bgp}. A corresponding alert is generated at the system.
- Congestion only triggers a partial reallocation in the scheme where the flow with the smallest b_k at the congested link is moved to a link where it can be accommodated. It is not moved if there are no such links available and reallocation is deferred to the next *update interval*.

Once allocation for a prefix is completed, link capacities are updated by:

$$C_l = C_l - b_k \text{ if } x_{lk} = 1$$

The above optimization is applied again, and so on for each prefix. It is worth noting here that we have considered $p_i = w_i$ in solving the knapsack problem where p_i and w_i are the profit and weight of the ith item respectively. Here, the knapsack problem intends to maximize profit p_i. In future, a different composite value for p can also be used to prioritize flows based on, e.g. the price the sending AS is willing to pay.

Since the optimization problem we have presented here is solved at the controller and not the routers or switches, we can expect to keep the *update interval* small so that QoS provision at the IXP does not suffer from coarse granularity. This resource allocation scheme may suffer from self-load, path oscillation, and other issues that we discuss in §6.

6. ARCHITECTURE AND IMPLEMENTATION

In this section, we discuss the architecture and pilot implementation of our model. We have implemented this model as a service on top of the Software Defined eXchange architecture presented in detail in [16] as shown in Figure 2. The SDX project is written in the Pyretic network programming language [24].

We have used Pyretic query policies in both parallel and sequential composition for OpenFlow counters required to measure per flow input and per port output. Python SNMP bindings are used to query edge routers for interface speeds as discussed in §5.2. These measurements, along with BGP route announcements and best path selections are stored in a MySQL database. This database is used to mark whether a BGP route is overridden and if yes, for which ASes. In this manner, the IXP can select and announce different paths to different ASes for the same prefix.

To measure latency, jitter, and loss, we have placed a special measurement host as shown in Figure 2 that acts as a participant of the IXP. This host has special layer 2 policy designed according to the IXP port configuration such that it can connect to all other participants and send ICMP echo requests to make round trip time measurements. The top five most active hosts for each route advertised by each AS are used to measure round trip time against. One echo request is sent every ten seconds to each such host to avoid excessive measurement traffic. These requests are uniformly spread out over the *update interval* to avoid sudden surges. Python Scapy is used for packet construction and parsing [5]. For a very large IXP with upwards of 1 million routes (counting different announcements for the same IXP), this may lead to scalability issues. To account for this, dedicated measurement hosts may be placed at each participant AS's edge or other distributed measurement mechanisms may be

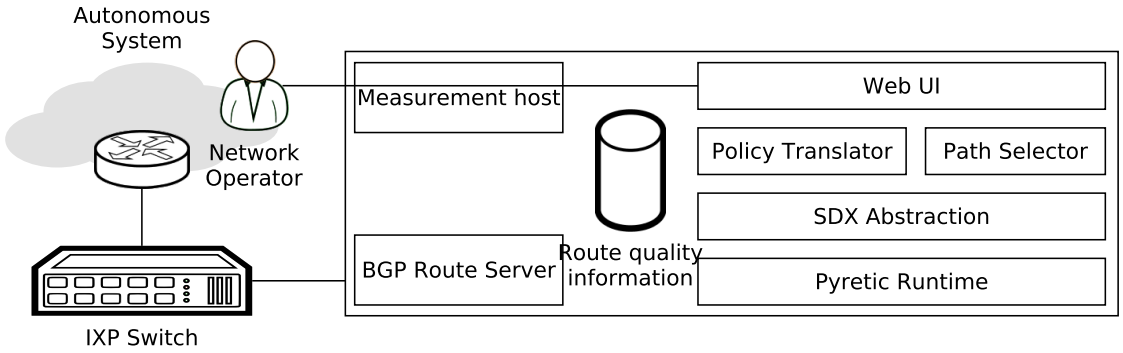

Figure 2: IXP system architecture

designed, the discussion of which, is beyond the scope of this work.

To respond to changes in quality parameters like latency, jitter, loss and link utilization, we have extended the DynamicPolicy class of Pyretic. The JSON policy received from network operators is translated into objects of this class. An object of this class queries the MySQL database every t seconds to see if all constraints as expressed in the participant policy are met. If not, the flow represented by this policy is marked as requiring a new path allocation according to the scheme discussed in §5.4 and a new path is allocated at the next update interval. The self.policy field of the object is reassigned to forward matching traffic to the new path. This reassignment results in the underlying Pyretic runtime to issue corresponding instructions to the controller to send rules to the relevant switch or switches. The time of day routing aspect of the JSON API discussed in §5.3 is implemented by scheduling periodic tasks that reassign the self.policy field to reflect changes in the path.

