Neighbor Selection Strategies in the Wild for CDN/V2V WebRTC Live Streaming: Can we learn what a good neighbor is?

Zhejiayu Ma, Soufiane Rouibia, Frederic Giroire, Guillaume Urvoy-Keller

Neighbor Selection Strategies in the Wild for CDN/V2V WebRTC Live Streaming: Can we learn what a good neighbor is?

Zhejiayu Ma*†, Soufiane Rouibia*, Frédéric Giroire†, Guillaume Urvoy-Keller†
*EasyBroadcast, Nantes, France
{ma.zhejiayu, rouibia}@easybroadcast.fr
†Université Côte d'Azur, CNRS, Sophia-Antipolis, France
{frederic.giroire, guillaume.urvoy-keller}@univ-cotedazur.fr

***Abstract*—A hybrid CDN/Viewer-to-Viewer (V2V) architecture is an attractive solution for HTTP (HLS) and MPEG-DASH-based live streaming providers. It combines a traditional CDN with a V2V overlay for exchanging video fragments, reducing the cost of the CDN while maintaining the quality of experience. This work explores machine learning models to address the key challenge of neighbor selection. Our goal is to predict the connection quality between two arbitrary viewers using features such as locality, access providers, operating systems, past CDN, and V2V throughput. The proposed solutions are validated using an A/B testing approach on our production system, demonstrating a significant improvement in key system metrics compared to the traditional locality-based methods. We observe 17% higher V2V throughput, 26% lower delay, 37% fewer lost chunks, 39% fewer re-buffering, and 20% fewer quality switches.**

***Index Terms*—hybrid P2P, live streaming, peer selection, machine learning, real deployment, A/B testing**

I. Introduction

The video streaming traffic represents an increasing fraction of Internet traffic with a compound annual growth rate of 34% between 2017 and 2022, according to Cisco [1]. Live streaming and VoD (Video On Demand) are two common ways to stream video online. With the rise of HLS[1] and DASH[2], PC and mobile users benefit from the convenience of HTTP-based streaming protocol for day-to-day entertainment. One important challenge is maintaining the quality of the service despite this increasing audience. Content delivery networks (CDNs) are a cornerstone component provided by many companies such as Akamai, Amazon, Google, and Azure [2]–[5]. However, the cost of maintaining CDN servers or purchasing services from providers becomes high as the audience grows [6]. To further reduce the costs, a viewer-to-viewer (V2V) overlay has been introduced into the live streaming and over-the-top (OTT) services with the maturing of WebRTC. WebRTC allows data exchange between browsers, and it greatly facilitates the deployment of such hybrid systems. With the help of WebRTC, V2V traffic can offload a significant fraction of the traffic from the CDN servers, consequently reducing the operational cost. Examples of such services or systems are: Limelight [7], Akamai, LiveSky [8], or EasyBroadcast [9].

The overlay construction is a demanding task that has been broadly investigated in the context of Peer-to-Peer (P2P) networks[3]. The dominant approaches are tree-based [10]–[12], mesh-based [13, 14], and tree-mesh overlays [15]. A tree-based overlay is known for its low overhead and its difficulty handling load fluctuations, such as flash crowd and peer churn, that can break the diffusion tree. Alternatives are mesh-based and hybrid overlays. In a mesh-overlay, neighborhoods are constructed at random. Many studies have been proposed to improve this baseline strategy, e.g. morphing the overlay into a hybrid tree-mesh overlay [15, 16] or mesh overlay organization including hierarchical structure creation, or locality-awareness [17]–[21].

Constructing an effective V2V overlay for a hybrid CDN/V2V live streaming system, however, bears unique challenges. Firstly, the traditional P2P live streaming requires the chunks to be majorly provided by the peers [12, 22, 23]. In contrast, in a hybrid system, the diffusion tree is shallower because the missing chunks can always be fetched from the CDN. Secondly, one essential characteristic of live streaming is that users need to be synchronised with the playback head. Hence, during a short period of time (typically less than 10s), the majority of users seek the same set of video chunks.

