Björn Richerzhagen, Julian Wulfheide, Heinz Koeppl, Andreas Mauthe, Klara Nahrstedt and Ralf Steinmetz. *Enabling Crowdsourced Live Event Coverage with Adaptive Collaborative Upload Strategies*. In Proc. 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2016. ISBN: 978-1-5090-2185-7

Enabling Crowdsourced Live Event Coverage with Adaptive Collaborative Upload Strategies

Björn Richerzhagen*, Julian Wulfheide*, Heinz Koeppl†, Andreas Mauthe‡, Klara Nahrstedt§ and Ralf Steinmetz*
*Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
E-Mail: {bjoern.richerzhagen|julian.wulfheide|ralf.steinmetz}@kom.tu-darmstadt.de
†Bioinspired Communication Systems Lab (BCS), Technische Universität Darmstadt, Germany
‡School of Computing and Communications, Lancaster University, UK
§Department of Computer Science, University of Illinois, IL, USA

***Abstract*—User-generated content, such as short video snippets or tweets, is increasingly used in event coverage even by professional media outlets. Especially in unforeseen events, or when dealing with large crowds, these snippets provide unique perspectives on the scene. While uploading a tweet does not impose much load on the communication system, uploading live video at today's camera resolutions consumes a significant amount of resources. At the same time, only a fraction of the uploaded streams is suitable for event coverage (e.g., shakiness of the video, focus on the scene, obstructions). By identifying the set of relevant streams early, and postponing the upload of other content, the available network resources can be dedicated to the upload of the most relevant streams. In this paper, we propose a set of strategies to collaboratively upload the most relevant streams at high quality by utilizing freed resources. We argue that these strategies can be exchanged during runtime to adapt to user dynamics and network heterogeneity, and present initial findings on the performance of our system.**

I. Introduction

Utilizing user-generated content is often the only way to report on unforeseen events or to provide insights into large crowds. The content itself ranges from short text snippets (e.g., *tweets* or facebook messages) to images, and, more recently, live videos. A prominent example for a platform that is specifically targeted towards sharing user-generated live streams is Periscope, a subsidiary of Twitter, asking *What if you could see through the eyes of a protester in Ukraine?*[1] to motivate its service. However, due to the bandwidth restrictions of today's mobile access networks and the utilized devices and tariffs, content that can be used in a live or near-to-live fashion is usually limited to low resolution videos. This becomes even more apparent when an event, such as the aforementioned protest, is being filmed by a large number of users. Each individual user occupies a certain share of the available resources to upload the respective video stream.

However, many of the generated video streams are redundant in their content and only vary concerning their quality w.r.t. camera resolution, movement, shakiness, or how well they capture the relevant aspects of the event. By identifying the most relevant streams early (ideally directly on the users' devices), one can postpone the upload of less relevant streams to free network resources. These freed resources can then be used to improve the upload performance for the most relevant streams. By incorporating other devices in a collaborative fashion, heterogeneities in the upload capabilities of individual devices can be utilized. If, for example, a device in close proximity to the video source has access to the Internet via a broadband Wi-Fi Hot Spot, this device can be utilized as a relay. The content encoding scheme used by the source further determines the utility of a specific upload strategy. H.264/SVC encoded content, for example, enables dedicated treatment for each quality layer of the video, whereas Multiple Description Coding (MDC) supports probabilistic schemes without strict delivery guarantees. Depending on the content of the stream or the type of the event, the desired properties of an upload scheme can vary as well. In the case of a live sports event, achieving low overall delay is crucial if the stream is desired to compete with traditional broadcast media or live tickers. In other situations, achieving higher bandwidth at the expense of an increased delay might be more desirable.

In this paper, we analyze the scenario of crowdsourced live event coverage and motivate the need for collaborative media upload (Sec. II). We propose a set of collaborative upload strategies, enabling media transmission to a server at higher bandwidth compared to the isolated approach. Our initial findings suggest that upload strategies need to be exchangeable to adapt to user dynamics and network heterogeneity (Sec. III). Finally, we discuss the potential of a feedback loop between the source selection mechanism and the collaborative uploading strategy to further increase the overall system performance and provide some insights into our ongoing works (Sec. IV).

