Efficient Media Routing for Media over QUIC: Balancing Cost and Latency

Zoltán Szatmáry, Balázs Szenczy, and Tamás Lévai

***Abstract*—Media accounts for over 80% of global Internet traffic, much of it being latency-sensitive. The IETF's Media over QUIC (MoQ) protocol draft leverages QUIC for scalable media delivery, using a relay-based architecture similar to CDNs. However, current MoQ implementations rely on full mesh relay networks, leading to high network utilization and scalability issues.**

In this work, we present a novel approach to enforce media routing and lower the overall costs of operating a MoQ relay topology while satisfying delay service level objectives. We formulate the problem mathematically, analyze its complexity, and develop optimal and heuristic algorithms. We integrate our approach with a widely-used MoQ implementation. Our evaluation using different synthetic and realistic topologies shows improvements in cost efficiency while keeping all delay constraints satisfied.

***Index Terms*—Real-time systems, Content delivery networks, Streaming media, Multimedia communication, Routing**

I. INTRODUCTION

We use media-based solutions in almost every aspect of our digital lives, from voice and video calls to live streaming, cloud gaming, and beyond. This contributes significantly to the network traffic on today's large-scale networks. Some surveys show that media traffic puts more than 80% of all network traffic [1], therefore, the cost-efficient distribution of large-scale media is crucial for current and future networks.

A major challenge in large-scale media streaming over public networks like the Internet is ensuring liveness (e.g., conversation in a video call) and, in many use cases, enabling real-time interactions (e.g., user inputs in cloud gaming) due to the lack of control over the intermediary devices involved during the transmission of packets from source to destination. Moreover, applications vary in bandwidth requirements, latency constraints, and communication patterns.

Recently, Media over QUIC (MoQ) [2] emerged as a streamlined, low-latency media delivery solution. MoQ supports a variety of use cases and maintains a straightforward architecture that includes publishers, subscriber clients, and relays. Relays are responsible for transmitting the content between the publisher and subscribers. The relay network enables high scalability potential for systems using MoQ. Although MoQ advances quickly, it lacks critical features, such as the media routing capabilities necessary for large-scale and cost-efficient deployments.

Zoltán Szatmáry, Balázs Szenczy, Tamás Lévai are with the HSN Lab, Department of Telecommunications and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary.

Tamás Lévai is also affiliated to HUN-REN-BME Information Systems Research Group.

(correspondence: levait@tmit.bme.hu)

DOI: 10.36244/ICJ.2026.2.1

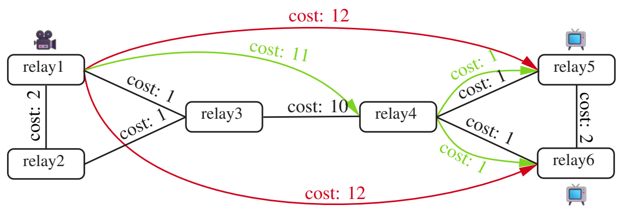

Figure 1. Topology of 6 MoQ relays: publisher connects to relay1, subscribers connect to relay5 and relay6; the naive approach (no media routing) costs 24, while cleverly routed media incurs 12 cost only.

To demonstrate the need for media routing, we present a MoQ network illustrated in Fig. 1. Relays are represented by boxes, with the publisher's relay marked with a camera, and the subscribers' relays marked with a television icon. Overlay connections are indicated by red and green arrows, while black lines represent underlay connections. We use overlay links to demonstrate IP routing, utilizing the same underlay links between two hosts but using OSI Layer 3 routing. In contrast, using underlay links enables routing media at Layer 7, leveraging protocol-defined tracks and groups. Media routing in Layer 7 can effectively utilize the underlay links (e.g., by path aggregation). The cost-benefit of media routing is evident even in this small example: with IP routing, using the red overlay links to send data to the two subscribers results in a cost total of 24. In contrast, with media routing, data is sent only once until `relay4`, where it is subsequently divided for the two subscribers, halving costs to 12 units. It is important to emphasize that in real-world provider networks, many more relays would likely be deployed to accommodate the large demand. However, our intention here is merely to illustrate the general concept of media routing in MoQ and its benefits, which remain clearly visible even in such a simplified network.

Intuitively the spectrum of possible solutions for MoQ media routing consists of two extremes: one minimizing the latency (provided by the state-of-the-art *no media routing* solution) and the other minimizing operational costs (provided by a Minimum Spanning Tree (MST) based optimization strategy that does not consider any delay bounds). In most cases neither of them suits our goals and a balance has to be found between the two. There is a trade-off: we have to compromise on one parameter to gain on the other, so the ideal solution to the problem should be flexible. While the main goal should be minimizing costs, when the delay bound cannot be satisfied via a certain link, we should switch strategies and adapt to the situation by falling back to using another link that may not be cost-optimal otherwise but conforms to the delay bounds.

