Real-Time Demonstration of Low-Latency Video Delivery via Hybrid Unicast-Broadcast Networks

Casper Haems*
Jeroen van der Hooft*
Hannes Mareen†
* *Department of Information Technology*
† *Department of Electronics and Information Systems*
Ghent University - imec
Ghent, Belgium
casper.haems@ugent.be
jeroen.vanderhooft@ugent.be
hannes.mareen@ugent.be

Peter Steenkiste
Departments of Computer Science and Electrical and Computer Engineering
Carnegie Mellon University
Pittsburg, United States of America
prs@cs.cmu.edu

Glenn Van Wallendael†
Tim Wauters*
Filip De Turck*
† *Department of Electronics and Information Systems*
* *Department of Information Technology*
Ghent University - imec
Ghent, Belgium
glenn.vanwallendael@ugent.be
tim.wauters@ugent.be
filip.deturck@ugent.be

***Abstract*—In response to the growing demand for low-latency video streaming, this paper presents a demonstration of a hybrid unicast-broadcast video delivery system that combines 5G terrestrial broadcasting with over-the-top (OTT) streaming methods. The demonstration features a scalable setup with an interactive dashboard, allowing users to experiment with various configurations and observe key metrics such as bandwidth usage, packet loss, buffer size, and live latency in real-time. Key techniques include Low-Latency DASH (LL-DASH) for HTTP Adaptive Streaming (HAS), packet recovery (PR) and Forward Error Correction (FEC) for reliability, Temporal Layer Injection (TLI) for enhanced quality, and Common Media Application Format (CMAF) with Chunked Transfer Encoding (CTE) for reduced latency. The demonstration shows that this scalable hybrid approach can effectively reduce unicast bandwidth to nearly 0 Mb/s in scenarios without packet loss on the broadcast network, and achieve similar bandwidth reductions in lossy broadcast networks with appropriate Forward Error Correction (FEC) settings, while maintaining a live latency lower than 1 second. These results demonstrate the system's potential for optimizing multimedia delivery, significantly reducing unicast bandwidth while maintaining low-latency streaming.**

***Index Terms*—Video delivery, Multimedia streaming, 5G terrestrial broadcast, Low-latency**

I. Introduction

Live multimedia content delivery is shifting from traditional broadcast television to over-the-top (OTT) streaming, straining existing internet infrastructure and necessitating novel adaptive video streaming approaches [1], [2]. The advent of 5G and upcoming 6G technologies enables efficient video streaming to large audiences across vast geographical areas through innovations such as High-Power High-Tower (HPHT) networks and 5G-enabled satellites [3], [4]. This paper presents a demonstrator setup that showcases the benefits of hybrid unicast-broadcast video delivery, focusing on improvements in latency, bandwidth efficiency, and robustness under various network conditions as previously discussed in [5]. The source code has been made available[1]. This paper reviews related work, describes the system architecture, details the demonstration setup, and concludes with a summary of findings and future research directions.

II. Related Work

Modern multimedia content delivery predominantly relies on hypertext transfer protocol (HTTP), with HTTP Adaptive Streaming (HAS) being the standard method. Efforts to optimize protocols such as Dynamic Adaptive Streaming over HTTP (DASH) and HTTP Live Streaming (HLS) focus on improving adaptivity and reliability, and reducing latency, often overlooking bandwidth-reducing hybrid streaming approaches [6], [7]. Studies on low-latency video delivery over 5G networks highlight potential advancements in content delivery systems [8]. However, comprehensive evaluations considering bandwidth and latency metrics from hybrid live stream experiments are scarce. Research on broadcast and multicast protocols such as File Delivery over Unidirectional Transport (FLUTE) highlights their efficiency and reliability for file delivery over broadcast networks [9], [10]. This paper aims to bridge the gap between existing solutions and comprehensive evaluations, providing insights into the combined effects of these techniques on bandwidth, latency, and adaptability.