We provide participants with a web interface to upload policy files, have a comprehensive view of their policy and automatically selected paths and rules relevant to them and receive alerts. We have extended the IXP-Manager project for this purpose: an open source IXP management web application released by INEX [4]. Our extension of this web portal also serves the purposes of a route collector by publishing participant ASes and their routes along with the collected quality information. Incorporating the visiting operator's AS traffic input records, the portal matches them against best route providers of the relevant prefixes and displays a list of important existing and potential peers.

6.1 MTHM Implemenation

We use the $MTHM$ algorithm to solve the problem of allocating links to traffic flows of known bandwidth. We implemented the algorithm in C and used *ctypes* to call it from the Python controller. We tested this implementation by generating random flow rate averages uniformly distributed between 1 and 100mbps and link capacities between 1 and 10gbps. On a modern Intel Core machine, this implementation solves the problem of allocating 10^6 flows to 10^4 links under 10ms. As we increase the problem size to 10^7 flows and 5×10^4 links, execution time increases to about 60ms on average. With the $O(n^2)$ complexity of the algorithm, we do not expect execution speed to be an issue in solving problems of similar scale at real IXPs.

6.2 Implementation Issues and Corrective Measures

An Intelligent Route Control (IRC) mechanism deployed at the IXP can suffer from issues similar to one designed for intradomain load balancing. These issues are discussed in [13]. The self-load effect is defined as when a previously uncongested or underutilized link suffers from congestion effects after a flow is routed to it. Though we are considering average rate of a flow from a previous interval, self-load effects can surface if a flow assigned to a link with high utilization experiences surge. This can be mitigated, to some extent, by predicting the transmission rate of a flow based on past traffic measurements. Several techniques have been proposed in the literature for such predictions. Also, increased loss rate along a path can serve to signal congestion in a distant link along it, i.e. a link beyond the edge router at the IXP.

The synchronization effect discussed in [13] is not applicable to our system as multiple IRCs are not at work and their respective measurement periods cannot coincide to project a false picture of link utilization and QoS parameters.

Since the IXP reacts to over utilization and quality degradation, it can suffer from path oscillation if performance churns quickly. Issues usually associated with BGP route flaps are relevant here, and techniques described in [31] are applicable. Path oscillations can be more severe here as multiple paths may have to be adjusted to update the path for one flow. Since we are reconfiguring the path oscillation scheme only after T_u seconds have passed, we are already using the fixed timer technique discussed in [31]. However, congestion resulting from over utilization of a link results in immediate reallocation of a flow and the fixed time rule is violated here. The following techniques can be used to minimize path oscillation in this model:

† **Transmission rate prediction:** Transmission rates of flows predicted from previous history can be used for path allocation instead of just the average rate from previous interval. This will result in less congestion alerts due to surges in transmission rates.

† **Stability sensitivity:** Previous history can be used to predict future stability of a route. Routes with higher stability prediction can be allowed a longer duration of degraded performance before a QoS failure is marked.

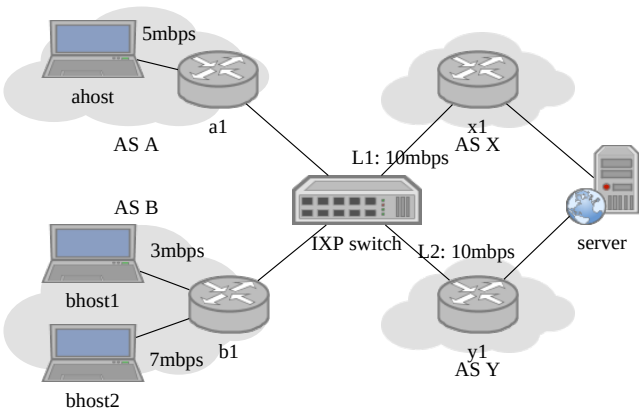

Figure 3: Experiment Topology

† **Fixing large, stable flows:** Flows with high transmission rates that are stable over many *update interval* durations can be permanently allocated to accommodating links such that these flows are excluded from the maximization problem discussed above and are not unnecessarily shuffled.

† **Only move non-satisfied flows at successive updates:** To avoid a full recompute of the routing scheme and thus moving many flows around, only flows with non-satisfied QoS constraints can be considered as items in the optimization problem of the next interval. This will introduce stability in the allocation scheme but the trade-off would be non-optimal allocation for resource utilization.

