# Demonstration of Viewport-Aware Hybrid Broadcast-Unicast Streaming for Volumetric Video

Casper Haems, Matthias De Fré, Tim Wauters, Filip De Turck
*Department of Information Technology*
*Ghent University - imec*
Ghent, Belgium
casper.haems@ugent.be, matthias.defre@ugent.be, tim.wauters@ugent.be, filip.deturck@ugent.be

***Abstract*—Real-time streaming of volumetric video requires low latency and high bandwidth, making it hard to scale efficiently with unicast delivery alone. This work presents a multi-path transport framework that combines broadcast and unicast to enable scalable, bandwidth-efficient delivery of interactive volumetric content. A base layer is transmitted via broadcast using File Delivery over Unidirectional Transport, while enhancement layers are fetched on demand via Dynamic Adaptive Streaming over HTTP based on the viewer's viewport. The approach reduces redundant transmissions and enables fine-grained adaptation without compromising interactivity. An end-to-end implementation is demonstrated with real-time viewport tracking, dynamic quality switching, and live monitoring under varying bandwidth constraints, improving reliability, achieving sub-40 ms latency and reducing per-client unicast load by 18 Mbps (15%) per object at the highest quality.**

***Index Terms*—Volumetric video, hybrid broadcast-unicast, multi-path transport, real-time streaming, immersive media**

## I. Introduction

Real-time immersive applications such as holographic conferencing and free-viewpoint telepresence require high data rates and ultra-low latency [1]. However, unicast-based immersive streaming does not scale efficiently to large audiences and often struggles under network impairments. To address this challenge, a multi-path delivery framework for real-time volumetric media is introduced. This architecture decouples base and enhancement content across broadcast and unicast paths, reducing redundancy and enabling personalized, bandwidth-aware adaptation. The demonstration focuses on a hybrid instance of this approach, using broadcast via File Delivery over Unidirectional Transport (FLUTE) to deliver a base scene and unicast via Dynamic Adaptive Streaming over HTTP (DASH) for viewer-specific enhancement layers. Viewport-aware logic activates higher-quality representations when a relevant object enters the user's field of view (FOV), ensuring efficient bandwidth use without compromising perceived quality. The end-to-end pipeline supports real-time visualization, offering both a first-person view and a global overview with position tracking and visualization of the user's frustum (i.e., visible 3D region). A live dashboard displays system performance, including latency, bandwidth, and point cloud resolution. Network conditions are configurable at runtime within Mininet, allowing for stress-testing and validation of adaptive behavior [2]. Figure 1 summarizes the system. The setup is open source and supports reproducible experimentation[1].

## II. Related Work

Research on real-time immersive media delivery spans protocol design, adaptive systems, and hybrid architectures. Conventional streaming protocols such as DASH and HTTP Live Streaming (HLS) introduce multi-second delays due to segment buffering and Transmission Control Protocol (TCP) overhead [3], [4]. Even low-latency variants such as Low Latency DASH (LL-DASH) and Low-Latency HLS (LL-HLS) remain above 1 second [5]. In contrast, Web Real-Time Communication (WebRTC) and Media over QUIC Transport (MOQT) use User Datagram Protocol (UDP) to achieve sub-second delivery, though MOQT remains in early stages of adoption [6], [7]. Volumetric streaming systems such as six degrees-of-freedom (6DoF) DASH and WebRTC-based conferencing platforms adapt content to the user's position and FOV, but rely on unicast-only delivery [8], [9]. Hybrid transport architectures reduce per-user load by broadcasting shared content while unicasting personalized refinements [10]. A prior demonstration in 2D streaming used FLUTE with Forward Error Correction (FEC) to reduce latency and bandwidth [11]. However, no open demonstration exists for real-time hybrid delivery of volumetric media. This demonstration bridges that gap with a novel and unique reproducible system that combines broadcast and unicast transport for immersive content, featuring real-time adaptation based on viewport dynamics.