Consequently, our goal with this paper is to enable latency-aware media routing in MoQ relay networks. Our contributions in this paper are as follows:

Analytical model. We present our take on modeling and optimizing MoQ relay topologies in §III. We show a mathematical formulation of the relay topology optimization problem.

Heuristics. We present a heuristic algorithm to solve the optimization in polynomial time, making it a good fit for real-life usage (§III).

Design, implementation, and evaluation. We extend the MoQ architecture with media routing capabilities. We present a practical implementation using the `moq-rs` implementation (§IV). In §V, we evaluate our model and heuristics numerically and demonstrate the effectiveness of our solution in a realistic testbed. Our implementation is available for download at [3]. We close the paper discussing related work (§VI) and deriving the main conclusions (§VII).

II. Media Routing in Media over QUIC

Media over QUIC offers a modern approach to media transmission, enhancing scalability and reducing latency by utilizing relays that streamline content distribution. This architecture makes it highly effective for dynamic and large-scale media applications. Yet, we argue that our problem with the cost-efficient and low-latency distribution of content in MoQ is closely related to other (usually video-based) media distribution problems. A ubiquitous problem that arises in virtually every content delivery system is that the centrality of the data has to be given up in order to provide better latency to users. MoQ builds upon the Content Delivery Networks (CDN) design distributing content across multiple servers (relays in MoQ) at several physical locations instead of relying on a single centralized server so that users can get the content from the nearest or otherwise optimal server, enabling low-latency distribution of on-demand content.

To utilize the whole potential of this architecture, MoQ requires media routing to optimize network usage and provide QoS. One well-known example of media routing is multicasting. Point-to-point unicast works well for many use cases; however, in cases where the same content needs to be distributed to multiple peers, this poses unnecessary duplications and thus increased utilization on the network. An early attempt at solving this came with IP Multicast. Although it has become a go-to solution for efficient media distribution on manged networks (e.g., for use in IPTV [4]), due to the lack of global support for IP Multicast protocols, it is unavailable for use on the public Internet [5]. Recognizing these constraints, Application Layer Multicast (ALM) emerged as a practical Layer 7 approach to bring multicast capabilities to networks without native IP Multicast support at Layer 3 [6].

Media routing solutions based on ALM provide the scalability needed for MoQ media distribution in large-scale networks. An ideal solution must also consider end-to-end (E2E) delays from publisher to subscriber to ensure service quality (i.e., QoS). Thus, our solution improves a state-of-the-art MoQ implementation by extending it with media routing capabilities to provide ALM and guarantee QoS.

MoQ relays are generic and contain no application logic by design [2]. Therefore, the media routing implementation requires an additional component (e.g., a routing API server) in the system that is aware of the QoS requirements, and can route media requests through the relay network accordingly. Take Fig. 1 as an example, subscriber at `relay6` subscribes to the track originally published at `relay1`. The router, overseeing the network, can detect that this content is already served to `relay5` via `relay4`, which is closer to `relay6`, so its routes `relay6` subscription to `relay4` thus *iteratively builds a distribution network for the track.*

By default, all relays in a network have one common API server to which they connect, and all clients know their next-hop relays. Before a publisher begins serving a track, its relay notifies this API server that the track will be available. When a subscription arrives at a relay, it checks whether the track is already available either locally (i.e., the publisher is directly connected) or remotely (i.e., an upstream subscription is already in place to a relay providing the track). If available, the relay acknowledges the subscription and begins forwarding the track as its objects arrive. Otherwise, it queries the API server to determine the appropriate relay to request the track from. The API server acts as a centralized controller, similar to a Software-Defined Networking (SDN) controller, and coordinates the creation of media forwarding paths through subscriptions between relays. Internally, it may use any method for path determination.

For example, `moq-api`, the API server of `moq-rs`, connects all relays in a full mesh topology. As a result, after consulting the API server, a subscriber's relay is always instructed to request the track directly from the publisher's relay. While this achieves distribution with minimal latency overhead, this method is not scalable.

Media routing decisions are made per relay, so the API server must know which relay is making the request to respond correctly. Since this information is not currently sent to the `moq-rs` API server, we modified the relays to address this limitation (§IV).

III. Modeling and Optimizing The Relay Topology

Media routing essentially defines a topology for each track, specifying the next-hop relay (i.e., the relay from which to request the track) for each relay involved in forwarding.

This section outlines the formalization of our media routing problem and describes various optimization methods (e.g., heuristic and ILP-based) that can be used by a media routing algorithm to determine the optimal topology, comparing their benefits and potential trade-offs.