III. System Architecture

The architecture is illustrated in Figure 1. At the core of the system is the server, serving as the central repository of multimedia content. It supports both unicast and broadcast transmission methods. Unicast delivers content directly to individual proxies, while broadcast simultaneously transmits to multiple proxies using FLUTE over Internet Protocol (IP) multicast, conserving network resources. Clients, positioned

This work has been funded by Flanders Innovation & Entrepeneurship (5G BROADCAST, Grant Agreement HBC.2021.0674, https://www.imec-int.com/en/research-portfolio/5g-broadcast)

[1]https://github.com/idlab-discover/5GBDash

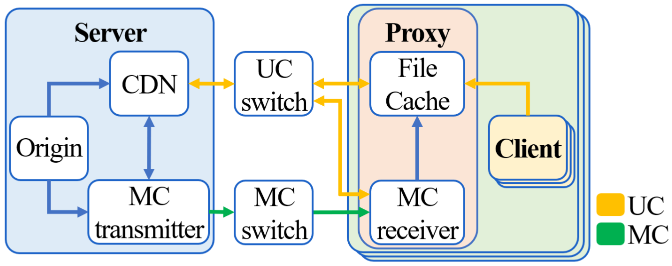

Fig. 1. The system architecture [5]. Multicast is used for emulating a broadcast network.

at the receiving end, stream the content and request video segments sequentially as needed. Proxies act as intermediaries between the server and clients. They enhance content delivery efficiency and reliability by caching content that was broadcast or fetched from the server. This reduces the need to retrieve segments from the server over unicast, minimizing latency and bandwidth consumption. To address challenges in bandwidth consumption, network robustness, quality adaptation, and latency reduction, the system leverages several important techniques. packet recovery (PR) improves stream reliability by retransmitting lost packets over unicast [5]. Forward Error Correction (FEC) is used to prevent the need for PR by adding redundant data that allows recovery from packet loss without retransmissions [11]. Temporal Layer Injection (TLI) enhances video quality by injecting temporal layers to provide better visual fidelity [12]. Common Media Application Format (CMAF) with Chunked Transfer Encoding (CTE) reduces latency by using chunked video segments, enabling faster start times, lower latency and minimized buffering [13], [14]. More detailed and technical aspects of the system architecture are described in the previous work [5].

IV. Demonstration

This demonstration showcases the hybrid unicast-broadcast video delivery system in action. It operates within a Mininet environment [15], comprising a server, up to 5 proxies, and up to 5 clients per proxy, each as separate nodes. The clients are headless Dash.js players [16], which can be made visible for demonstration purposes. Through this emulation, the system's performance can be thoroughly evaluated under various network conditions, such as bandwidth limitations, packet loss, and latency. Attendees can interact with a dashboard, shown in Figure 2, enabling manual experimentation and displaying key metrics to tune the evaluation scenarios. Users can compare live data with previously recorded data on metrics such as multicast and unicast bandwidth, packet loss, buffer size in a Dash.js player, and total live latency. There are controls for adjusting bandwidth, latency, and packet loss in real time on the multicast link, and configuring the number of proxies and the number of clients per proxy. Users can also display or hide video streams from up to two clients, enable or disable FEC, TLI, and low latency, and set the live latency goal. Additional controls include configuring settings such as selecting which video files to stream, determining the quality representation sent over multicast, setting segment duration, specifying chunk count, and choosing the number of video files to multicast from the input list. Buttons allow opening additional graphs and a media player for side-by-side viewing of two quality representations of the same video, highlighting the effects of TLI. Enabling TLI in the demonstration will enhance stream quality by injecting temporal layers while still reducing unicast bandwidth compared to a non-broadcast scenario. The dashboard metrics include multicast, unicast, and total bandwidth, bytes transmitted by the server over multicast and unicast, multicast packet loss, client buffer size, and live latency. More detailed metrics can be shown in a popup, such as the number of frames dropped during playback, the quality index of the current segment, and the number of times a PR was initiated. An example scenario, shown in Figure 2, involves running an experiment with only unicast for five proxies with one client each, followed by an experiment where multicast is enabled. This demonstrates the reduction in traffic over unicast. With 10% packet loss and no FEC, the setup needs to recover 10% of the multicast traffic times the number of proxies over unicast, but a significant reduction in unicast traffic is observed. Figure 3 shows the unicast bandwidth for the same experiment with different losses and number of proxies and clients. The demonstration also includes a comparison of segment durations and their impact on live latency. By comparing a 4-second segment stream with a 1-second segment stream, attendees will observe the trade-offs between latency and bandwidth usage. Enabling CMAF on the 4-second segments divides them into smaller chunks, reducing live latency to below 1 second while optimizing bandwidth efficiency. This scenario highlights the effectiveness of using CMAF chunks with CTE to achieve low latency and efficient bandwidth usage. Attendees will have the opportunity to switch between the configurations using the dashboard controls. Real-time graphs will display the live latency, bandwidth usage, and other relevant metrics for each configuration, emphasizing the advantages of using CMAF-CTE to optimize both latency and bandwidth. Attendees can expect to see the system's resilience to packet loss and how it handles various network conditions. Custom scenarios can be set up on request to test specific conditions and demonstrate the system's performance.