## III. System Architecture

This system demonstrates the hybrid multi-path architecture introduced by Haems et al. [1]. The system follows a modular architecture that supports real-time volumetric media streaming over multiple transport paths with multiple protocols. Content can be partitioned based on quality layers, object semantics, or spatial regions, enabling flexible mappings onto broadcast and unicast delivery channels. This design generalizes to various immersive use cases, including scalable telepresence and interactive multi-user scenarios. Most relevant components are implemented in Rust[2] using a multi-threaded design with protocol-agnostic interfaces. Each transport protocol connects to a shared pipeline, allowing different delivery mechanisms to

1. https://github.com/idlab-discover/Multi-path-XR
2. https://www.rust-lang.org/

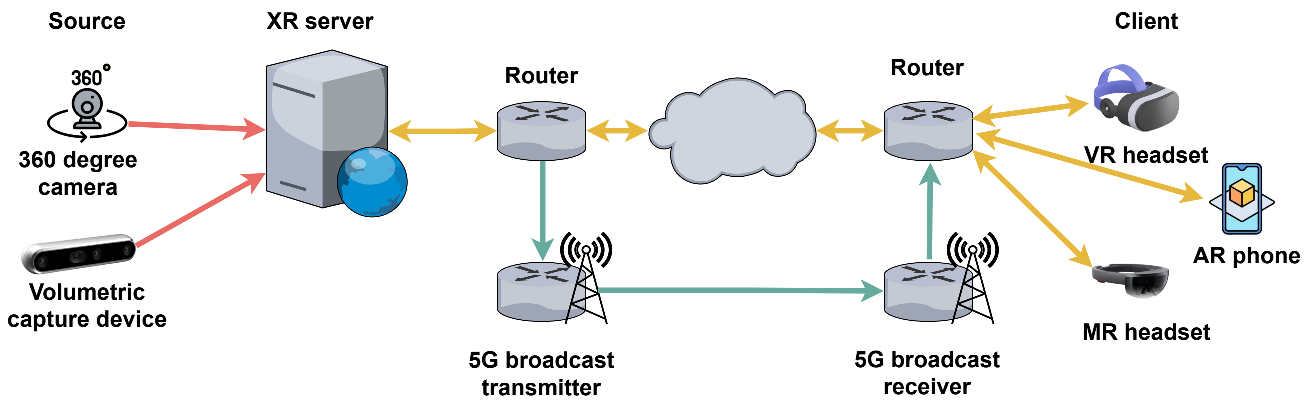


Fig. 1. Overview of a hybrid end-to-end delivery system for real-time volumetric media. Content is captured using either a 360° camera or a volumetric capture system generating point cloud video. After processing on an XR server, the media are distributed using two complementary paths: a conventional Internet Protocol broadband network and a 5G broadcast channel. The resulting scene can be rendered on Virtual Reality headsets, Mixed Reality devices, or smartphones with Augmented Reality capabilities.

be combined without altering the core logic. The modular architecture supports runtime adaptation, content prioritization, and protocol switching. On the receiver side, decoded content is exposed through a shared interface, enabling integration with diverse rendering clients. For instance, a headless receiver can support analytics or storage, while a graphical front-end may render the scene in real time. The system separates content acquisition, delivery, and rendering concerns, allowing individual components to evolve independently and be reused across deployment contexts.

## IV. Demonstration

The demonstration showcases real-time volumetric streaming using a hybrid unicast-broadcast delivery pipeline and provides quantitative evaluation metrics. A base scene with two objects is transmitted over FLUTE using IP multicast for efficient broadcasting, while one of the objects is enhanced on demand via DASH based on the viewer's viewport. All components run on a single host with network constraints emulated using Mininet.