Further research can be conducted to determine utility of the above techniques in a real IXP deployment. This may lead to a solution of the 0-1 multiple knapsack problem better suited to the QoS constraint model we have here and where weights can be attached to flow movement such that minimum flows are reallocated at each successive update.

7. DEMONSTRATION

We demonstrate our model on a one-switch IXP simulation testbed implemented in Mininet. We use Linux traffic control commands to rate-limit links, and simulate latency, jitter, and loss in the simulated topology. Testbed topology can be seen in Figure 3. We conduct two experiments on this simulation: **a)** iperf [14] TCP bandwidth tests and **b)** frame corruption comparison for high quality video streaming. In these experiments, *ahost*, *bhost*1 and *bhost*2 send TCP test data and stream video to *zhost* which they can reach through both ASes X and Y.

7.1 iperf Tests

We run two tests to conduct the iperf experiment: (1) with traditional BGP (2) with our IXP implementation. For each test case, 60 second iperf TCP bandwidth tests were run with TCP window sizes of 85.3 KB on all hosts. *ahost* has its upstream link rate limited at 5mbps, *bhost*1 at 3mbps and, *bhost*2 at 7mbps through Linux traffic control commands. The edge routers of ASes X and Y, $x1$ and $y1$ each have a 10mbps link with the IXP switch. *ahost* and *bhost*1 establish TCP connections to server and start transmission

Test Case	Average Throughput		
	ahost	**bhost1**	**bhost2**
Test Case 1	3.42	2.34	3.16
Test Case 2	4.78	2.87	5.86

Table 1: Average throughput after bhost2 starts transmission

before time $t = 0$ while *bhost*2 starts at $t = 20$. End-to-end throughput for each connection is recorded at the *zhost* node and link utilizations for $L1$ and $L2$ are monitored. At the end, we compare the average end-to-end throughputs for all three sessions for both test cases to verify that our IXP implementation utilizes the available links more efficiently.

† **Test Case 1:** For reaching *zhost*, BGP path selection algorithm selects edge router $x1$ as next hop (e.g due to smaller AS_PATH length). This next hop is advertised to both ASes A and B such that all traffic to *zhost* is sent over $L1$. Initially, *ahost* and *bhost*1 transmit at about 5mbps and 3mbps respectively (see Figure 4(a)) as their uplinks are rate limited as such. Total traffic through link $L1$ remains at about 8mbps, i.e. well under its capacity of 10mbps, so we do not see any signs of congestion. However, after *bhost*2 establishes connection to *zhost* at about $t = 20$ and attempts to increase its transmission rate to 7mbps, we see adjustments as BGP does not react to increasing utilization of link $L1$ as shown in Figure 4(c). About 15mbps of traffic is forced to pass through a 10mbps rate limited link $L1$ as no alternative paths are selected. This forces all three connections to backoff and surge as observed in Figure 4(a). Eventually, the transmission rates stabilize but with lower throughput for all connections. As a result, average throughputs of *ahost*, *bhost*1, and *bhost*2 are lowered as shown in Table 1.

† **Test Case 2:** When the same experiment is conducted with our IXP implementation, and *bhost*2 starts transmission at time $t = 20$, the IXP starts detecting above 97% utilization of link $L1$ (see 4(d)). This signals overutilization of $L1$ and triggers a reallocation routine that will move some flows from $L1$ to bring utilization under normal levels. We have introduced an artificial delay of 10 seconds for demonstration purposes. After this delay, at $t = 30$, the IXP intervenes and redirects *bhost*2 to send data over $L2$. Since $L2$ has an unutilized capacity of 10mbps, *bhost*2 reaches its 7mbps transmission rate (see 4(b)) and $L1$ is relieved of the extra load resulting in higher average throughputs as shown in Table 1.

7.2 Video Streaming

This experiment is conducted by streaming a high quality video over UDP through the IXP testbed with controlled network conditions. Testbed topology is as previously shown in Figure 3. All rate limits are removed from links in the simulation. Here, *ahost* takes the role of a video server and streams video to *zhost*. As before, *ahost* can reach *zhost* through both ASes X and Y.