II. Crowdsourced Live Event Coverage

Figure 1 illustrates the concept of crowdsourced live video streaming. Several mobile devices are utilized to film a certain event, whereby each device streams its recorded video via a distribution network to a number of interested clients. Examples of such distribution networks include sites like Twitch, Ustream, or the recently announced Periscope. The quality of the video is adjusted to the available resources – uploading via a lower bandwidth cellular connection leads to lower video quality. As some of the platforms offer an archival feature, media streams are often uploaded even if no client is interested in watching the stream at that point in time.

For the collection and processing of crowdsourced video data, systems like [5] utilize resources in close proximity to the

[1] www.periscope.tv/about [accessed Dec. 2, 2015]

978-1-5090-2185-7/16/$31.00 ©2016 IEEE

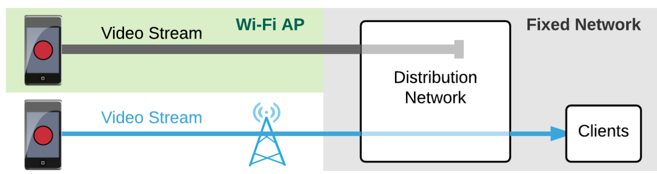


Fig. 1. Parallel upload of multiple live media streams. Each stream consumes resources, even if it not being consumed by any client. The stream quality depends on the available bandwidth.

users, e.g., cloudlets based on virtual machines. However, the resource consumption between the video source and the first server, or cloudlet, is not considered. To save resources and to upload only relevant streams, systems that are capable of identifying and selecting relevant sources have been proposed. Often, the goal of such systems is to compose a single stream of high aesthetic quality out of a number of user-generated input streams. Therefore, the role of a *director* is introduced. This entity chooses the active input stream out of a range of sources either automatically (e.g., based on rules or a pre-defined script), or through human interaction. Still, systems like [1] require all streams to be available to the director, which results in a waste of resources, as streams need to be uploaded, even if they are not part of the composed video lateron. To address this issue, recent proposals include an early identification of relevant streams based on lightweight meta data (such as sensor readings or video quality metrics) [6], [8]. The resulting communication pattern is illustrated in Figure 2. Here, only the active source uploads its video stream to the distribution network, while the director operates on monitored meta data to perform source selection. The currently unselected video sources record the video only to their local memory, uploading it at a later point in time for archival (e.g., at home while charging).

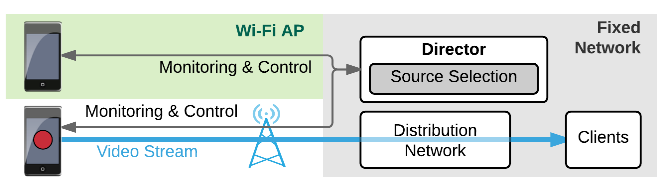


Fig. 2. Source selection based on monitored video data. Only the most relevant streams are uploaded to the distribution network. Still, resource utilization can be sub-optimal.

Still, for the video upload itself, only the resources of the selected source can be utilized. In Figure 2, for example, the active source uploads its stream via a cellular network, while the Wi-Fi connection that is available to a nearby inactive source is not utilized. In the following, we propose collaborative upload strategies to further incorporate idle resources in the overall streaming process. Similar to the concept of cellular offloading [2], [9], our proposed system coordinates mobile users in close proximity to enable resource sharing. However, in contrast to related works, we utilize monitored data that is already required for the source selection process and benefit from a centralized view on the relevant part of the system (i.e., the set of active devices in proximity to the event that is to be covered). The goal of the proposed strategies is an efficient utilization of the available network resources to improve the streaming quality in heterogeneous and dynamic scenarios.

III. Collaborative Upload Strategies

A simple collaborative upload strategy is illustrated in Figure 3. The active source no longer uploads the stream at a low bitrate via its own cellular connection (in the following referred to as *direct* upload), but instead *relays* it to a nearby device, using technologies such as Wi-Fi Direct. The relaying device now utilizes its connection to a Wi-Fi Access Point to upload the stream at a higher bitrate to the distribution network. In addition to direct upload and the relay via ad hoc connectivity, we also introduce a hybrid approach, where the relay node is used to augment the cellular connection. In this *hybrid* strategy, content is split into two substreams, whereby one of the streams is relayed via the access point, and the other one is directly uploaded by the source node. The respective uploading strategy is coordinated by the director, that operates on monitoring data that is previously collected during the source selection process (c.f. Fig. 2).