### A. Content and Adaptivity Setup

The demonstration uses the `longdress` and `red and black` objects from the MPEG Point Cloud Compression (PCC) dataset. In this demonstration, only the `longdress` object is enhancement-enabled. Each object consists of 300 frames, corresponding to 10 seconds of content at 30 frames per second (FPS), which is played in a continuous loop. All frames are encoded using the Draco codec[3]. Each object is downsampled to 100k points per frame. Without considering network protocol overhead, this results in approximate bitrates of 92 Mbps for `longdress` and 84 Mbps for `red and black`. To enable adaptivity, each object is split into three non-overlapping subsets of 60k, 25k, and 15k points, following a Multiple Description Coding (MDC) structure. This setup allows the client to compose additional *virtual qualities* from one object by combining multiple subsets. For example, streaming both the 15k and 25k subsets yields an effective 40k quality, while combining all three restores the original 100k quality. These combinations provide finer granularity without requiring additional encoding. In the hybrid scenario, two delivery paths are used. Per object, the 15k base layer is transmitted over FLUTE, ensuring that all receivers obtain a minimal representation. Enhancement layers are delivered over DASH as 25k and 60k subsets, as well as a pre-merged 85k subset that combines the 25k and 60k subsets. When displayed with the 15k broadcasted base layer, these enhancement layers enable adaptive qualities up to 40k, 75k, or 100k, depending on viewport visibility and available bandwidth. Figure 2 shows the unicast throughput for the `longdress` object under varying bandwidth constraints. The FLUTE stream for the 15k baseline (`longdress` only) consumes roughly 18 Mbps and is constant. Depending on the available unicast bandwidth and selected quality, the hybrid setup (in red) reduces DASH throughput by up to 18 Mbps (15%) compared to a pure unicast scenario (in blue). Figure 3 displays the corresponding end-to-end latency. While the currently used DASH implementation lacks smoothing and does not coordinate with the FLUTE stream, the setup exhibits stable behavior. Quality switching remains slightly aggressive due to basic adaptivity logic, but hybrid delivery achieves lower unicast load and comparable or improved latency.

### B. Interactive Behavior and Viewport Logic

The demonstration streams the enhancement-enabled `longdress` object while also streaming a second object (`red and black`) that is not enhancement-enabled. This contrast allows users to observe how the system responds when the hybrid `longdress` object enters the viewport, triggering the DASH enhancement stream, while the `red and black` object remains at the 15k base quality regardless of visibility. The client renders the scene from two synchronized

3. https://google.github.io/draco/

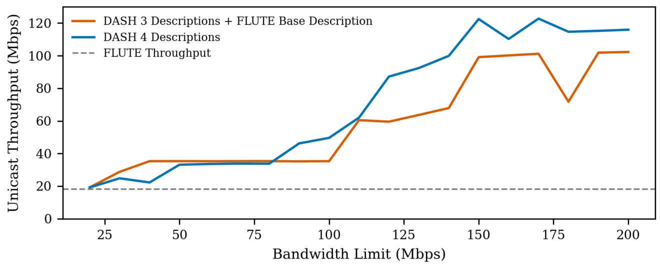


Fig. 2. Average unicast throughput as a function of the unicast bandwidth cap. The 18 Mbps FLUTE stream for 15k of the `longdress` object is shown as a reference.

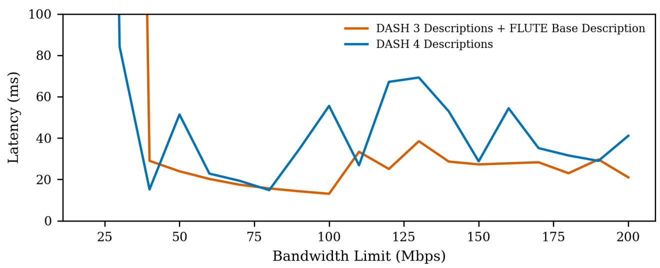


Fig. 3. Average latency as a function of unicast bandwidth cap. Latency is influenced by the selected representation and available capacity.

perspectives: a first-person view (user viewport) and a third-person overview that displays the user's position and viewing frustum. An example is shown in Figure 4. Viewport tracking determines which objects intersect with the user's frustum. When the enhancement-enabled object becomes visible, the DASH player fetches additional quality layers to augment the FLUTE base. Once the object exits view, the DASH stream is paused, reverting to the baseline.