We run two tests to conduct this experiment: (1) with traditional BGP (2) with our IXP implementation. For each

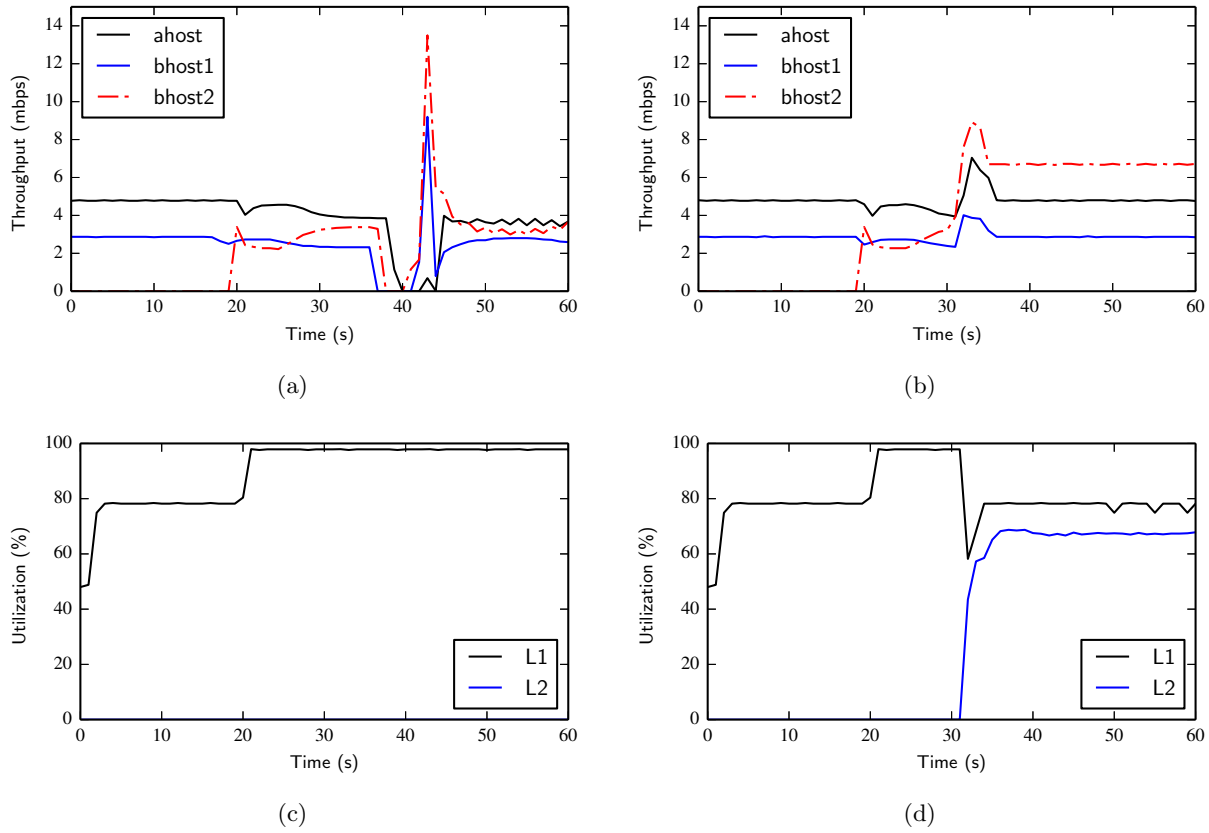

Figure 4: (a) Observed throughput under BGP (b) Observed throughput under quality-aware path selection (c) Link utilization under BGP path selection. Utilization of $L2$ remains at 0% (d) Link utilization under quality-aware path selection

test case, a 30 minutes 1920x1080 H.264 video is streamed from *ahost* to *zhost* to record video playback statistics. Latency and packet loss conditions of links $L1$ and $L2$, as shown in Figure 5(a) and 5(c), are made to toggle between two typical link conditions throughout the tests with Linux traffic control commands. This is done to simulate changes in the network QoS. Rate limits of all links are relaxed so that they are large enough to allow full speed video transmission. In this experiment AS A defines QoS constraint for traffic destined to *zhost* through our JSON API as shown below:

```
"outbound": {
  "120.0.0.0/16": [ {
    "from": "*",
    "srcport": "*",
    "dstport": "*",
    "time": "0, 24",
    "latency": "< 40ms",
    "jitter": "*",
    "loss": "< 0.2%",
    "whitelist": [],
    "blacklist": []
    }
  ]...
```

Here, assuming *zhost* lies in the prefix 120.0.0.0/16, AS A demands QoS where latency is kept under 40ms and loss rate is kept under 0.2% for all traffic destined to *zhost*.