Our take on modeling network topologies formed by relays resembles key characteristics of CDNs. We define relay topologies (or networks) as special graphs where the nodes represent relays and the links represent logical interconnections (over Layer 7) between each two nodes of the graph. These interconnections leverage the end-to-end connectivity provided by lower-level layers; therefore, when modeling links between relay nodes, a physical path (e.g., the shortest path) going through a series of routers and other network devices between two nodes is relaxed as a single logical link.

In our model, we abstract the fact that in MoQ, (original) publishers and (end) subscribers are not actually relays, but simple end devices that connect to (possibly to the nearest) relays. We can do this without loss of generality, since it is not a significant difference to real life. For this reason, we will refer to publishers and subscribers as their relays, in contrast to the terminology used in the IETF draft [2].

The media content in our model is organized into tracks. We strive to make an optimal topology for each track. However, for each track we only account for the potential traffic generated by the distribution of itself, which means that even when there are multiple tracks present, each of those will have a topology optimized independently of each other in complete isolation. This relaxation is acceptable under our conditions, due to the fact that we are working with logical links, of which there may be many possible mappings to disjoint paths of physical links; therefore, two independent tracks sharing the same logical link will likely not result in bandwidth constraints due to massive overprovisioning. Furthermore, studies have shown that while the capacity of certain fiber-optic systems is starting to saturate, proposed techniques to overcome this limitation [7] already exist [8].

A. The Optimization Problem

Unlike prior CDN cost models that optimize replica placement or request routing over a fixed server infrastructure [9], or ALM solutions that construct multicast trees in a distributed, one-shot manner [10], [11], our formulation incrementally builds per-track media forwarding trees as subscribers join, enforcing E2E delay bounds within MoQ's relay-based pub/sub model. Specifically, the goal is to establish an interconnection among relays between a track's publisher and its subscribers, minimizing the total operational cost while satisfying delay requirements. The optimization takes a relay network and one or more tracks as input. It produces an optimal forwarding topology for each track.

The relay network is given as a directed graph $G(V, E)$, with $v \in V$ being the relays and $e \in E$ the links between them. Each link carries a cost c_{ij} and a delay d_{ij} for $(i, j) \in E$. The cost is an abstract value encoding the relative preference for using a link as higher cost indicates a link the optimizer should avoid unless necessary. Delay values are physically measurable quantities, though in practice they may be approximations of the true path delay (e.g., based on propagation delay only).

Each track t is given with four attributes: its publisher denoted by $P^t \in V$, its set of subscribers $S^t \subseteq V$, a mapping of roles R_f, and a delay bound $D^t \in \mathbb{N}$ assigned to it.

The roles of the publisher and the subscriber in a track are formalized by streams (or flows) indicated by $f \in F^t$, each with a path from the publisher of a subscriber across G, and with a vector R_f which encodes roles of the relays with regard to that flow. Therefore, for each $f \in F$ and $i \in V$, R_i^f is set to -1, if $i \in P^t$, +1, if $i \in S^t$, or 0 otherwise.

The delay bound is the maximum tolerated E2E delay between the publisher and each subscriber, where the E2E delay is measured independently for each subscriber by adding the delay values along the path between the publisher and the subscriber in the topology.

To measure link utilization, we introduce two other variables: $x_{ij}^f : (i, j) \in E, f \in F^t$ which represents the theoretical "transmission bitrate" (measured in units) of stream f on link (i, j); and $y_{ij}^t : (i, j) \in E$ that represents the link usage of the track on (i, j) for each $t \in T$. To include the delay bound, we add an additional variable $z_{ij}^f \in \{0, 1\}$ to indicate whether the link (i, j) is on the path of stream $f \in F^t : \forall t \in T$.

The objective function is characterized as the sum of link costs in the topology for each track $t \in T$: $\sum_{t \in T} \sum_{e \in E_t} c(e)$. The solution is an optimal topology $G_t(V_t, E_t) : V_t \subseteq V, E_t \subseteq E$ for each track $t \in T$. A topology is optimal if *1)* the operation of the topology is cost-optimal and minimizes extra network usage, and *2)* meets the E2E delay requirements.

Putting it all together, the Integer Linear Program that satisfies these requirements is the following:

$$\begin{array}{lrr} \min & \displaystyle\sum_{t \in T} \sum_{(i,j) \in E} c_{ij} y_{ij}^t & \\ & y_{ij}^t \geq x_{ij}^f & \forall (i,j) \in E, \forall f \in F^t : \forall t \in T \\ & \displaystyle\sum_{j:j,i \in E} x_{ji}^f - \sum_{j:i,j \in E} x_{ij}^f = R_i^f & \forall i \in V, \forall f \in F^t : \forall t \in T \\ & x_{ij}^f \geq 0 & \forall (i,j) \in E, \forall f \in F^t : \forall t \in T \\ & y_{ij}^t \geq 0 & \forall (i,j) \in E, \forall t \in T \\ & z_{ij}^f * M \geq x_{ij}^f & \forall (i,j) \in E, \forall f \in F^t : \forall t \in T \\ & \displaystyle\sum_{i,j \in E} z_{ij}^f * d_{ij} \leq D^t & \forall f \in F^t : \forall t \in T \end{array}$$

This problem is referred to as the Bounded Diameter Minimum Steiner Tree (or Bounded Diameter Steiner Minimum Tree [12]) problem [13], which is known to be NP-hard.