The *immediate* neighborhood thus highly determines the efficiency of the V2V system. A key challenge is then to predict the performance between one viewer and its *immediate* neighbors and use this information to select neighbors. Legacy approaches focus on proximity metrics such as the same ISP, city, country, or the smallest geographical distance. We compare these strategies with a machine learning (ML) approach in this work.

Our contributions are the following: (1) We proposed different ML models to predict the throughput between two viewers. These models are based on different information such as locality, ISP, operating systems, past CDN throughput, and

This work has been supported by the French government through the UCA JEDI (ANR-15-IDEX-01) and EUR DS4H (ANR-17-EURE-004) Investments in the Future projects, and by Inria associated team EfDyNet.

[1] Apple's HTTP Live Streaming (HLS)

[2] Dynamic Adaptive Streaming over HTTP

[3] We interchangeably use the notion of peer or viewer, where a viewer is a WebRTC client that can be seen as a special type of peer.

978-3-903176-40-9

V2V throughput. (2) We implemented these models in an operational hybrid CDN/V2V live streaming web application with tens of thousands of viewers per day in North Africa and Europe. *To the best of our knowledge, we are the first to investigate the behavior of neighbor selection strategies in a real-world hybrid CDN/V2V live streaming setup, as well as their impact on QoE.* (3) We tested and compared the different peer selection strategies (random, locality-based, and ML-based) in production during several A/B testing campaigns. These deployments demonstrate that (a) locality features are key features; (b) learning-based methods can significantly improve the performances of a hybrid streaming system as compared to legacy approaches; (c) improving throughput has indirect important side advantages: it improves delay, chunk losses[4], re-buffering, quality switches.

The paper is structured as follows: In Section II we present the considered neighbor selection policies, the system characteristics, the A/B testing strategy, and the metrics used to measure the effectiveness of the policies, and discusses the features, the model training, and its deployment. Section III presents the aggregate and detailed results of several A/B tests. Finally, we present the related work in Section IV.

II. Methodology

Architecture: In a typical hybrid CDN-V2V live streaming system, there are three major components: (1) a tracker which keeps track of viewers and *selects neighbors* to build an overlay network, (2) a JavaScript library loaded by users' browsers which contacts the tracker to connect to other viewers, and (3) the log database for performance telemetry.

Experiments: To compare different neighbor selection algorithms, we devised an A/B test scheme in which viewers are split into two separate overlay networks in real-time. Given the split's randomness, we can infer the impacts of the algorithms.

Evaluation metrics: We evaluate their impacts on the Quality of service (QoS) and on the Quality of Experience (QoE) metrics. The QoS metrics evaluates how well the V2V overlay network functions including the *V2V throughput* among clients measured from their chunk exchanges, the *end-to-end delay* among clients showing how fast the availability information[5] can be disseminated, the *chunk loss rate* measuring the percent of chunks that arrive after the deadline[6], and the *V2V ratio* measuring the traffic offload from CDN. The QoE metrics evaluate objectively the users' watching experience, including the *re-bufferings count*, the video *chunk size* indicating the bitrate, and the *quality switches count* showing the fluctuation of qualities.

Algorithms: The algorithms to compare include: (1) random, (2) locality-based (3), and ML-based neighbor selections. The locality-based algorithms include: *Geo*, *Geo-ISP*, and *Geo-Distance* neighbor selections. *Geo* prioritizes the peers in the following order: (1) peers from the same city, (2) peers from the same country, (3) peers from the same continent, and (4) peers from another location. *Geo-ISP* picks peers from the same ISP in priority and, then, ranks them as Geo. Last, *Geo-Distance* prioritizes close neighbors in terms of geographical distance.