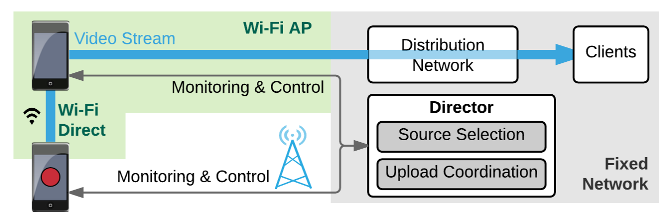
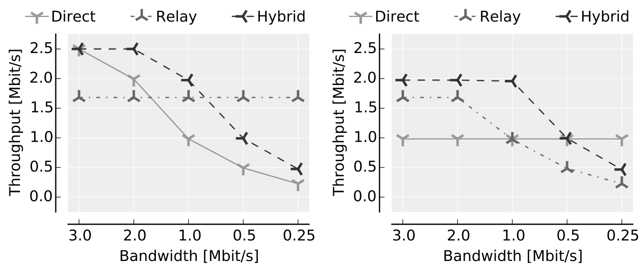


Fig. 3. Source selection and coordination of a collaborative upload. In this case, a device connected to a Wi-Fi Access Point is utilized to upload the stream of the selected source having only limited cellular bandwidth.

Within the scenario of mobile crowdsourced event coverage, the source of the media stream is dynamically determined by a director. The capabilities of the selected source, along with its current surroundings, pinpoint the most appropriate strategy for media upload. To motivate the need for different upload strategies, as well as transitions between those strategies, we conducted a proof-of-concept evaluation on the Simonstrator platform [3]. We simulated a single source trying to upload a video. The source has cellular connectivity (1 Mbit/s upload) and is in ad hoc communication range to one potential relay node. The relay node is connected to a Wi-Fi access point with a bandwidth of 2 Mbit/s and available for the whole duration of the simulation. The source and relay node communicate with each other via Wi-Fi ad hoc, using the NS-3 [4] reference model for 802.11g. We vary the cellular bandwidth of the source node as well as the bandwidth available to the relay node via the Wi-Fi access point. In both cases, the observed performance varies depending on the selected upload strategy.

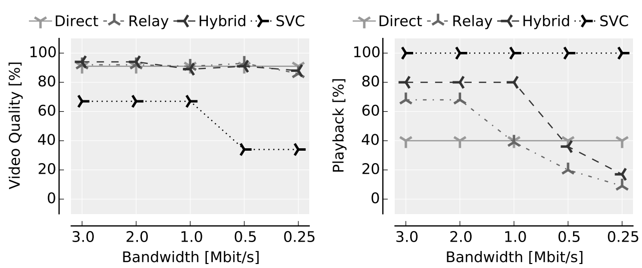
Figure 4 shows the achieved throughput for different configurations of the cellular bandwidth (Fig. 4a) and of the relay's access point bandwidth (Fig. 4b). The video has a bitrate of 2.5 Mbit/s and, thus, cannot be uploaded directly be the source with sufficiently high bitrate. For direct upload, the performance is solely limited by the cellular upload bandwidth. When uploading the stream solely via the relay node, the achievable throughput is confined to around 1.7 Mbit/s, which results from the direct ad hoc transmission. This saturation becomes even more apparent when varying the access point



(a) Varying Cell bandwidth (b) Varying AP bandwidth

Fig. 4. Achieved throughput of upload strategies for a fixed video bitrate of 2.5 Mbit/s and varying access point and cellular bandwidth.

bandwidth (Fig. 4b). The effect can be counteracted to some extent by utilizing both the cellular upload, and the upload through the access point via a relay node. In both scenarios, switching between different strategies depending on the available resources leads to higher achieved throughput.



(a) Playback quality using SVC (b) Continuity using SVC

Fig. 5. Impact of the chosen video codec on the playback performance resulting from different upload strategies under varying access point bandwidth.