B. Heuristics

The ILP-based solution is complex to solve. It is NP-hard (see [14]), and for a considerably small network with only 5 to 10 active relays its execution time takes up minutes, making it limited for real deployments.

Due to the long processing times, we propose a simple heuristic approximation to find close-to-optimal routes for media streams over the relay network. We opted for a dynamic approach that does not require full recomputation of the topology when new subscribers join. This design enhances real-world applicability and potentially improves computational speed. We employed an algorithm that follows a greedy expansion strategy with local refinements leading to close-to-optimal solutions with modest computational overhead.

The full mesh nature of the input graph allowed us to leverage the constrained MST relaxation [10], which enables to efficiently compute (locally) optimal connections to the distribution tree for new subscribers. The intuition behind the algorithm is simple and can generally be described with two main ideas. *1)* For each subscriber, we try to find a link that connects the subscriber to the existing distribution tree in a way that satisfies the delay constraints but is the least expensive among such links. If such a link exists, use it to create the new subscriber to the tree; otherwise, the problem is infeasible. *2)* See if there is a way to reduce costs by reaching another subscriber with a link that goes from this newly added

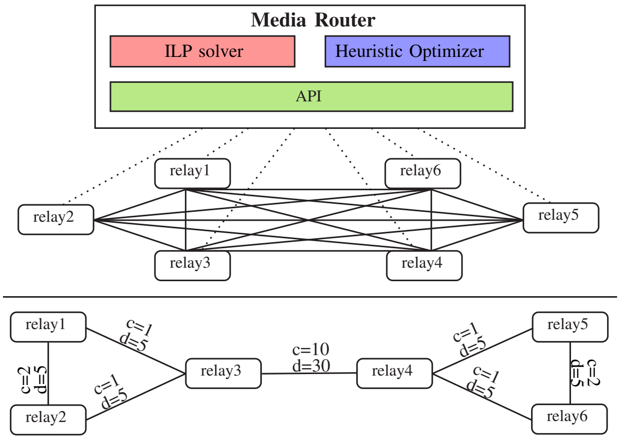

Figure 2. Architecture: optimizer and the relay network (top) with the underlay network (bottom).

subscriber. If there is, choose the one with the lowest cost and the least E2E delay.

Notice how this algorithm can be easily converted to work with dynamically added subscribers. We call this the *Heuristic Media Routing* in later sections. The total complexity of the heuristic algorithm is $O(n^3)$.

IV. Implementation

Based on our theoretical findings, we extended MoQ with media routing capabilities that make media distribution cost-efficient and latency-aware. To confirm our results and assess the feasibility of using our model as a media routing extension to MoQ, we devised two interdependent solutions. Firstly, we created an optimizer prototype that implements the integer linear program and the *Heuristic Media Routing* algorithm described in §III. Secondly, we enhanced `moq-rs` to enable support for dynamic topology construction. Next, we will outline the high-level architecture along with the primary design decisions considered. The source code is publicly available on GitHub [3].

A. Architecture

We strove to make all control in our system centralized, so that relays are fully coordinated. In Figure 2, we can see how the Media Router acts as a single source of truth controlling every relay in the system, resembling the widely-used SDN architecture [15].

The Media Router component consists of two subcomponents: the topology optimizer and the API that provides access to the former through HTTP endpoints.

The API server shares the same endpoints as the upstream `moq-api`. When the relay consults the API (e.g., upon a new subscription to a track that is not currently available at the relay), instead of using the original `moq-api`, an endpoint on the Media Router's API subcomponent is invoked, triggering an immediate recalculation of the optimal topology upon request.

The Topology Optimizer determines the topology for distributing a specified track throughout the network. It takes as input a network containing all system relays, along with the track for which the topology needs to be optimized. The Topology Optimizer employs multiple strategies, including the state-of-the-art No Media Routing solution, an Integer Linear Program solver, the *Heuristic Media Routing*, and a Minimum Spanning Tree-based approach to find an optimal or a near-optimal topology.

V. Evaluation

Our evaluation goals were twofold: *1)* to compare the theoretical complexities of the different solvers presented to their empirical performance; and *2)* to demonstrate the usability of our system with realistic components. We test our theoretical solutions on a benchmark to see how they scale in a simulated environment. Then, we use a Mininet-based testbed to see how the system behaves when using realistic components. Our evaluation tools are available on GitHub [3].