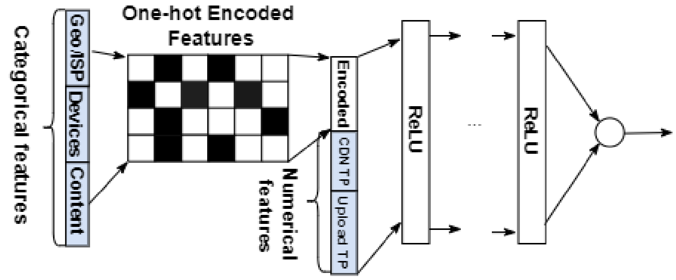

Fig. 1: The neural network structure: the categorical variables are one-hot encoded then concatenated with the throughput features. A classical fully-connected neural network is trained to produce the prediction of potential throughput between two viewers.

The ML-based algorithm tackles the insufficiency of single-criteria policies, as discussed in our previous work [24]. However, in this real-world deployment, we include more features and configure four different feature sets.

1) fs_1 includes geo-location and ISP of both peers[7];
2) fs_2 includes fs_1, the channel, and the devices.
3) fs_3 includes fs_2 and the CDN download throughput.
4) fs_4 encompasses fs_3 and the upload throughput measurements of the seeder.

During the cross-validation process, we determined the best period over which to train our models using historical data: training with less than 14 days of date incurs larger errors while using more data does not significantly improve the model performance. During the experiments presented in the next section, we compared *NN7-fs1* (NN model using feature set *fs1* trained over last 7 days) and *NN14-fs1* to validate our cross-validation result.

III. Experimental results

We carried out 8 A/B test experiment campaigns to compare the performances of the presented neighbor selection strategies. Table I provides an overview of their results. The metrics ending with a 0 (resp. 1) corresponds to Policy 0 (resp. 1). The *Imp* columns correspond to the improvement of Policy 1 over Policy 0.

1) Geo significantly improves over Random neighbor selection: During the 7-day experiment of *Random* versus *Geo* (first row of Table I), the average improvement throughput is 15%. All the other metrics are improved, e.g., the delay *DL* is reduced by 26% and the Chunk Loss Rate (CLR) by 14%. In terms of QoE, we observe 8% fewer quality switches, 12% less re-buffering, and a slightly larger chunk size. *Geo* neighbor selection, which constructs a locality-aware overlay, thus resulting in better V2V performance than a random mesh. The addition of further information like the ISP or an estimate

[4] A chunk loss event corresponds to the case in which a video chunk is not received fast enough from a neighbor viewer, forcing to revert to the CDN server.

[5] A peer notifies its neighbors when it has received a new chunk.

[6] The deadline imposed to avoid waiting for the V2V chunks for too long.

[7] All categorical variables are one-hot encoded

Policy0	Policy1	Throughput (kbit/s) TP0	TP1	ImpTP	Delay (ms) DL0	DL1	ImpDl	Chunk Loss Rate (%) CLR0	CLR1	ImpCLR	Quality Switches (#) Swt0	Swt1	ImpSwt
Random	Geo	2 763.86	3 180.80	1.15	677.58	503.19	0.74	0.31	0.26	0.86	5.06	4.67	0.92
Geo	Geo-ISP	3 206.91	3 216.68	1.00	485.95	458.20	0.94	0.26	0.25	0.98	4.90	4.85	0.99
	Geo-Distance	3 138.06	3 139.19	1.00	499.45	512.25	1.03	0.25	0.24	0.98	5.72	5.60	0.98
	NN7-fs_1	3 143.39	3 228.37	1.03	530.68	520.85	0.98	0.29	0.25	0.88	4.93	4.58	0.93
	NN14-fs_1	3 040.43	3 200.75	1.05	577.95	500.89	0.87	0.26	0.23	0.90	4.55	4.23	0.93
	NN14-fs_2	2 957.57	3 195.89	1.08	591.09	482.97	0.82	0.27	0.22	0.83	4.94	4.55	0.92
	NN14-fs_3	3 005.87	3 453.98	1.15	526.08	402.23	0.77	0.25	0.19	0.74	**3.86**	**3.11**	**0.80**
	NN14-fs_4	**2 993.50**	**3 508.40**	**1.17**	**489.94**	**361.93**	**0.74**	**0.25**	**0.16**	**0.63**	3.63	3.01	0.83