V. Conclusion

The demonstration of low-latency hybrid unicast-broadcast video delivery underscores the potential of combining OTT streaming methods with 5G terrestrial broadcasting for efficient multimedia delivery. Through an interactive dashboard, users can experiment with different configurations and observe real-time metrics such as bandwidth usage, packet loss, buffer size, and live latency. The real-time comparisons and metrics showcased in the demonstration provide valuable insights into the performance and usability of this hybrid network solution. The demonstration highlights several key techniques, including PR, FEC, TLI, and CMAF, which together address challenges in bandwidth consumption, network robustness, quality adaptation, and latency reduction. Users are able to

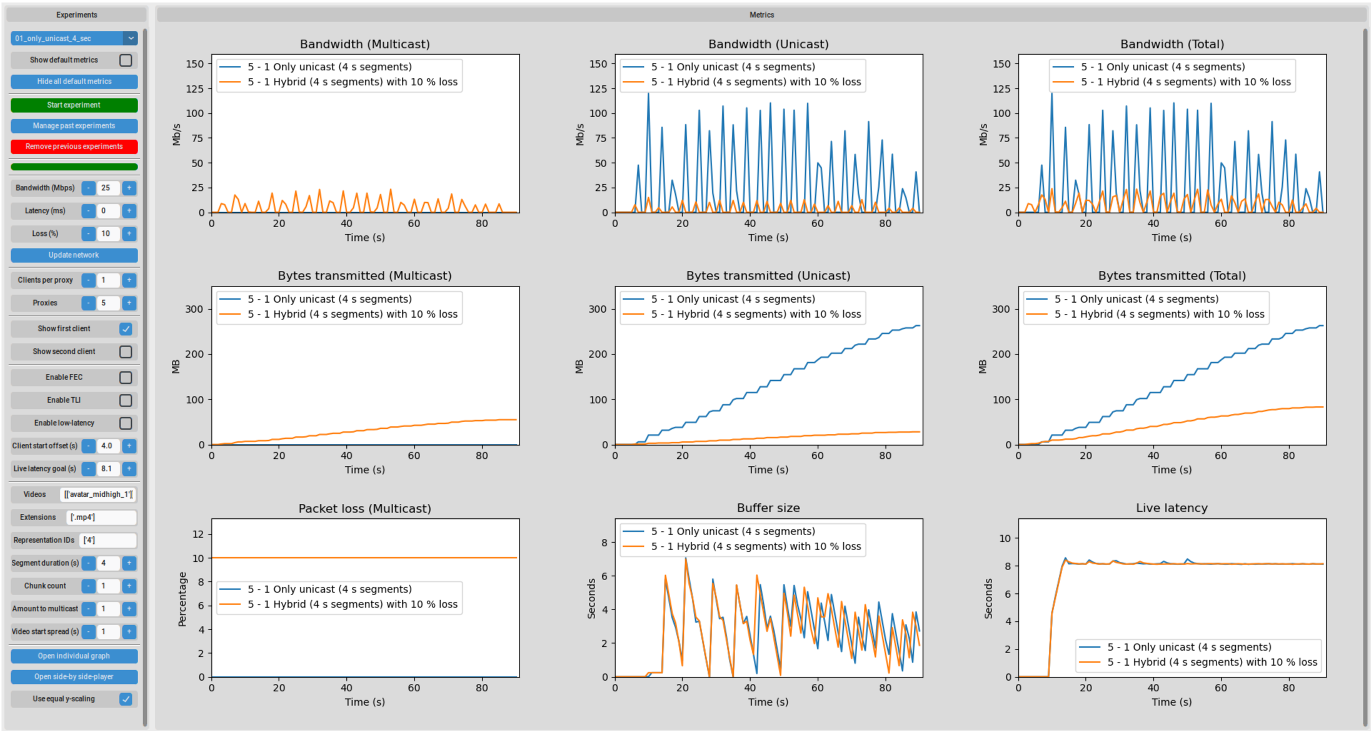

Fig. 2. Dashboard for interactive demonstration and experimentation. The top graph in the middle shows the unicast bandwidth, clearly illustrating a significant bandwidth decrease with the hybrid setup compared to a unicast-only approach, even with 10% loss and no FEC. Other metrics displayed include multicast bandwidth, packet loss, buffer size in the Dash.js player, and total live latency, providing a comprehensive view of the system's performance.