We used a computer with an AMD EPYC 7502P (32 CPU cores) and 128 GB RAM, running Ubuntu 22.04 with Python 3.10 and Rust 1.77.2 installed. We switched to performance mode and fine-tuned network configuration parameters.

A. Benchmarks (Numerical Evaluation)

For numerical evaluation of the media routing algorithms, we construct a large-scale realistic network we call *Azure-GÉANT*: we place MoQ relays at various Microsoft Azure Regions worldwide using approximate geographical locations [16], and we connect these relays via the links of the GÉANT research network to simulate the underlay of a large-scale network [17]. The GÉANT map consists of cities connected by links grouped into regions. We leveraged this for cost assignments: interregional links received a cost equal to the total number of cities in the network, while intraregional links had a cost of 1. To map the underlay to the overlay, we first linked each relay to its nearest city with zero-cost "last-mile" connections. Next, we formed a full-mesh overlay by connecting each unique relay pair through the shortest path between their corresponding cities, assigning delay values based on straight-line distance and costs by summing the costs of the underlay links used.

To compare the complexity of the different optimizers at scale, we performed simulations on *Azure-GÉANT*. In each simulation run, we have measured the total time taken to run the optimization process, as well as the total cost and the maximum E2E latency present in the resulting topology. The optimizers included in the benchmarks were: (1) No Media Routing (baseline, the state of the art); (2) Heuristic Media Routing; (3) Integer Linear Program (ILP); and (4) Minimum Spanning Tree (MST). Optimizers (2) and (3) have been well described in §III, (1) and (4), on the other hand, are implementations corresponding to the two extreme points of the delay-cost spectrum introduced in §I which we have included for comparison.

We have set up the simulation with a total of 40 peers geographically located in multiple continents, of which an ever-increasing subset is selected for the track (with the first

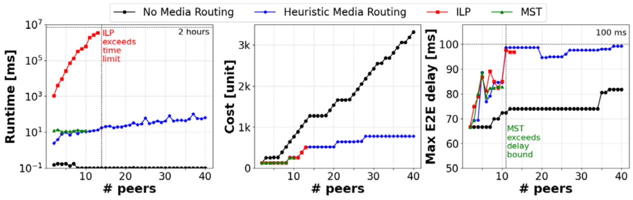

Figure 3. Numerical Evaluation: Optimizer performance metrics for a single track with 100 ms delay bound over an increasing number of peers.

peer included in the subset being the publisher and every other a subscriber) in each test run of the simulation.

In Figure 3, we can see the results as the number of peers increases. Note that for better visualization of orders of magnitude, we have plotted the runtime values on a logarithmic scale. When an optimizer cannot produce a suitable topology adhering to the delay bound within a 2 hour time limit, the plotting of that optimizer ends. We have denoted these breakpoints with vertical lines, and specified rationales.

No Media Routing is the state of the art for MoQ implementations. Compared to other algorithms, it is a simple algorithm that provides the best delays on the cost of massive resource usage (4× increment in costs with 40 peers). MST cannot satisfy the 100 ms delay bound when the number of peers exceeds 11, which confirms that naively optimizing for costs may not always be sufficient. The ILP follows an exponential runtime curve due to its NP-completeness, and just after 14 peers, its optimization process could not finish in a 2 hour time frame. Conversely, the Heuristic Media Routing produces outputs well within the time limit thanks to its $O(n^3)$ algorithmic complexity. It can effectively minimize costs while ensuring the delay bound.

A staircase pattern, observable in all optimizers, arises from the discrete jump in both cost and delay when a subscriber is reached via an inter-regional link rather than an intra-regional one. This is a structural property of the *Azure-GÉANT* topology, where inter-regional costs are set to N (total city count), rather than 1 for intra-regional links. This figure also shows that the idea of the delay-cost spectrum (§I) is present in practical scenarios as well.

B. Real-life Measurement in a Testbed

To demonstrate the usability of MoQ media routing and our optimizer, we recreated the network of Fig. 1 using the Mininet [15] network emulator. We use our `moq-rs` version (recall §IV). The *Original* method is the current no media routing behavior, while the *Optimized* includes the media route optimizer. The optimizer gives similar routes as in Fig. 1. Our measurements focused on two metrics: end-to-end (E2E) delay and network usage. Table I shows the results of 5 consecutive runs. We see a slight increase in E2E delays due to the extra relay in the path. The benefit of media routing is apparent in the network usage of the relay network. With the optimization, the system transmits 13.5 MB data instead of 40.7 MB (appr. 67% saved).

