Exploiting Path Diversity for Networked Music Performance in the Publish Subscribe Internet

Yannis Thomas, George Xylomenos and George C. Polyzos
Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
Patision 76, Athens 104 34, Greece
Email: {thomasi,xgeorge,polyzos}@aueb.gr

***Abstract*—The high bandwidth and low delay requirements of Networked Music Performance (NMP) often lead to doubts about its feasibility in wide area networks. Multipath routing with Quality of Service (QoS) constraints is one way to meet these requirements, as it offers bandwidth aggregation while keeping latency low. However, IP does not natively support multipath QoS routing, requiring complicated extensions to network operation. In contrast, the Publish-Subscribe Internet (PSI) architecture, an Information-Centric Networking (ICN) approach, natively supports multicast, centralized path selection and source routing, all the main elements needed for multipath QoS routing. In this paper, we discuss the implementation of multipath QoS routing in PSI and experimentally evaluate its benefits for NMP services.**

I. Introduction

Networked Music Performance (NMP) is a multimedia service that interconnects users with video and audio streams, so as to allow scattered musicians to perform together in real-time. NMP differs from other high-bandwidth media streaming services, such as *Video on Demand* (VoD) and *Live Video Streaming* (LVS), in its non-negotiable ultra low-delay requirements, which are critical for allowing musicians to synchronize their performances. Unlike VoD and LVS which allow a startup buffering period of seconds, NMP requires mouth-to-ear latency to be kept below 25 msec [1].

NMP also differs from real-time media streaming services, such as *Voice over IP* (VoIP), due to its bandwidth requirements. VoIP systems commonly rely on a centralized server, also known as a *Multipoint Conferencing Unit* (MCU), which receives media streams from all users, mixes them, and sends a single mixed stream to each user. NMP however requires each user to receive the media transmitted by all others, so as to perform its own mixing; this can be achieved either by the user sending a single media stream to a relay server that sends a copy to each receiver, or by the user directly sending a separate copy of its media stream to each receiver. In past work [2] we compared NMP delay with a relay against direct communication between users, finding that the triangular communication imposed by the relay can increase network delay by more than 10%. Unfortunately, direct communication requires the sender to inject more traffic into the network.

Since the traffic volume of NMP in direct communication mode cannot be reduced without increasing delay, we can instead attempt to better spread it across the network, so as to amortize its impact. Multipath transmission is a well-known content delivery technique that serves this purpose, offering bandwidth aggregation, network load balancing and resilience to link failures. Nevertheless, the suitability of multipath for low delay applications is questionable, since its overall service latency is the latency of the slowest path used. To address this problem, *Quality of Service* (QoS) constraints can be used in the routing process to balance bandwidth aggregation and latency augmentation, via multi-metric path selection algorithms, in an attempt to provide the required throughput, while keeping the delay of the slowest path low.

IP networks cannot support these techniques without complicated modifications to network logic, such as establishing media proxies or overlay networks with *Multi Protocol Label Switching* (MPLS) routing. The complexity and scalability issues of MPLS-based QoS are due to the lack of information awareness in the network [3]. This limitation is resolved in *Information-Centric Networking* (ICN) [4], a clean-slate Internet architecture that considers information as the center of all network operations, unlike IP which is based on end-host interaction. *Publish-Subscribe Internet* (PSI) [5] is an ICN architecture that natively supports multicast, centralized path selection and source routing. These features are the base for implementing advanced QoS routing techniques in PSI that can make NMP services feasible. This paper considers a wide range of QoS constrained multipath algorithms for PSI and experimentally evaluates their performance in an NMP setting.

The rest of this paper is organized as follows. We first discuss video coding for multipath streaming and IP solutions to the problem in Section II. We then present PSI, introducing our design for accommodating NMP in Section III. In Section IV, we illustrate the bandwidth and latency constraints that characterize NMP and in Section V we experimentally investigate the enhancements that multipath QoS routing can offer to NMP. Finally, we conclude by summarizing the contribution of our work and our future plans in Section VI.

II. Related work

A. Multistream video coding

While audio and video latency are equally critical for NMP, audio requires much less bandwidth than video, therefore a single low latency path is normally sufficient for audio transmission. In contrast, the increased throughput requirements of video streaming have led researchers towards video coding methods that can exploit rich network topologies, such as *Multiple Description Coding* (MDC) [6]. MDC breaks a high quality video stream into autonomous lower quality substreams, commonly with lower spatial or temporal resolution, that can be recombined to produce the initial high quality video. These substreams can be transferred via multiple parallel paths, thus offering easier adjustment to congestion patterns across the network, reduced correlation between consecutive packet losses and increased throughput [7].