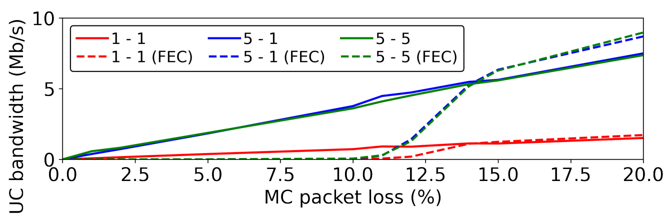

Fig. 3. Impact of packet loss on the required unicast bandwidth while multicasting one video [5]. The first number in the legend stands for the number of proxies, while the second number stands for the number of clients per proxy.

see firsthand how these techniques improve the streaming experience. The results from the demonstration show that the approach can reduce unicast bandwidth to nearly 0 Mb/s in scenarios without packet loss on the broadcast network, and achieve similar reductions in lossy broadcast networks with appropriate FEC settings. This significantly enhances scalability, as the bandwidth requirement remains constant regardless of the number of clients or proxies. Additionally, the use of PR and the proxy setup ensures high reliability. CMAF allows for low latency with sub-1-second live latency, and TLI enhances the video quality without significant bandwidth increase. Future work will focus on developing mechanisms that dynamically select which live streams to broadcast based on viewer interest, and on broadcasting in real-time extended reality environments.

References

[1] M. Shon, D. Lee, and J. H. Kim, "Are global over-the-top platforms the destroyers of ecosystems or the catalysts of innovation?" *Telematics and Informatics*, vol. 60, p. 101581, 2021.

[2] M. Franke, J. Holland, and S. Schmid, "Mcquic – a multicast extension for quic," 2023.

[3] J. Dobruna, E. Spahiu, M. Pogačnik, and M. Volk, "5g streaming: Ip-based vs. high-power high-tower broadcast," in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2022, pp. 1507–1511.

[4] M. Corici et al., "Satellite is 5g," nov 2020.

[5] C. Haems et al., "Enabling adaptive and reliable video delivery over hybrid unicast/broadcast networks," in *Proceedings of the 34th Edition of the Workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV '24. ACM, 2024, p. 29–35.

[6] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[7] R. Pantos and W. May, "HTTP Live Streaming," RFC 8216, Aug. 2017.

[8] Broadcast Networks Europe, "5G Broadcast Trials in Europe," September 2023, accessed on May 29, 2024. [Online]. Available: https://broadcast-networks.eu/wp-content/uploads/5G-broadcast-trial-leaflet.pdf

[9] T. Paila, R. Walsh, M. Luby, V. Roca, and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport," RFC 6726, Nov. 2012.

[10] EBU, "Tech report 063 - 5g broadcast network planning and evaluation," August 2021, accessed on 29 may 2024. [Online]. Available: https://tech.ebu.ch/docs/techreports/tr063.pdf

[11] V. Roca, M. Watson, and A. C. Begen, "Forward Error Correction (FEC) Framework," RFC 6363, Oct. 2011.

[12] H. Mareen et al., "Temporal layer injection for fast bitrate ladder creation in video live streaming," in *2023 IEEE ISM*, 2023, pp. 44–51.

[13] ISO, "Information technology - Multimedia application format (MPEG-A) - Part 19: Common media application format (CMAF) for segmented media," Tech. Rep., 2024.

[14] R. T. Fielding, M. Nottingham, and J. Reschke, "HTTP/1.1," RFC 9112, Jun. 2022.

[15] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. ACM, 2012, p. 253–264.

[16] DASH-IF, "Dash reference client," 2024, accessed on May 29, 2024. [Online]. Available: https://reference.dashif.org/dash.js/