To conclude our evaluation, the results show that our media routing algorithm scales well. Optimized media routing can reduce link usage effectively. Still, in some cases, it might add minimal delays, making them a practical choice for scalable media routing for MoQ.

TABLE I

TESTBED: PERFORMANCE COMPARISON OF THE NO MEDIA ROUTING (ORIGINAL) AND THE LATENCY-AWARE MEDIA ROUTING (OPTIMIZED) CONFIGURATIONS IN THE NETWORK OF FIG. 1: SLIGHT EXTRA DELAY DUE TO THE EXTRA RELAY, BUT SIGNIFICANT SAVING IN RESOURCE USAGE BY PATH AGGREGATION.

Metric	Method	Mean	Min	Max
End-to-end delay [ms]	Original	45.65	45.50	45.852
	Optimized	**48.94**	48.72	49.117
Network usage [MB]	Original	40.73	40.56	40.876
	Optimized	**13.56**	13.52	13.60

C. Limitations and Scope

The current formulation optimizes each track independently, which is well-founded when logical links map to disjoint physical paths (a reasonable assumption in overprovisioned backbone networks). However, in congested deployments where multiple high-bandwidth tracks compete for the same physical bottleneck, per-track independence may underestimate resource contention, requiring joint multi-track optimization. This is a natural extension of the present work.

The testbed evaluation is intentionally limited to the six-relay topology shown in Fig. 2, which clearly illustrates the core cost-saving mechanism. The numerical benchmark on *Azure-GÉANT* validates scalability up to 40 peers per track; evaluating concurrent multi-track scenarios at that scale is future work. We note, however, that the per-track heuristic's $O(n^3)$ complexity holds regardless of the number of tracks, so the computational argument for real-world deployability is unaffected.

Finally, the heuristic is designed for incremental subscriber arrivals and does not perform a full topology recomputation due to churn. In highly dynamic settings with rapid subscriber joins and leaves, the maintained topology may degrade over time. Periodic reoptimization or churn-aware extensions are straightforward directions that we leave for future work.

VI. RELATED WORK

Media over QUIC offers a modern approach to media transmission, enhancing scalability and reducing latency by using relays that streamline content distribution, making it well-suited for dynamic, large-scale media applications [18], [19] and beyond [20]. The importance of relay topology design for MoQ has been recognized within the IETF itself [21], and standardized relay benchmarking frameworks are an emerging direction [22]. Before MoQ, prior solutions included HTTP Live Streaming (HLS), Dynamic Adaptive Streaming over HTTP (DASH) [23], [24], WebRTC [25], and Real-Time Messaging Protocol (RTMP) [26]. Each of these focused on a subset of media transmission use cases; none of them can be applied universally. For example, while WebRTC is the prevailing solution for delay-sensitive, live media transmission over the Internet, it is commonly criticized for being difficult to scale and adapt to application-specific requirements. This is because QoS-affecting parameters, such as behavior under packet loss or insufficient throughput, are mostly baked into the protocol and the browser implementations. HLS and DASH, in contrast, rely on HTTP's request-response model, in which clients explicitly request individual media segments. While this approach scales well for static content distribution, such as Video on Demand, it introduces latency that limits its suitability for interactive applications. Finally, RTMP is primarily designed for media contribution from publishers to streaming servers rather than for large-scale content distribution. MoQ aims to become a unifying standard covering the use cases of all of these. While MoQ represents a novel integration of multiple techniques, many of its underlying concepts have been explored in earlier work. Our problem with the cost-efficient and low-latency distribution of content in MoQ is closely related to existing media distribution problems. Next, we survey prior work that motivates and contextualizes our approach.

The general principle of content delivery systems is to give up on the centrality of data to achieve better latency for users [27]. Content Delivery Networks (CDNs) work by distributing content across multiple servers at several physical locations, instead of relying on a single centralized server, so that users can access content from the nearest or otherwise optimal server. The main benefits of CDNs are caching, cost-efficient operation (due to only having to move large volumes of data once) and low-latency distribution of content [9].

Another relevant approach is multicasting, which involves simultaneous transmission of the same data to one or multiple destinations [28]. Media delivery is, in fact, a particularly common application of multicasting. A prime example of media multicasting is IPTV [4]. The main problem with multicasting, however, is that it has no direct support in the Internet Protocol, and therefore, implementing Layer 3 multicasting in IP networks requires specialized protocols such as Protocol Independent Multicast and Internet Group Management Protocol. Thus, multicast is mostly limited to private, managed networks with appropriate equipment supporting the required protocols and configured accordingly (e.g., telco providers' IPTV networks).

Consequently, neither CDN nor IP Multicast is sufficient on its own for live media distribution over the Internet since Content Delivery Networks are tailored for Video on Demand use cases, and IP Multicast is constrained to managed networks.