Although MDC enhances service quality by better utilizing network resources, it does not mitigate the impact of video streaming on network load. *Scalable Video Coding* (SVC) [8] was proposed for reducing the volume of data transported in the network by exploiting the increasing diversity of end-user devices. SVC breaks a high quality video stream into lower quality substreams and serves different sets of substreams to individual users, based on their capabilities. Therefore, SVC avoids sending excessive data to low resolution devices (e.g., tablets and smartphones), while meeting the intensive requirements of high resolution ones (e.g., desktops and laptops).

The main difference between MDC and SVC is the decoding dependencies of substreams. All MDC substreams are autonomous, thus allowing decoding when all other streams are lost or late, whereas SVC streams present a hierarchical dependency. An SVC video consists of two types of layers; the base layer (BL) and the enhancement layers (EL). The BL is the lowest quality layer and is considered autonomous, as it can be decoded in the absence of any enhancement layers. An EL layer improves the quality provided by the lower sublayers, hence its decoding requires all the lower ELs and the BL. A useful comparison between MDC and SVC is provided in [9].

B. Path diversity in IP

The suitability of path diversity for multistream video coding is well known and has been investigated in the past [9], [10], [11]. For exploiting path diversity, two methods are proposed: multisource and multipath.

- Multisource is mainly used in *Content Delivery Networks* (CDNs), where information is stored in multiple physical locations. A request can be served by any CDN node that has the required information, thus increasing content availability; usually the closest node sends the data, thus minimizing service latency and reducing network load. However, CDNs are not designed for distributing live content, hence they are not suitable for NMP.
- Multipath is supported in IP networks for multihomed users only, in which case it suffers from path convergence. The hop-by-hop forwarding scheme of IP does not guarantee that the paths will be disjoint, thus limiting route diversity. MPLS which employs source routing is commonly utilized for path diversity, but only in networks where it is available.

MPLS is used in backbone networks, where it applies QoS-based traffic control by classifying flows and forwarding them via predefined routes. Nevertheless, MPLS requires maintaining state in the routers (MPLS tables) and it cannot apply fine grained QoS control within a service. For example, when SVC is used, MPLS cannot prioritize the BL substream which is required by all other substreams, since all substreams are classified as a single flow in the access router of the MPLS network, based on basic TCP/IP header fields. Adding logic in MPLS routers to distinguish substreams would severely increase their computational cost, which goes against the MPLS goal of pushing complexity to the edges of the network.

A more agile source routing technique, *Software Defined Networking* (SDN), has been recently introduced [12]. SDN splits the control and forwarding functions of a network: an SDN controller node programs the network forwarding nodes, the SDN switches. The controller sends to the switches explicit rules that bind certain flows to their next-hops, thus creating virtual paths. In SDN, flow characterization is based on packet headers, hence to enable multipath transmissions the SDN nodes must somehow differentiate each flow based on a packet header field, for example, the UDP port or the RTP session ID. SDN does not support notifications from the SDN controller to the sender, therefore the SDN controller and the source must implicitly agree on the header fields to be used, which is suboptimal in a constantly changing environment, otherwise the SDN controller will not be able to distinguish substreams, which leads to the same problems as with MPLS routing.

III. Publish Subscribe Internet

Publish Subscribe Internet (PSI) is an Information Centric Network (ICN) architecture based on the *pub-sub* paradigm. Content providers are the *publishers* and content consumers are the *subscribers*. A piece of content is called an *information item* and is assigned a statistically unique identifier that is used for addressing that very content, regardless of its location. To maintain pub-sub's loose coupling, all content related requests, that is, content *publications* and *subscriptions*, are addressed to the network, in contrast to IP's end-to-end interaction. Therefore, the network undertakes request handling, information discovery and delivery via three core functionalities: Rendezvous, Topology and Forwarding. This clear separation of network functionalities creates an agile network architecture with optimized methods for managing information dissemination. In the context of this paper, we focus on the Topology and Forwarding functionalities, which allow content-based multipath QoS routing via centralized path formation and native source routing.