However, until now, the focus of our discussion was single layer video and strategies that are agnostic of the utilized video codec. Therefore, we evaluated the individual strategies with streams that we encoded using the layered H264/SVC codec. The overall bitrate of the streamed video remained set to 2.5 Mbit/s. We encoded the video with a 500 kbit/s base layer and two enhancement layers with 1 Mbit/s each. Receiving the base layer is sufficient for playback at low quality, and each enhancement layer increases the video quality, but the base layer is still required for playback. The results are depicted in Figure 5. Note, that the achieved throughput remains the same for the aforementioned strategies (c.f. Fig.4). This also determines the playback continuity achieved by the strategies (Fig. 5b). For SVC, we introduce a codec-aware upload strategy that prioritizes packets containing the base layer over packets containing enhancement layers. If there is sufficient bandwidth available via relay nodes, the corresponding number of enhancement layers is sent to the relay. Otherwise, only the base layer is uploaded.

This simple codec-aware strategy maintains continuous playback (Fig. 5b) at the cost of reduced video quality for the viewer (Fig. 5a). These initial finings motivate transitions between different such upload schemes depending on the selected streaming source, as well as the current environmental conditions. While the aforementioned results were obtained in a rather static setting, a real world deployment of the proposed system would operate under more dynamic conditions. User movement, as well as varying transmission quality, requires additional, more robust upload strategies. Additionally, incentives for relaying users would need to be provided, as these users contribute their upload capabilities and energy. In our envisioned setting, such incentives could be provided by the platform owner by turning the high quality coverage into a premium feature. When postponing the upload of a video and contributing idle resources for the sake of providing an overall composed stream at higher quality, a user could be awarded with free access to the premium service. Similar concepts have already proven to be beneficial for peer-to-peer live video streaming systems [7].

IV. Conclusion

In this paper, we motivated the need for adaptive collaborative upload strategies to support the composition of a high quality stream in crowdsourced live event coverage. Therefore, we propose to extend the role of the director from exclusively compositing, to a more active orchestration of the involved devices. Instead of solely selecting sources, the director aids in the delivery process by dynamically selecting the currently appropriate upload strategy from a set of predefined strategies, utilizing nonetheless gathered monitoring data.

While the set of strategies presented in this paper is rather simple, they already motivate the need for transitions between strategies in dynamic scenarios with changing conditions, or based on the video codec utilized by the active source. However, we expect a clear tendency towards simple strategies especially in highly dynamic settings. Next steps besides researching on more sophisticated strategies is the combined evaluation of a feedback loop between source selection and upload coordination to better cope with highly dynamic scenarios.

References

[1] X. Bao and R. R. Choudhury, “VUPoints: Collaborative Sensing and Video Recording through Mobile Phones,” *Computer Communication Review*, vol. 40, no. 1, 2010.

[2] X. Bao, Y. Lin, U. Lee, I. Rimac, and R. Choudhury, “Dataspotting: Exploiting naturally clustered mobile devices to offload cellular traffic,” in *Proc. IEEE INFOCOM*, April 2013, pp. 420–424.

[3] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, “Simonstrator: Simulation and prototyping platform for distributed mobile applications,” in *Proc. SIMUTOOLS*. ICST, 2015.

[4] G. Riley and T. Henderson, “The NS-3 Network Simulator,” in *Modeling and Tools for Network Simulation*. Springer, 2010.

[5] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, “Scalable crowd-sourcing of video from mobile devices,” in *Proc. MobiSys*. ACM, 2013.

[6] D. Stohr, S. Wilk, and W. Effelsberg, “Monitoring of user generated video broadcasting services,” in *Proc. Workshop on Internet-Scale Multimedia Management (WISMM)*. ACM, 2014.

[7] M. Wichtlhuber, P. Heise, B. Scheurich, J. Rückert, and D. Hausheer, “vINCENT: An incentive scheme supporting heterogeneity in Peer-to-Peer content distribution,” in *Proc. IEEE LCN*, 2014.

[8] S. Wilk and W. Effelsberg, “Mobile video broadcasting services: Combining video composition and network efficient transmission,” in *Proc. Conference on Multimedia (MM)*. ACM, 2014.

[9] F. Zhang, C. Xu, Y. Zhang, K. Ramakrishnan, S. Mukherjee, R. Yates, and T. Nguyen, “Edgebuffer: Caching and prefetching content at the edge in the mobilityfirst future internet architecture,” in *Proc. WoWMoM*. IEEE, 2015.