At the end of the experiment, we compare video frame corruption at the client, i.e. *zhost*, for both test cases to verify that our IXP delivers demanded QoS through intelligent, constraint-based route selection.

† **Test Case 1:** Initially, video traffic from *ahost* to *zhost* is routed through AS X by BGP route selection as in the iperf experiment. Quality of link $L1$ degrades after a certain time as shown in Figures 5(a) and 5(c), but, as traditional BGP is insensitive to link quality variations, traffic is still routed through AS X over $L1$. We see that $L2$ is ignored throughout the test even when it offers a higher quality link to *zhost* that satisfies the QoS demand of AS A. Traditional BGP does not change the path in response to QoS changes, thus failing to meet the QoS demand of AS A and degrading the quality of the video at *zhost*. Frame corruption due to inferior QoS is shown in Figure 5(e). The average number of frames corrupted in this test case is 4.24 frames per second.

† **Test Case 2:** When the same experiment is conducted with our IXP implementation, AS A's outbound policy is implemented where it demands link to *zhost* prefix with QoS constraints as shown above. When link $L1$ of AS X degrades and fails to provide the QoS demanded by AS A, we see that the IXP reacts to these variations, and, according to QoS demands from the AS A, overrides routing configuration to re-

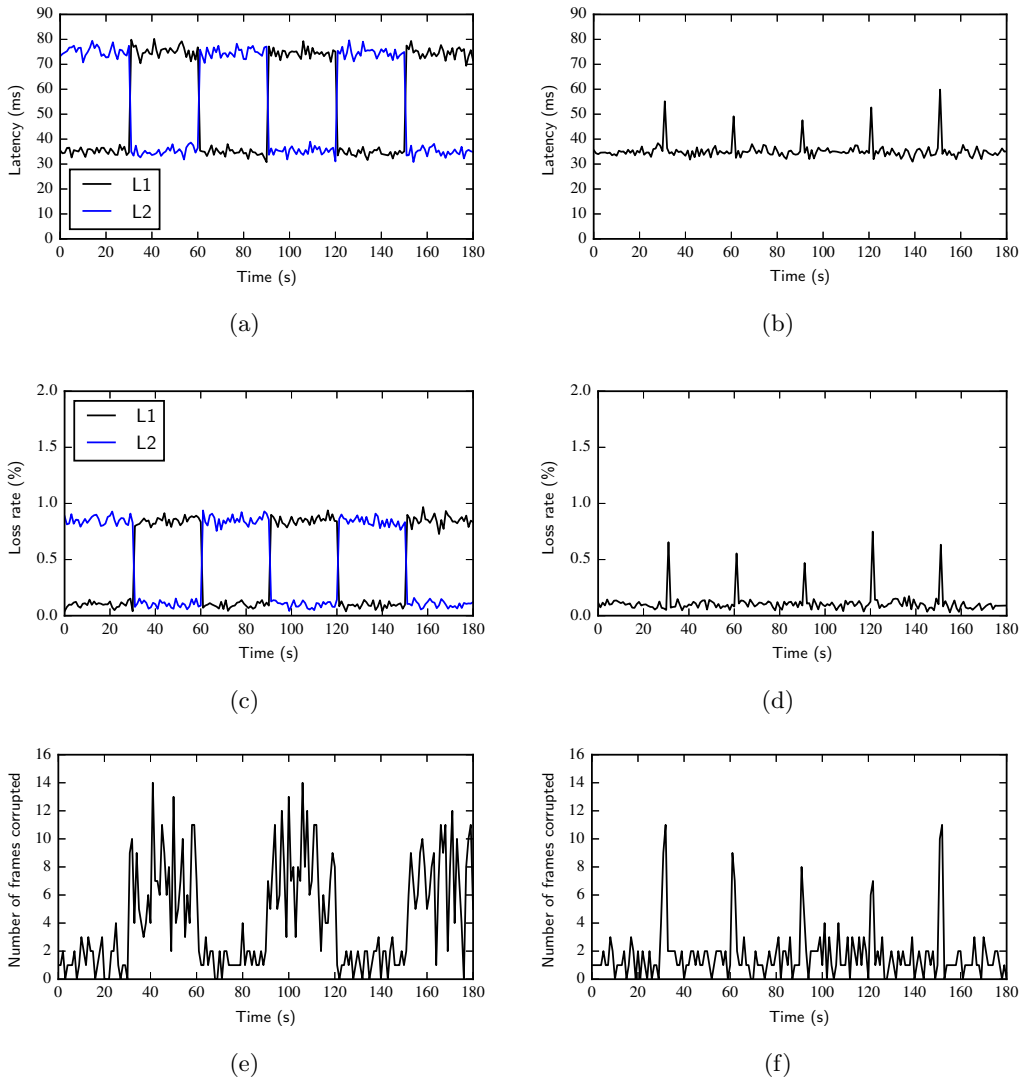

Figure 5: (a) Simulated latency on the links passing ASes A and B. (b) Observed latency on path between media server and client with route control. (c) Simulated packet loss on the links passing ASes A and B. (d) Observed packet loss on path between media server and client with route control. (e) Frame corruption with traditional routing. (f) Frame corruption with route control.

quired latency and packet loss. Average latency is kept under $40ms$ and packet loss under 0.2%, as shown in Figure 5(b) and 5(d). This results in demanded QoS being delivered and thus lesser frame corruption, as shown in Figure 5(f). The average number of frames corrupted in this test case is 1.71 frames per second.

8. DISCUSSIONS AND FUTURE WORK

In the previous sections, we have seen how our IXP model addresses the problems associated with public peering at IXPs and provides more control with more ease to participating network operators. The quality measurements our model takes and disperses through the web interface leads to more transparency for participants so they can make better decisions. In this section, we discuss how this model, given a few extensions can impact the current IXP industry more profoundly. To differentiate between peering and transit, consider two autonomous systems A and X. Pertaining to the size of their respective domains, the possibilities are:

- A has a larger network and more links, thus handles more traffic. A will see little incentive in peering with X and will thus charge X a fee for exchange of traffic between A and X. We call this a transit relationship where A acts as a transit provider for X.
- X has a significantly larger network than A and can act as transit provider for A similar to the above.
- Both A and X have domains of roughly the same size. They will both be interested in exchanging each other's traffic to expand their domains and diversify their respective interdomain routing spectra. This may lead to a settlement free peering relationship between A and X where no financial deals are made. The ASes just exchange traffic out of mutual benefit.