A. Topology functionality

In PSI, source routing is a native feature, hence the topology functionality involves the discovery of the appropriate dissemination routes between a publisher and one or more subscribers[1]. This operation is executed by a logical entity

[1]PSI allows native multicast transmission, but unicast is also available.

called the *Topology Manager* (TM) which, as implied by its name, is aware of the complete network, including link capacities, error-rates and propagation delays. The TM receives requests from the network's Rendezvous elements, formulates the appropriate transmission paths and sends them to the end-users for instant communication. A request for the TM's assistance also carries a "strategy" flag, which is set by the pub/sub users and defines a preferred dissemination pattern, such as multipath, multisource, unicast or other.

In addition to user requirements, the TM can exploit implicit or explicit intelligence for monitoring network needs. The TM can implicitly estimate the approximate available capacity on every link based on the history of past computed dissemination paths. Furthermore, the TM can explicitly monitor network state by receiving notifications from network routers. All in all, the TM is the best candidate for forming dissemination routes that satisfy both application and network needs.

The centralized nature of the TM raises questions about PSI's feasibility, since the TM must compute the data paths of all network connections. In [13] the authors have verified that an intra-domain TM is feasible and affordable by using a reasonable number of TM instances with precomputed paths. Moreover, since the vast majority of Internet traffic consists of video streaming, deploying dedicated TM instances to discover paths for video streaming related services only is intuitively an appealing solution.

B. Forwarding functionality

The forwarding functionality controls the delivery of information to the requesting hosts. PSI routers, also called *Forwarding Nodes* (FNs), statelessly forward incoming packets to their destination based on LIPSIN [14], a Bloom filter based technique for network routing. Under LIPSIN forwarding, each network link and node is assigned a unique *LIPSIN Identifier* (LID), that is, a fixed-size bitmap. A forwarding path is encoded into a *Forwarding Identifier* (FID), that is, the result of the OR operation of the LIDS assigned to the path links. The FID is placed on a packet's header and it is used by the FNs for determining the next hop. The forwarding decision is based on a binary AND and a comparison operation of the FID and the LIDs of the FN's attached links.

LIPSIN forwarding offers native source routing and multicast to PSI networks. The gains of multicast are well investigated, especially in throughput intensive services such as video streaming. Source routing allows service providers to apply QoS routing, as it guarantees that the selected paths will be used. Source routing is also important for multipath transmissions when path disjointness is required, since it allows using pre-computed disjoint paths. Disjoint paths are useful for many purposes, such as for reducing the impact of bursts of errors, maximizing bandwidth aggregation or, even, achieving TCP-friendliness towards unicast connections.

C. NMP in PSI

NMP in PSI can be supported in two ways: *one-to-one multipath* and *one-to-many multipath*. In the first approach, the TM computes multiple unicast paths among each pair of users and the publisher explicitly manages transmissions to each subscriber. In the second approach, the TM computes multiple multicast paths from each publisher to all subscribers and the publisher sends video to a multicast group.

- *One-to-one multipath*: Each user subscribes to the streams of each other musician, with the strategy flag set to multipath. The TM computes multiple paths among all users, encodes them in directed FIDs and sends a set of FIDs (one set per subscriber) to each publisher. Each such set of k FIDs provides k distinct *handles* for sending content to one subscriber, with overall throughput B and delay D. The publisher can then decide on the utilization of these handles based on explicit path feedback, such as *Real Time Control Protocol* (RTCP) reports.
- *One-to-many multipath*: Each user explicitly subscribes to a subset of the substreams produced by other musicians, possibly different for each user. The TM composes a multicast FID per video substream for each publisher. Each multicast FID encodes one tree that routes a substream's data from that publisher to all interested subscribers. Each publisher then sends each substream over the corresponding FID to all subscribers, without multiplexing the substreams among the FIDs.

The advantages of the one-to-many approach include a reduction in network load and in the computational cost of the publishers. Multicast is known to reduce the traffic footprint on the network [2], minimizing the bandwidth requirements of a service. Given that data travels only once up to each branching node of the multicast tree, bandwidth consumption can be reduced up to $n-2$ times, with n the number of musicians. In addition, treating users as multicast groups allows a static substream allocation among the same multicast FIDs for all publishers, thus unburdening NMP nodes from significant computational stress, that is otherwise induced by end-to-end traffic control schemes. If a recipient suffers from congestion, it will simply unsubscribe from some substreams, rather than operating a congestion control loop with the sender.

