Drone for live video multicasting: Understanding how reliability parameters affect playback delay

Rasmus L. Bruun*, Martin Eriksen†, Rasmus S. Mogensen‡ and Kasper W. Mortensen§
Department of Electronic Systems, Networks and Distributed Systems
Aalborg University
Email: *rbruun13@student.aau.dk, †merik13@student.aau.dk, ‡rmogen13@student.aau.dk, §kwmo13@student.aau.dk

***Abstract*—Most video codecs do not tolerate data loss. Therfore, a reliable multicast scheme is preferable for live video streaming from a drone to multiple receivers. We examine how adaptation of video, transmission and erasure coding rate can improve reliability of multicast. Based on the examination we design a joint adaptation policy, for continuous and delay bounded video playback. First, we analyse the characteristics of the wireless channel when using a drone through a measurement campaign. This measurement campaign resulted in traces for different transmission rates specified in 802.11g as well as video traces from a Bebop 2 drone. Second, we leveraged these traces to understand the conditions for our joint design looking not only at the achievable video rate, but also at the inherent delay introduced on a per frame basis. Our results show that transmission rate adaptation does not improve the reliability in the examined test scenario. Changing the amount of redundancy added by erasure coding combined with adaptation of the video rate increase the reliability. We propose two adaptation policies that minimize video playback freezes while maximizing the video quality for both playback methods.**

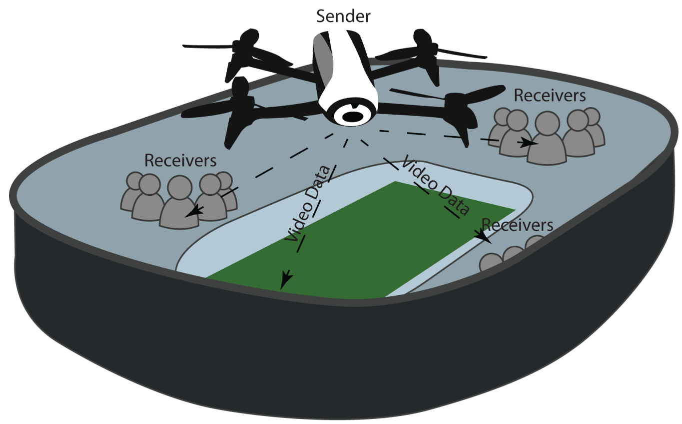

Fig. 1: A drone multicasts a live video stream to spectators at a football stadium.

I. Introduction

The popularity of drones has increased in recent years due to their applicability in a broad spectrum of scenarios in different fields. Particularly, live video streaming has shown to be of interest in search and rescue, law enforcement, military operations, industrial usage and entertainment [1], as shown in Figure 1. In several of these applications, there are strict delay requirements regarding the video stream, e.g. in military operations delayed information of the enemies position can have severe consequences. A reliable video stream to multiple receivers can be achieved with unicast if the group size and the amount of video data does not exceed the capacity of the wireless channel. To improve reliability during unicast the transmission rate is adapted based on feedback from the receiver gaining different properties in terms of noise sensitivity and throughput. To account for packet erasures a reliable transmission protocol can be used.

If the capacity of the wireless channel is exceeded due to an increased amount of receivers, multicast can be used to service the group. However, multicast is not without challenges. Current standardized protocol stacks do not implement feedback mechanisms for multicast. Therefore, the MAC layer is unable to adapt to changes in channel conditions. The lack of feedback is also problematic at the transport layer, where only connectionless protocols support multicast, which means that packet erasures are unhandled. The specific multicast technology examined in this paper is IEEE 802.11 [2].

For most video codecs, data loss is not acceptable as losses degrade quality and in the worst case, the video becomes unplayable. Therefore, plain 802.11 multicast which lacks transmission rate adaptation and packet erasure handling is not suitable. In [3], [4] the authors suggest using an erasure correcting code to handle packet erasures. They adjust the redundancy dynamically based on the packet error rate (PER).

Transmission rate (TxR) adaptation schemes for multicast has been proposed in multiple papers [5]–[7]. Adaptation is desirable when the channel conditions change, e.g. when distance between the sender and receivers change. The authors of [5] use signal to noise ratio to adapt the TxR, whereas [6] suggest a PER based adaptation. Adaptation of TxR and erasure code rely on feedback. In [4], [7] the authors provide solutions to gather feedback from multiple receivers. The authors of [4] suggest collecting feedback from every receiver, while [7] select a few receivers which send feedback periodically.

Redundancy from the erasure code takes up usable resources from the TxR. Additionally, changing the TxR imposes limitations on how much video data can be serviced. This means that the video rate should be adjusted based on these limitations. Some applications require harder deadlines for video arrival and other applications require good video quality. Therefore, these requirements are additional parameters that should be considered when adjusting the video rate.