Public peering at IXPs implement relationships of the third kind above. Thus, mostly, ASes participating in public peering at a regional IXP have domains of roughly the same size and are engaged in settlement free peering to keep local traffic local. This is the reason the largest local AS is usually the last to join the IXP as it sees little incentive there before the combined domains of all other ASes present an opportunity to expand its own.

If traffic logs of our IXP model are used for billing ASes and delivering value to larger ASes, we can blur the lines between peering and transit. The JSON API our model provides can be used to define different pricing brackets and bundles for different destination prefixes, application layer protocols, times of day, and average volumes of traffic. OpenFlow counters on the other hand, as discussed in §5.2 can be used for granular traffic logs and billing. This would incentivize joining the IXP for larger players in the local market. They may charge for exteral traffic they provide transit for and not charge for ingress traffic destined for their own domains.

The quality measurements our model takes means that potential peers know the highest quality paths to reach a given destination under different time of day and protocol conditions and thus the AS providing that path will be able to charge more for it. This will promote a healthy competition between transit providers to improve their link qualities.

Another area of expansion for this work could be better layer 7 intelligence. For example, we could allow a participant to express a policy such that latency for a given prefix and protocol get minimized while keeping costs under defined threshold. A large AS could define inbound policy to maximize revenue such that the IXP keeps the AS's outbound link at maximum utilization with traffic it charges the most for. Such intelligent decisions are possible because of SDN's inherent control plane separation, the quality information our model gathers and the abstract expression of policy our JSON API affords.

For better evaluation of our system, deployment at a real IXP is essential. We are working with a regional IXP to see exactly how average throughputs are improved in a real scenario. We are also working to figure out how pricing brackets and bundles could be coupled with traffic logs to perform central billing at the IXP such that no AS has a disadvantage in joining the IXP.

9. INCENTIVES FOR ADOPTION

Network operators have traditionally been reluctant to experiment with new technologies and protocols fearing outage risks and losses. However, with declining profit margins, they are also under pressure to reduce operating costs. In this context, presenting a new framework requires significant financial incentives to encourage network operators to adopt it. Our model provides the following financial and operational incentives to IXPs and participants:

- The ubiquitous presence of the Border Gateway Protocol has been a major hindrance in the way of adoption of improved interdomain routing protocols. BGP and BGP-speaking hardware necessitate a difficult industry-wide overhaul for the adoption of any new protocol. Our model intends to escape that fate by working alongside BGP. For normal operation, i.e. without defining special behavior using our JSON API, an IXP's migration to our model will be completely transparent to its participants. The operators will not be required to handle any new protocols or operation procedures. The only requirement is to set appropriate value of the ifSpeed SNMP object as discussed in §5.2.