In contrast, the one-to-one approach minimizes path latency, allows more accurate adaptation to network conditions and improves load balancing. First, unicast connections optimize service delay, offering the least possible latency. Second, they allow explicit packet scheduling among the FIDs towards each subscriber, thus providing agile traffic control that dynamically adapts to network conditions and unburdens network operation, at the cost of operating individual control loops between the sender and each recipient. Finally, sending substreams via multiple paths per subscriber can enhance data spreading across the network, thus better balancing network load and increasing path richness.

At the moment, we do not strongly oppose any of these models or, even, a hybrid scheme. Nevertheless, we lean towards one-to-one multipath for NMP in particular, as this approach has the highest potential to reduce network delay.

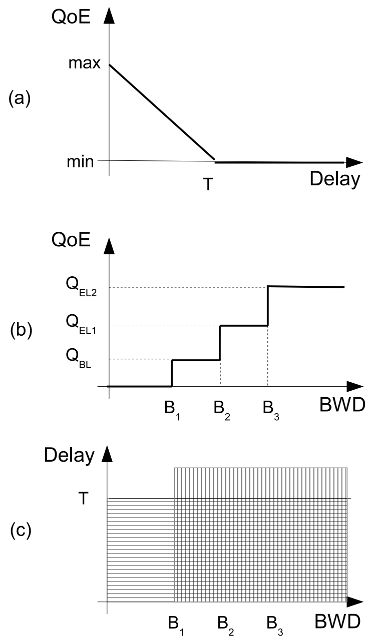

Fig. 1. Relationship of QoE with delay and bandwidth.

IV. Quality of Service and Quality of Experience

In order to provide an acceptable level of *Quality of Experience* (QoE) to NMP users,[2] the dissemination paths must provide enough bandwidth and low enough network latency. In media streaming, the service bandwidth is pseudo-additive, as it equals the capacity of the shared bottleneck of the overlapping paths or the sum of the capacities of the disjoint paths, while service delay is convex, as it equals the delay of the slowest among the selected paths. In this work, we assume that QoE is purely subject to these two constraints, a relationship illustrated in Fig. 1. Figure 1.(a) demonstrates QoE as a function of delay. We argue that delay and QoE are inversely proportional, since high latency degrades user synchronization. Additionally, we consider a time threshold T at which QoE is nullified; T is the point after which users become unsynchronized, for example, 50 ms for NMP applications. Note that we do not imply that the relationship between QoE and delay is linear; the exact nature of this relationship is however beyond the scope of this work.

Figure 1.(b) depicts QoE as a function of the available bandwidth. Assuming SVC streaming with three sub-streams, we mark three important bandwidth values B_1, B_2 and B_3, representing the throughput requirements for transmitting the base layer and two enhancement layers, respectively. Thereupon, if B_1 of bandwidth is available, the service scores the baseline QoE level Q_{BL}. Similarly, if B_2 or B_3 of bandwidth are available, then QoE reaches Q_{EL1} and Q_{EL2}, respectively. Note that B_1 is the service threshold, since below that point video transmission fails. Moreover, QoE does not improve beyond B_3, as the maximum required bandwidth is already consumed. Unlike Fig.1.(a) which implies a continuous relationship between QoE and delay, the relationship between QoE and bandwidth is a stepwise function, as shown in Fig.1.(b). In Fig. 1.(c) we summarize the bandwidth and delay constraints, annotating with a grid the range of values for which NMP is feasible. Based on the previous figures, the service delay t must be lower than T and the available bandwidth b must be higher that B_1. Essentially, the figure shows how the QoS metrics (delay and bandwidth) are related to the QoE.

[2] We distinguish here QoS which is what the network offers from QoE which is what the user perceives.

V. Evaluation

In this section we experimentally evaluate the gains in bandwidth augmentation and delay reduction that can arise from multipath routing with QoS constraints. For this purpose, we implemented unicast, multicast, single-constraint and multi-constraint routing algorithms in PSI's TM and we conducted a series of simulations measuring the average path throughput and latency offered by each algorithm.