Before our work, numerous studies proposed shifting multicasting from the network layer to the application layer, Application Layer Multicasting (ALM), in which multicast topologies are constructed as overlay networks atop large-scale infrastructures such as the Internet. A notable example is peer-to-peer (P2P) streaming networks, which have traditionally relied on ALM by design. One well-known implementation of P2P streaming is the Narada protocol, which realizes the End System Multicast (ESM) architecture [29]. Narada constructs the overlay network in a self-organizing manner, using network metrics obtained through periodic heartbeat measurements between end systems participating in multicast content delivery.

In contrast, other approaches such as Topology Aware Grouping (TAG) [11] incorporate knowledge of the underlying physical network into overlay multicast routing. Although such topology information is not always available, it proves highly valuable when it is. Without it, peers participating in media delivery must conduct their measurements to infer the network topology, which limits scalability. Therefore, this ability to leverage existing topology information is a key differentiator between TAG and ESM.

A solution combining the strengths of CDNs and ALMs has the potential to provide an efficient distribution model for live media on Internet-scale networks. For this reason, we built on top of the idea of topology-aware overlay networks. Still, our goals were different regarding how and where the routing should occur. For example, we aimed for centralized control, similar to SDN networks, rather than the inherently distributed TAG, ESM, and other P2P streaming solutions. Second, while TAG is core-based, our extension of the MoQ API server ensures that relays contact only their next-hop relays within a single track. Finally, many of the aforementioned ALM solutions explicitly tie topology construction to a specific metric, such as bandwidth; however, our solution can use any cost function as long as it satisfies the triangle inequality.

More recent works have applied similar ALM ideas to WebRTC-based systems. For example, Silva *et al.* propose a hybrid architecture combining P2P multicast trees with a centralized control plane [30], foreshadowing the SDN-inspired centralized routing approach we adopt for MoQ.

VII. Conclusions

Media over QUIC is a novel initiative for large-scale media distribution, which uses a network of relays for transmission. Current implementations use full mesh relay topology, leading to high network utilization and limited scalability.

In this paper, we presented a novel approach to enforce media routing by optimizing the MoQ relay topology based on service delay requirements. Our solution lowered the overall costs and enabled high scalability. We formulated the media routing optimization problem and introduced heuristic routing algorithms. We integrated our approach with a widely-used MoQ implementation. We demonstrated that our solution improves cost-efficiency while meeting service delay bounds in synthetic and realistic environments.

Future work will address joint multi-track optimization, larger-scale multi-stream evaluation, and churn-aware heuristic extensions.

Acknowledgment

Special thanks to Gábor Rétvári, Felicián Németh, István Pelle, and Gergely Pongrácz for the discussions that led to this line of work. We thank Ericsson Hungary for providing the infrastructure to run our measurements. We also express our gratitude to Luke Curley and the supportive `moq-rs` community.

REFERENCES

[1] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach (8th Edition)*, 8th ed. London, UK: Pearson, 2020.

[2] S. Nandakumar, V. Vasiliev, I. Swett, and A. Frindell, "Media over QUIC Transport," Internet Engineering Task Force, Internet-Draft draft-ietf-moq-transport-11, Apr. 2025. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-moq-transport/11/

[3] HSN Lab, "Github repository: Optimizer and testbed," https://github.com/hsnlab/moq-relay-topo-opt, 2025.

[4] S. Zeadally, H. Moustafa, and F. Siddiqui, "Internet protocol television (iptv): Architecture, trends, and challenges," *IEEE Systems Journal*, vol. 5, no. 4, pp. 518–527, 2011.

[5] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *IEEE network*, vol. 14, no. 1, pp. 78–88, 2000.

[6] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 3, pp. 58–74, 2007.

[7] C. Xie and B. Zhang, "Scaling optical interconnects for hyperscale data center networks," *Proceedings of the IEEE*, vol. 110, no. 11, pp. 1699–1713, 2022.

[8] G. Rademacher, B. J. Puttnam, R. S. Luís, J. Sakaguchi, W. Klaus, T. A. Eriksson, Y. Awaji, T. Hayashi, T. Nagashima, T. Nakanishi *et al.*, "10.66 peta-bit/s transmission over a 38-core-three-mode fiber," in *optical fiber communication conference*. Optica Publishing Group, 2020, pp. Th3H–1.

[9] B. Zolfaghari, G. Srivastava, S. Roy, H. R. Nemati, F. Afghah, T. Koshiba, A. Razi, K. Bibak, P. Mitra, and B. K. Rai, "Content delivery networks: State of the art, trends, and future roadmap," *ACM Comput. Surv.*, vol. 53, no. 2, Apr. 2020.

[10] Q. Zhu, M. Parsa, and J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum-cost multicasting," in *Proceedings of INFOCOM'95*. New York, NY, USA: IEEE, 1995, pp. 377–385 vol.1.