- Our operational model expects the participant edge routers to just be traditional L3 BGP speaking routers and does not require any new hardware, such as OpenFlow-enabled switches. This avoids requiring all participants to do a huge hardware overhaul that would have been a major hindrance in the adoption of this model.

- Designing, implementing, and maintaining external routing policies is a cumbersome task considering the limitations of BGP. Our model makes policy specification much easier and closer to the actual high level policies of the AS instead of requiring strict conformance to a very limited protocol.

- Our model makes quality parameters of links advertised by each participant available to others. This would be welcomed by ASes that maintain better infrastructure and peering links.

- As we have seen in §4, traffic engineering at IXPs is cumbersome, especially for public peering; which is the dominant form at large IXPs on the European IXP model [10]. Accordingly, controlling inbound and outbound traffic means using arcane methods that become rigid and prove a hindrance to the dynamic nature of peering our model promotes. Participants will see our JSON API of §5.3 as an easy and expressive mechanism to use the IXP as an optimized route control service provider to utilize network aware load balancing and QoS provision functions of the new model or a vendor neutral *pipe* or anywhere in between.

- Owing to new route control and QoS services provided to participants, the IXP can charge an increased membership fee from those that choose to avail these services.

- As discussed in §8, automatic pricing and billing can be easily incorporated into this model. This can be a financial incentive for large ASes who have better quality infrastructure as they have more to gain from this model by offering higher quality transit to more destinations.

- Small, local ASes on other hand may see the establishment of an open transit marketplace as more favorable to them compared to closed SLAs with few providers. This will not affect the already established settlement-free peering culture at IXPs because a local AS charging for traffic destined to its own network does not make much sense.

- The IXP may get into a royalty-based agreement with participants so that it gets a share of the financial transactions taking place in its premises. This will be a big financial incentive for for-profit IXPs.

10. CONCLUSION

Building upon the innovative designs of the SDX project [8] and Pyretic [24], we have designed a new IXP model that addresses many problems experienced by network operators at current IXPs.

Our JSON API allows participants to express high-level, next hop agnostic and QoS aware policies in an abstraction much closer to the actual interdomain AS policies. Our QoS-based resource allocation scheme allows for an optimal configuration of flow paths. This scheme keeps the whole network in view and thus achieves better utilization of available resources while also meeting QoS constraints of participants.

Bringing together known techniques and existing technologies, we have designed a comprehensive IXP model, presented a viable architecture and a pilot implementation of a model that could potentially revolutionize the way IXPs operate. However, with the methods of route control, optimal path allocation, and QoS provision described in this paper, we have clearly just scratched the surface of what is possible if intelligent route control is brought to vantage points as big as modern IXPs. Our work opens a new chapter in the continued evolution of IXPs from infrastructure providers to market developers.

11. REFERENCES

[1] Ams-ix historical timeline, `https://ams-ix.net/about/historical-timeline`. Technical report.

[2] Calculate bandwidth utilization using snmp. `http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/8141-calculate-bandwidth-snmp.html`.

[3] De-cix traffic statistics, `http://www.de-cix.net/about`. Technical report.

[4] Ixp manager. `https://github.com/inex/IXP-Manager`.

[5] Python scapy. `http://www.secdev.org/projects/scapy/`.

[6] Openflow specification. `https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf`, 2013.

[7] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a large european ixp. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 163–174. ACM, 2012.

[8] J. Bailey, R. Clark, N. Feamster, D. Levin, J. Rexford, and S. Shenker. Sdx: A software defined internet exchange. *Open Network Summit (Research Track)*, 2013.

[9] N. Chatzis, G. Smaragdakis, J. Böttger, T. Krenc, and A. Feldmann. On the benefits of using a large ixp as an internet vantage point. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 333–346, New York, NY, USA, 2013. ACM.

[10] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. There is more to ixps than meets the eye. *SIGCOMM Comput. Commun. Rev.*, November 2013.

[11] C. de Launois, B. Quoitin, and O. Bonaventure. *Leveraging Network Performances with IPv6 Multihoming and Multiple Provider-Dependent Aggregatable Prefixes*, volume 3375 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005.