A. Routing algorithms

We classify the routing algorithms studied below in two axes: the number of paths allowed (single or multiple) and the number of constraints supported (single or multiple). We first investigated three single-constraint algorithms, with different approaches to exploiting path richness. First, we used Dijkstra's shortest path algorithm, a single path routing algorithm that minimizes a certain additive metric, in our case path delay. Dijkstra's complexity in the worst case is $O(N^2)$, where N is the number of graph vertices. Then, we used Yen's algorithm which finds the k shortest unicast paths in terms of path delay [15]. Yen's algorithm allows overlapping paths and, similarly to Dijkstra's, has $O(N^2)$ worst case complexity. To explore the impact of path diversity, we considered a routing algorithm proposed by Apostolopoulos et al. [16] which includes a simple heuristic for enhancing path disjointness.

We then turned to multi-constraint routing algorithms. SAMCRA [17] is a multi-constraint single path routing algorithm that runs in polynomial time when the constraints/metrics are not real numbers. In our tests, we use SAMCRA to optimize bandwidth and delay, expressed as natural numbers, thus avoiding the exponential complexity resulting from non integers. We then considered DIMCRA [18], a multipath routing algorithm based on SAMCRA. DIMCRA supports the discovery of two edge-disjoint paths that optimize multiple constraints, such as bandwidth and latency.

B. Experiment setup

We implemented[3] the algorithms of Section V-A in our TM, simulating the functionality presented in [13], which relies on precomputing paths at system startup. The TM parses the topology and computes all the available dissemination paths among all access nodes[4] using every routing scheme.

[3] Our implementation is available at http://mm.aueb.gr/ under "software".
[4] An access node is a node with a link degree equal to one.

In order to evaluate the routing algorithms in realistic *Autonomous System* (AS) graphs, we created 50 synthetic topologies of scale-free graphs with 50 nodes that follow the Barabási-Albert model [19], with initial degrees $m_0 = 1$. We uniformly distributed link capacities and propagation delays. Although the uniform distribution is not a realistic approach, it provides a clearer picture of the relationship between our two metrics. In addition, we included random competing traffic to create further path diversity across the network. Specifically, we selected 50 random paths between random access nodes and we removed 25% of the first link's capacity from each link on that path. We repeated each experiment 50 times, so as to present results with a statistical error of less than 1%.

	Yen	Heuristic	mod DIMCRA
Path discovery success	0.85	0.85	0.65
Delay	1.25	1.51	1.56
Bandwidth	1.19	1.21	1.25

TABLE I
PERFORMANCE OF ROUTING ALGORITHMS NORMALIZED TO DIJKSTRA'S.

C. Single-constraint algorithms

First, we investigate the gains possible by exploiting path diversity. We tested four single-constraint routing algorithms, with delay as the metric: Dijkstra (single path), Yen (overlapping paths), Heuristic (overlapping but maximally disjoint paths) and modified DIMCRA (completely disjoint paths); modified DIMCRA uses Dijsktra's algorithm to find the two shortest disjoint paths instead of SAMCRA. We limited the number of computed paths by Yen's and Heuristic to two, to be comparable with modified DIMCRA. The results are presented in Table I, which shows the performance of the Yen, Heuristic and modified DIMCRA algorithms normalized to Dijkstra.

As shown in the first row of the table, while we can find two overlapping paths 85% of the time with Yen's or with the Heuristic algorithm, we can only find two disjoint paths 65% of the time with modified DIMCRA. Furthermore, as we make paths more disjoint, we increase total bandwidth (the capacity of the shared bottleneck of the overlapping paths or the sum of the capacities of the disjoint paths) but we also increase total latency (the delay of the slowest among the selected paths). These are both expected results, indicating that the bandwidth gains of disjoint paths come with measurable delay costs. We also note that Yen's algorithm offers the best tradeoff, as it exchanges a 25% increase in latency for a 19% bandwidth augmentation; the other algorithms increase delay far more than the increase in bandwidth.

D. Multi-constraint algorithms

We then investigate multi-constraint routing algorithms, expecting to improve the relative bandwidth-delay increase due to the disjoint paths. We used the per topology *average* path bandwidth and delay produced by single path Dijkstra as the *baseline* bandwidth and delay constraints for DIMCRA and SAMCRA. We studied two cases: the bandwidth constraint varying from 50% to 150% of the baseline, while the delay constraint is set to 100% of the baseline, and the delay constraint varying from 50% to 150% of the baseline, while the bandwidth constraint is set to 100% of the baseline. While single-metric algorithms do not support such constraints, we examined *a posteriori* their logs and analyzed the ratio of the paths produced by them that satisfy the constraints and, among those paths, the exact metrics produced.