Policy0	Policy1	Rebuffering (#) Rebuf0	Rebuf1	ImpRebuf	V2V Ratio (%) V2VRatio0	V2VRatio1	ImpV2VR	Average Chunk Size (MB) AvgCS0	AvgCS1	ImpCS	# Session
Random	Geo	2.35	2.07	0.88	0.37	0.38	1.05	0.98	0.99	1.01	174 680
Geo	Geo-ISP	2.31	2.23	0.97	0.40	0.39	0.99	0.98	0.99	1.00	132 402
	Geo-Distance	2.49	2.47	1.00	0.40	0.40	1.00	1.05	1.05	1.00	112 356
	NN7-fs_1	2.32	1.97	0.85	0.39	0.38	0.97	0.97	0.98	1.01	101 723
	NN14-fs_1	2.15	1.74	0.81	**0.37**	**0.38**	**1.02**	0.98	0.99	1.01	156 141
	NN14-fs_2	2.63	2.10	0.80	0.39	0.38	0.99	0.97	0.99	1.02	107 958
	NN14-fs_3	2.52	1.88	0.75	0.40	0.38	0.96	**0.97**	**1.02**	**1.05**	126 088
	NN14-fs_4	**2.13**	**1.30**	**0.61**	0.39	0.40	1.01	0.97	1.01	1.04	125 892

TABLE I: A/B Test Result Summary. The table lists the summarised results of eight experiments comparing the *Policy0* (1st column) and *Policy1* (2nd column). We compare seven different metrics between the two rival policies. For instance, *TP0* and *TP1* are the average throughputs using *Policy0* and *Policy1*, respectively. The *ImpTP* is the improvement of the throughput if we use *Policy1* instead of *Policy0*. We keep the same annotation for others metrics. The last column shows the number of sessions in the experiment. The best improvements over the *Geo* policy are marked in bold text.

of the physical distance to the original *Geo* model does not bring significant additional improvement. See Rows 2 and 3.

2) The locality-based methods perform similarly: Addition of further information like the ISP or an estimate of the physical distance to the original *Geo* model, does not bring significantly additional improvement in terms of any metrics (throughput, delay, CLR, etc.), see Rows 2 and 3 of Table I. Also, note that the plain categorical models only using geographical and ISP information (*NN7-*fs_1 and *NN14-*fs_1) have only slightly better performances than the *Geo* model (Rows 4 and 5): throughput is improved by 3 and 5%, respectively, delay by 2 and 13%, CLR by 12 and 10%.

3) Upload and download throughput information are keys for the NN models to improve over Geo: Using a neural network based on the geographical features per se is not enough to further enhance the quality of the distribution network, as the *Geo* over *NN7-*fs_1 experiment highlights (Rows 3 and 4 of Table I). The best model is *NN14-*fs_4 which utilizes the geographical information, ISP, users' operating system, the channel, CDN throughput, and the upload throughput.

4) Improving the throughput ameliorates other metrics, e.g. it reduces the fraction of lost chunks: In a V2V system, the link quality directly relates to the success of chunk delivery. If a chunk takes too long to arrive (timeout), the peer falls back to the CDN server to preserve the QoE. The chunk loss rate (CLR) thus is an important metric to optimize in a hybrid CDN/V2V system. We show here that our simple strategy of optimizing the estimated throughput between two arbitrary peers also lowers the CLR. Indeed, the higher the throughput, the less time it takes to deliver a chunk, and the lower the LR should be.

Figure 2 shows the improvement of throughput versus the one in terms of CLR. One point in the graph corresponds to one day of an A/B test experiment. We see a significant Pearson correlation value (-0.794) between both metrics. We observed this trend for all performed experiments.