[11] M. Kwon and S. Fahmy, "Topology-aware overlay networks for group communication," in *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, 2002, pp. 127–136.

[12] K.-H. Vik, P. Halvorsen, and C. Griwodz, "Evaluating steiner-tree heuristics and diameter variations for application layer multicast," *Computer Networks*, vol. 52, no. 15, pp. 2872–2893, 2008.

[13] A. Mashreghi and A. Zarei, "When diameter matters: parameterized approximation algorithms for bounded diameter minimum steiner tree problem," *Theory of Computing Systems*, vol. 58, no. 2, pp. 287–303, 2016.

[14] R. M. Karp, *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, 1972, pp. 85–103.

[15] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM HotNets*. New York, NY, USA: ACM, 2010, pp. 1–6.

[16] Microsoft, "Microsoft datacenters," https://datacenters.microsoft.com/globe/explore, 2025, accessed: 2025-05-04.

[17] G. Association, "GÉant connectivity map," https://map.geant.org/, 2024, accessed: 2024-10-31.

[18] Z. Gurel, T. Erkilic Civelek, A. Bodur, S. Bilgin, D. Yeniceri, and A. C. Begen, "Media over quic: Initial testing, findings and results," in *Proceedings of the 14th ACM Multimedia Systems Conference*, ser. MMSys '23. New York, NY, USA: ACM, 2023, p. 301–306.

[19] A. C. Freeman, "Scalable event-based video streaming for machines with moq," in *Proceedings of the 4th Mile-High Video Conference*, ser. MHV'25. New York, NY, USA: ACM, 2025, p. 60–66.

[20] S. Nandakumar, "SPACE - Scalable Pubsub, Asymmetric and CachEd transport for Deep Space communications," Internet Engineering Task Force, Internet-Draft draft-nandakumar-deepspace-moq-00, Oct. 2024, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-nandakumar-deepspace-moq/00/

[21] H. Shi *et al.*, "Design Space Analysis of MoQ," IETF, Tech. Rep., Oct. 2022, internet-Draft draft-shi-moq-design-space-analysis-of-moq-00.

[22] T. Evens, T. Rigaux, and S. Henning, "Media over QUIC Relay Benchmark Methodology," *IETF, Tech. Rep.*, Nov. 2025, internet-Draft draft-evens-moq-bench-00.

[23] R. Pantos and W. May, "HTTP Live Streaming," RFC 8216, Aug. 2017. [Online]. Available: https://www.rfc-editor.org/info/rfc8216

[24] International Organization for Standardization, Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats, ISO/IEC Std. 23009-1:2014, May 2014.

[25] N. Blum, S. Lachapelle, and H. Alvestrand, "Webrtc: real-time communication for the open web platform," *Commun. ACM*, vol. 64, no. 8, pp. 50–54, Jul. 2021. [Online]. Available: DOI: 10.1145/3453182

[26] H. Parmar and M. Thornburgh, "Adobe's real-time messaging protocol (rtmp)," Adobe Systems Incorporated, Tech. Rep., 2012.

[27] A. Popescu, D. D. Kouvatsos, D. Remondo, and S. Giordano, "Content distribution over ip: developments and challenges," in *Network Performance Engineering: A Handbook on Convergent Multi-Service Networks and Next Generation Internet*. Heidelberg, DE: Springer, 2011, pp. 979–987.

[28] L. Peterson and B. Davie, "Computer networks: A systems approach - multicast," https://book.systemsapproach.org/scaling/multicast.html, 2025, accessed: 2025-05-07.

[29] Y.-h. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on selected areas in communications*, vol. 20, no. 8, pp. 1456–1471, 2002.

[30] M. A. Silva, R. Bertone, and J. Schäfer, "Topology distribution for video-conferencing applications," in *2019 10th International Conference on Networks of the Future (NoF)*. IEEE, 2019, pp. 90–97.

Zoltán Szatmáry is an MSc student in Computer Engineering at the Budapest University of Technology and Economics (BME). His research interests include network softwarization, data center networking, and IT security.

Balázs Szenczy has received his MSc degree in Computer Engineering from the Budapest University of Technology and Economics (BME) in January 2025. He currently serves as a DevOps Engineer at Nokia. His professional focus includes cloud-native networking solutions, containerization, and automated deployment pipelines. While maintaining an industry position, he actively follows developments of Media over QUIC and its community.

Tamás Lévai received the MSc in Computer Engineering at the Budapest University of Technology and Economics (BME) in 2016. He received his PhD degree in Informatics in 2022 at BME. Currently, he is an Assistant Professor at BME, and a Research Fellow at HUN-REN TKI. His research interest focuses on computer networks and distributed computing, mainly software-defined networking, cloud native computing and real-time communications.