Figure 2 depicts the performance of SAMCRA, DIMCRA, Dijsktra, Yen and the Heuristic algorithm for each metric, under the two cases mentioned above. The plots in the first column illustrate the ratio of paths discovered that meet the constraints, that is, the service feasibility ratio. In both scenarios, DIMCRA outperforms every other routing algorithm. However, in most cases, SAMCRA and Dijkstra perform close to DIMCRA. When the bandwidth or the delay constraint is set to 150%, Yen's takes the second place, since either the bandwidth constraint is too high to be satisfied by one path, or the delay constraint is so loose that the problem becomes widest-path discovery.

Considering only the *feasible* paths, that is, those meeting the constraints, the second and third columns analyze the delay and bandwidth of those paths. Since each algorithm performs differently in terms of service feasibility, these results are biased: the algorithms that scored poorly in path discovery, shape their performance by a smaller subset of paths. Nevertheless, the plots provide interesting indications about the behavior of the algorithms. Specifically, we see that the augmentation of one constraint does not necessarily impact the performance of the second. For instance, increasing the bandwidth constraint from 50% to 150% does not provoke significant delay changes. Similarly, augmenting the delay constraints from 50% to 150% does not affect bandwidth much. Only Yen's algorithm presents changes that can not be explained solely by the changes in the set of feasible paths: Yen's increases its throughput by almost 20% when the delay constraint loosens, signifying that the utilization of the second path is severely restrained by the low delay requirement. DIMCRA is not affected by this, because it does not deploy a second path when the first satisfies both constraints, hence its bandwidth score is not significantly affected by delay.

To sum up, DIMCRA is reasonably superior to any other investigated path selection algorithm, as it provides bandwidth aggregation but also manages to keep delay low. In contrast, SAMCRA does not offer much more than Dijkstra, meaning that without exploiting an extra route, the feasibility of NMP services is not measurably enhanced. Finally, the other multipath algorithms do provide substantial bandwidth aggregation, but with the cost of a non acceptable delay increase, therefore they are not appropriate for NMP applications.

VI. CONCLUSION AND FUTURE WORK

In this paper we have discussed the feasibility of implementing NMP services over the PSI architecture. Focusing on the exploitation of path richness via multipath dissemination routes, we argued that PSI is a suitable environment