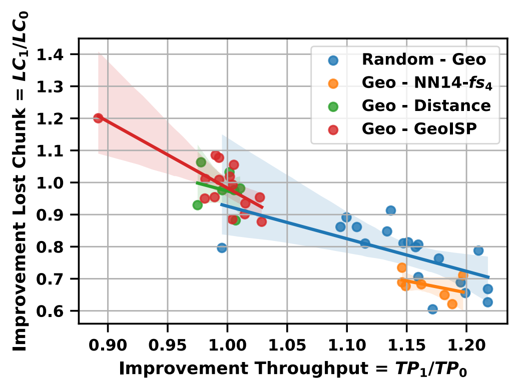

Fig. 2: Correlation analysis of the improvements of the Throughput and the Chunk Loss Rate in the results of 4 A/B test campaigns corresponding to 4 colors. Each dot represents one day of experiment. Each line is the linear regression fitting the dots.

Similar results are obtained when observing the correlation of throughput improvement with other metrics such as the E2E delay or rebuffering. Figures are omitted here due to lack of space.

IV. Related Work

The P2P approach has been widely investigated as a candidate to offer video streaming. The overlay construction is one of the most discussed topics in this context. There exist different overlay classes, including tree-based, mesh-based, and hybrid networks. [25] uses the tree-based structure for lowers latency, but it suffers from peer churn. The mesh-based network is the most resilient overlay w.r.t. load variation. However, it suffers from a higher latency, and the organization of the mesh is nontrivial. Therefore, much research is aimed at improving mesh-based overlays.

One option is to alter the tree-overlay with mesh components or vice versa. A DHT-aided mesh-overlay can enable the newcomers to be assigned such that the chunks are transmitted throughout a dynamic tree structure with top-down decreasing bandwidth [16]. ASTREAM [15] integrates the location/upload capacity information to calculate a reputation score and build a hybrid overlay that uses the location/upload capacity information to form a tree-like overlay.

Another direction is to (re-)create the overlay from scratch because re-organization incurs much less overhead than adaption [26]. When the mesh-based P2P networks were first proposed, a random peer selection of neighbors was the default approach to construct the overlay. More efficient methods have been proposed and proved to improve over a random peer selection. These approaches directly relate to our work.

Locality-awareness. The authors of [17] demonstrate the importance of locality through a large-scale measurement of the P2P-TV system. [18] points out that locality awareness is not enough and introduces the concept of ISP-awareness to minimize the inter-domain / inter-ISP traffic. [19] emphasized the necessity to keep a volume of inter-ISP traffic to achieve a "minimum cloud bandwidth."

Latency-awareness. In [20], the authors propose a method to construct a delay-based overlay consisting of neighborhoods with similar latency. Their simulation results demonstrate improvements in terms of video quality and frame freezing rate over the random mesh. In [21], the peers are classified into classes based on their contribution. The classes are then used to organize the peers, avoiding overloading the peers with the best connectivity.

Serviceability-awareness. Several works [15, 27]–[29] investigated the benefits of building diffusion overlays that consider the serviceability (upload capacity, age, location) of peers and move the better peers closer to the source.

As an extension to our previous work [24] which presents an *offline* data analysis and model testing, this work goes a few steps further by investigating practical issues for deploying a machine learning model in production: (1) feature selection (2) model tuning and, (3) model deployment. Our system uses a hybrid approach that simplifies the diffusion tree. We also consider an enriched set of features such as locality, ISP, device type, past CDN, and upload bandwidth. Several real-world A/B testing experiments demonstrate that an ML approach can improve over heuristic / deterministic neighbor selection strategies focusing only on a limited set of features.

References

[1] "global 2021 forecast highlights - cisco.com." [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf

[2] Akamai cdn. [Online]. Available: https://www.akamai.com

[3] Amazon CloudFront. [Online]. Available: https://aws.amazon.com/cloudfront/

[4] Microsoft azure cdn. [Online]. Available: https://azure.microsoft.com/en-us/services/cdn/

[5] Cloud CDN. [Online]. Available: https://cloud.google.com/cdn

[6] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, and R. Weber, "Pushing CDN-ISP collaboration to the limit," *ACM SIGCOMM Computer Communication Review*, 2013.

[7] C. Huang, A. Wang, J. Li, and K. W. Ross, "Understanding hybrid CDN-p2p: why limelight needs its own red swoosh," in *ACM NOSSDAV*, 2008.