### C. Monitoring and Adaptation

A real-time dashboard visualizes key performance metrics, including latency (typically under 40 ms), bandwidth usage per protocol, displayed point count per object (15k-100k), and system resource usage. Bandwidth constraints and player viewport can be changed during runtime to demonstrate the system's adaptability. Under reduced bandwidth conditions, the DASH enhancement layer is disabled, maintaining a consistent user experience through FLUTE fallback. The system achieves sustained framerates of 30 FPS with network latencies below 40 ms. Under constrained emulated links, the DASH stream disables while the FLUTE layer remains visible without interruption.

## V. Conclusion

This demonstration presents a real-time hybrid delivery framework for volumetric video that leverages the complementary strengths of broadcast and unicast. By decoupling a reliable base layer from user-specific enhancements and enabling viewport-aware activation, the system achieves sub-40 ms latency while reducing unicast bandwidth and efficiently adapting to bandwidth variability. The setup illustrates how even under constrained or fluctuating conditions, users maintain a seamless immersive experience. Live monitoring provides visibility into the system's performance, offering a transparent and reproducible environment for further research. The modular design supports multiple protocols and runtime reconfiguration, making it a flexible open source platform for future experimentation in scalable XR delivery, adaptive streaming, and hybrid network architectures.

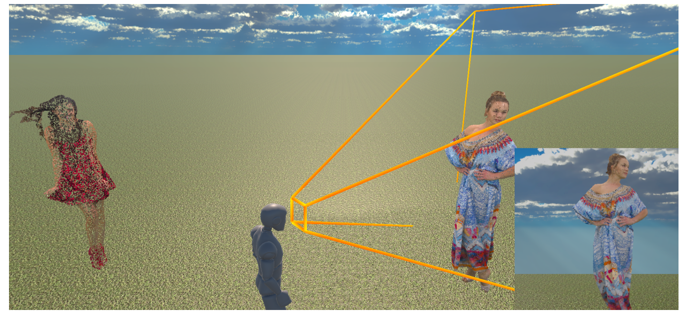

Fig. 4. Screenshot showing the dual visualization: global overview with the player and their frustum, with the user view on the bottom right. Content that is in the user's viewport is enhanced using DASH.


## Acknowledgment

This work has been funded by the European Union (SPIRIT project, Grant Agreement 101070672, https://www.spirit-project.eu/).


## References


[1] C. Haems et al., "Towards efficient transport for Real-Time immersive applications over hybrid networks," in *NoF '25*, 2025. [Online]. Available: https://spirit-project.eu/wp-content/uploads/sites/85/2025/07/Towards-Efficient-Transport-for-Real-Time-Immersive-Applications-over-Hybrid-Networks.pdf

[2] N. Handigol et al., "Reproducible network experiments using container-based emulation," in *Proc. ACM CoNEXT '12*, 2012, pp. 253–264.

[3] ISO, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," Tech. Rep., 2022. [Online]. Available: https://www.iso.org/standard/83314.html

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

[5] Z. Gurel et al., "Media-over-QUIC transport vs. low-latency dash: a deathmatch testbed," in *Proc. ACM MMSys '24*, 2024, pp. 448–452.

[6] B. Jansen et al., "Performance evaluation of webrtc-based video conferencing," *SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 3, pp. 56–68, 2018.

[7] L. Curley et al., "Media over QUIC Transport," 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-moq-transport/03/

[8] J. van der Hooft et al., "Towards 6DoF HTTP adaptive streaming through point cloud compression," in *Proc. ACM MM '19*, 2019, pp. 2405–2413.

[9] M. De Fré et al., "Scalable MDC-based volumetric video delivery for real-time one-to-many WebRTC conferencing," in *Proc. ACM MMSys '24*, 2024, pp. 121–131.

[10] C. Haems et al., "Hybrid unicast-broadcast video delivery for scalable low-latency live streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, 2025.

[11] ——, "Real-time demonstration of low-latency video delivery via hybrid unicast-broadcast networks," in *CNSM '24*, 2024, pp. 1–3.