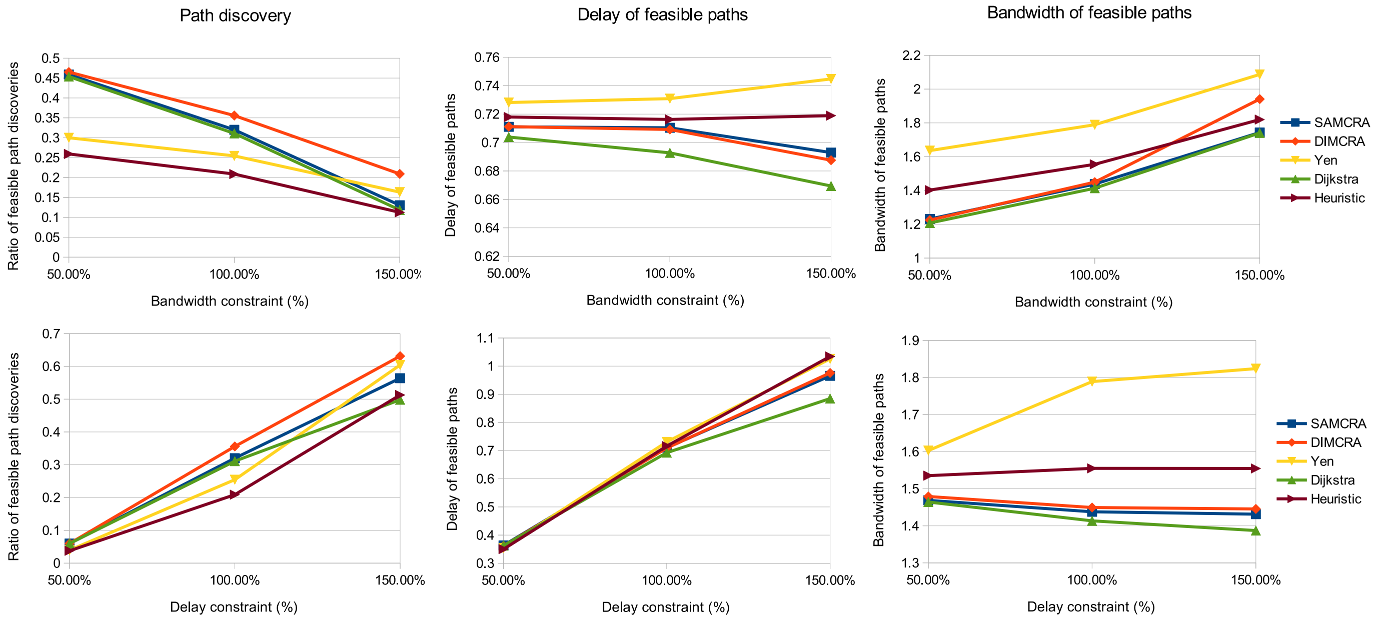

Fig. 2. Results with delay and bandwidth constraints. All results are normalized to the average performance of unconstrained Dijkstra.

for bandwidth-intensive time-critical services. We investigated several path selection schemes for improved performance. Finally, we evaluated those solutions through simulations and found that multiconstraint multipath QoS routing increases service feasibility by up to 15% compared to unicast solutions. In the future, we aim to investigate multipath routing for NMP in real topologies and actual WAN testbeds. At the same time, we intend to further compare PSI with IP source-routing forwarding schemes such as MPLS and SDN, in terms of the computational and storage costs imposed on forwarding nodes.

ACKNOWLEDGEMENTS

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS–MusiNet.

REFERENCES

[1] A. Renaud, A. Carôt, and P. Rebelo, "Networked music performance: State of the art," in *Proceedings of the AES 30th International Conference*, 2007.

[2] C. Stais, Y. Thomas, G. Xylomenos, and C. Tsilopoulos, "Networked music performance over information-centric networks," in *Proc. of the IEEE ICC Workshops*, 2013, pp. 647–651.

[3] M. F. Al-Naday, A. Bontozoglou, V. G. Vassilakis, and M. J. Reed, "Quality of service in an information-centric network," in *Proc. of the IEEE GLOBECOM*, 2014, pp. 1861–1866.

[4] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[5] N. Fotiou, D. Trossen, and G. C. Polyzos, "Illustrating a publish-subscribe internet architecture," *Telecommunication Systems*, vol. 51, no. 4, pp. 233–245, 2012.

[6] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Processing Magazine*, vol. 18, no. 5, pp. 74–93, 2001.

[7] L. Golubchik, J. C. S. Lui, T. F. Tung, A. L. Chow, W.-J. Lee, G. Franceschinis, and C. Anglano, "Multi-path continuous media streaming: What are the benefits?" *Performance Evaluation*, vol. 49, no. 1, pp. 429–449, 2002.

[8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[9] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," *Multimedia Systems*, vol. 10, no. 4, pp. 275–285, 2005.

[10] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming," *IEEE Communications Magazine*, vol. 42, no. 8, pp. 80–87, 2004.

[11] A. C. Begen, Y. Altunbasak, and O. Ergun, "Multi-path selection for multiple description encoded video streaming," in *Proc. of the IEEE ICC*, vol. 3, 2003, pp. 1583–1589.

[12] K. Kirkpatrick, "Software-defined networking," *Communications of the ACM*, vol. 56, no. 9, pp. 16–19, 2013.

[13] B. A. Alzahrani, M. J. Reed, J. Riihijärvi, and V. G. Vassilakis, "Scalability of information centric networking using mediated topology management," *Journal of Network and Computer Applications*, 2014.

[14] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "Lipsin: line speed publish/subscribe inter-networking," *Proc. of the ACM SIGCOMM*, pp. 195–206, 2009.

[15] J. Y. Yen, "Finding the k shortest loopless paths in a network," *anagement Science*, vol. 17, no. 11, pp. 712–716, 1971.

[16] J. Apostolopoulos, T. Wong, W.-T. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proc. of the IEEE INFOCOM*, vol. 3, 2002, pp. 1736–1745.

[17] P. Mieghem and F. Kuipers, "Concepts of exact quality of service algorithms," *IEEE/ACM Transactions on Networking*, 2004.

[18] Y. Guo, F. Kuipers, and P. Van Mieghem, "Link-disjoint paths for reliable QoS routing," *International Journal of Communication Systems*, vol. 16, no. 9, pp. 779–798, 2003.

[19] A.-L. Barabási, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A: Statistical Mechanics and its Applications*, vol. 281, no. 1, pp. 69–77, 2000.