[8] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Design and deployment of a hybrid CDN-p2p system for live video streaming: experiences with LiveSky," in *ACM MM*, 2009.

[9] EasyBroadcast end-to-end OTT streaming for broadcast, enterprise, and education. [Online]. Available: https://easybroadcast.fr/

[10] Jiangchuan Liu, S. Rao, Bo Li, and Hui Zhang, "Opportunities and challenges of peer-to-peer internet video broadcast," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 11–24, 2008.

[11] M. Zhang, L. Zhao, Y. Tang, J.-G. Luo, and S.-Q. Yang, "Large-scale live media streaming over peer-to-peer networks through global internet," in *ACM P2PMMS*, 2005.

[12] S. Xie, B. Li, G. Y. Keung, and X. Zhang, "Coolstreaming: Design, theory, and practice," *IEEE Transactions on Multimedia*, 2007.

[13] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *Springer International Workshop on Peer-to-Peer Systems*, 2005.

[14] N. Magharei and R. Rejaie, "Prime: Peer-to-peer receiver-driven mesh-based streaming," *IEEE/ACM Transactions on Networking*, 2009.

[15] B. U. Maheswari and T. K. Ramesh, "An improved delay-resistant and reliable hybrid overlay for peer-to-peer video streaming in wired and wireless networks," *IEEE Access*, 2018.

[16] H. Shen, Z. Li, and J. Li, "A DHT-aided chunk-driven overlay for scalable and efficient peer-to-peer live streaming," *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[17] T. Silverston, L. Jakab, A. Cabellos-Aparicio, O. Fourmaux, K. Salamatian, and K. Cho, "Large-scale measurement experiments of p2p-TV systems. insights on fairness and locality," *Elsevier Signal Processing: Image Communication*, 2011.

[18] X. Zhang, N. Wang, and M. Howarth, "A hybrid peer selection scheme for enhanced network and application performances," in *IEEE CCNC*, 2013.

[19] J. Zhao, C. Wu, and X. Lin, "Locality-aware streaming in hybrid p2p-cloud CDN systems," *Peer-to-Peer Networking and Applications*, 2015.

[20] J. Chakareski and P. Frossard, "Delay-based overlay construction in p2p video broadcast," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1985–1988.

[21] E. C. Miguel, C. M. Silva, F. C. Coelho, I. F. S. Cunha, and S. V. A. Campos, "Construction and maintenance of p2p overlays for live streaming," *Multimedia Tools and Applications*, 2021.

[22] C. Wu, B. Li, and S. Zhao, "Exploring large-scale peer-to-peer live streaming topologies," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, no. 3, pp. 1–23, 2008.

[23] B. Li and D. Niu, "Random network coding in peer-to-peer networks: From theory to practice," *Proceedings of the IEEE*, 2011.

[24] Z. Ma, S. Roubia, F. Giroire, and G. Urvoy-Keller, "When locality is not enough: Boosting peer selection of hybrid cdn-p2p live streaming systems using machine learning," in *Network Traffic Measurement and Analysis Conference (IFIP TMA 2021)*, 2021.

[25] K. Pal, M. Govil, and M. Ahmed, "A new hybrid approach for overlay construction in p2p live streaming," in *ICACCI*, 2015.

[26] R. Beverly and M. Afergan, "Machine learning for efficient neighbor selection in unstructured p2p networks," *SysML*, 2007.

[27] R. J. Lobb, A. P. Couto da Silva, E. Leonardi, M. Mellia, and M. Meo, "Adaptive overlay topology for mesh-based p2p-TV systems," in *ACM NOSSDAV*, 2009.

[28] B. Biskupski, M. Schiely, P. Felber, and R. Meier, "Tree-based analysis of mesh overlays for peer-to-peer streaming," in *IFIP DAS*, 2008.

[29] A. G. Sheshjavani, B. Akbari, and H. R. Ghaeini, "A free-riding resiliency incentive mechanism for VoD streaming over hybrid CDN-p2p networks," in *IEEE IST*, 2016.