[12] A. Fonte, M. Curado, and E. Monteiro. Interdomain quality of service routing: setting the grounds for the way ahead. *annals of telecommunications - annales des tÃl'lÃl'communications*, 63(11-12):683–695, 2008.

[13] R. Gao, C. Dovrolis, and E. Zegura. Avoiding oscillations due to intelligent route control systems. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[14] E. Goldoni and M. Schivi. End-to-end available bandwidth estimation tools, an experimental comparison. In *Proceedings of the Second International Conference on Traffic Monitoring and Analysis*, TMA'10, pages 171–182, Berlin, Heidelberg, 2010. Springer-Verlag.

[15] F. Guo, J. Chen, W. Li, and T. cker Chiueh. Experiences in building a multihoming load balancing system. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1241–1251 vol.2, March 2004.

[16] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. Sdx: A software defined internet exchange. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 551–562, New York, NY, USA, 2014. ACM.

[17] B. Huffaker, M. Fomenkov, D. Plummer, D. Moore, and k. claffy. Distance Metrics in the Internet. In *IEEE International Telecommunications Symposium (ITS)*, pages 200–202, Brazil, Sep 2002. IEEE.

[18] E. Jasinska, N. Hilliard, R. Raszuk, and N. Bakker. Internet exchange route server. Internet-Draft draft-ietf-idr-ix-bgp-route-server-02, February 2013.

[19] V. Kotronis, X. Dimitropoulos, and B. Ager. Outsourcing the routing control logic: Better internet routing based on sdn principles. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 55–60, New York, NY, USA, 2012. ACM.

[20] V. Kotronis, X. Dimitropoulos, R. Klöti, B. Ager, P. Georgopoulos, and S. Schmid. Control exchange points: Providing qos-enabled end-to-end services via sdn-based inter-domain routing orchestration. In *Presented as part of the Open Networking Summit 2014 (ONS 2014)*, Santa Clara, CA, 2014. USENIX.

[21] F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz. An overview of constraint-based path selection algorithms for qos routing. *Communications Magazine, IEEE*, 40(12):50–55, Dec 2002.

[22] K. K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford. Routing as a service. Technical Report UCB/EECS-2006-19, EECS Department, University of California, Berkeley, Feb 2006.

[23] S. Martello and P. Toth. Heuristic algorithms for the multiple knapsack problem. *Computing*, 27(2):93–112, 1981.

[24] C. Monsanto, J. Reich, N. Foster, J. Rexford, D. Walker, et al. Composing software-defined networks. In *USENIX NSDI*, page 2, 2013.

[25] D. Perkins and E. McGinnis. *Understanding SNMP MIBs*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[26] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. Practical solutions for qos-based resource allocation problems. In *Real-Time Systems Symposium, 1998. Proceedings., The 19th IEEE*, pages 296–306, Dec 1998.

[27] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), Jan. 2006. Updated by RFCs 6286, 6608, 6793.

[28] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger. Peering at peerings: On the role of ixp route servers. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 31–44, New York, NY, USA, 2014. ACM.

[29] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. Overqos: An overlay based architecture for enhancing internet qos. In *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, pages 6–6, Berkeley, CA, USA, 2004. USENIX Association.

[30] J. Van der Merwe, A. Cepleanu, K. D'Souza, B. Freeman, A. Greenberg, D. Knight, R. McMillan, D. Moloney, J. Mulligan, H. Nguyen, M. Nguyen, A. Ramarajan, S. Saad, M. Satterlee, T. Spencer, D. Toll, and S. Zelingher. Dynamic connectivity management with an intelligent route service control point. In *Proceedings of the 2006 SIGCOMM Workshop on Internet Network Management*, INM '06, pages 29–34, New York, NY, USA, 2006. ACM.

[31] C. Villamizar, R. Chandra, and R. Govindan. BGP Route Flap Damping. Technical Report 2439, Nov. 1998.

[32] L. Xiao, K.-S. Lui, J. Wang, and K. Nahrstedt. Qos extension to bgp. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 100–109, Nov 2002.

[33] L. Xiao, J. Wang, K.-S. Lui, and K. Nahrstedt. Advertising interdomain qos routing information. *Selected Areas in Communications, IEEE Journal on*, 22(10):1949–1964, Dec 2004.

[34] X. Xiao and L. M. Ni. Internet qos: A big picture. *Netwrk. Mag. of Global Internetwkg.*, 13(2):8–18, Mar. 1999.