This paper examines reliable multicast for live video streaming from a drone. Inspired by [4], we form a policy that based on PER adapts transmission rate, erasure coding and video rate. We base our examination on the erasure coding random linear network coding (RLNC) [8] and the video encoding scheme H.264 [9]. In addition we account for video playback delay when considering two different playback methods, which to the best of our knowledge has not been considered in prior art. In particular, we customise the policies to work for drones by basing our investigation on channel throughput and video traces taken from a commercial drone, which has not been attempted previously.

The remainder of the paper is organized as follows. Section II contains a description of the system and the scenario in which it operates. Section III gives a brief presentation of implementation details on the three adaptable system components. Section IV describes the methods used for trace collection and system analysis. Section V contains the results of the measurements performed. Section VI contains the resulting policy and the reasoning behind it. Section VII summarizes the policy, gives concluding remarks and future work.

II. System setup and scenario

The system examined in this paper is illustrated in Figure 2. It consists of a drone-based sender that is able to adjust three mechanisms to obtain reliability: the TxR, the amount of redundancy introduced by RLNC and the video rate. We base the adaptation of the three mechanisms on the PER. This limits our policy to a single feedback measure, which has three advantages (i) it results in a simple policy, (ii) it does not require specialized equipment at the receivers and (iii) it is easy to compare the conditions at multiple receivers. The adjustment of the mechanisms, depicted in Figure 2, is the following: 1) the sender receives information about the PER, which is made available to both the TxR and the RLNC component; 2) based on the PER information, the transmitter follows a decision policy to change the redundancy and TxR; 3) the video rate component is notified of the remaining data available and adjusted accordingly; 4) video data is delivered to the RLNC component and redundancy is added; 5) the coded packets are transmitted with the rate determined by the TxR component.

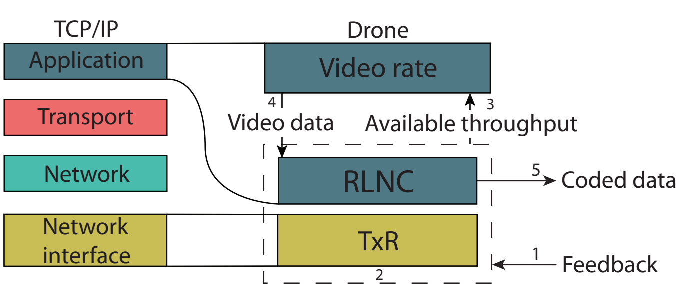

Fig. 2: The adjustment flow of reliable live video streaming and how the components relates to the TCP/IP stack.

The scenario examined is an open field where a drone captures live video and multicasts it to a number of receivers. The drone moves in an unpredictable pattern, but within a confined space of a circle with radius 100 m, where the receivers are placed in the center. For safety reasons the minimum hover height is 2 m. All receivers are located close together and within line of sight of the sender. Furthermore, it is assumed that all receivers have similar hardware and software configurations.

III. Implementation details

This section presents a brief description of the three components of the system and how their implementation effect the system's behaviour and video playback delay.

A. Transmission rate

While the TxR sets the actual speed at which data is sent, it is not a direct measure of the amount of data, which can be transferred from the application layer over an extended period of time. We use throughput to denote the maintainable TxR. 802.11 utilize both packet headers and tails, which lower the throughput. Parts of the header is always transmitted at the lowest TxR for compatibility reasons. The largest contributor to why the throughput is lower than a selected data rate is the protocol behaviour. The protocol behaviour renders the channel idle for certain periods of time [10], [11]. The usable throughput of 802.11 is the amount of payload it is able to service. This can be measured as the number of MAC layer service data units (SDU) transmitted over a period of time. An SDU is the payload a layer provides to the above layer. The MAC SDU can be seen on Figure 3.

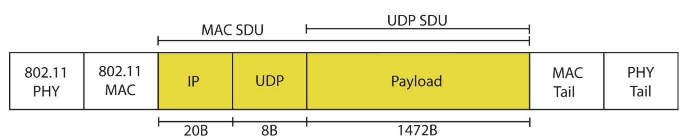

Fig. 3: 802.11 frame with network interface, network and transport layer headers marked

The protocol overhead also imposes limitations since the MAC SDU is limited to 1500 bytes (B) and IP & UDP also require additional overhead. The IP header takes 20 B and has no protocol behaviour which impose further overhead. UDP takes an additional 8 B, leaving 1472 B of the MAC SDU throughput available to user applications.

Because 802.11 is a wireless communication technology the physical placement of the sender and receiver effect the performance. As the signal propagates the strength decreases with distance and reflections/interference can corrupt packets in transit. The different TxR described in 802.11 have different signal modulation which means the probability of packet corruption differs. By lowering the TxR, packets are less sensitive to noise and interference, therefore it is a trade-off between transmission speed and sensitivity.

B. Random Linear Network Coding

RLNC works by creating coded packets that are random linear combinations of a set of original packets called a generation. A generation consists of n packets and can be used to generate an arbitrary amount of coded packets of size $n + k$, see Figure 4. k is the number of redundant packet and can be varied such that it matches the packet loss in the channel. It is possible to obtain the original data if at least n linearly independent packets are received.

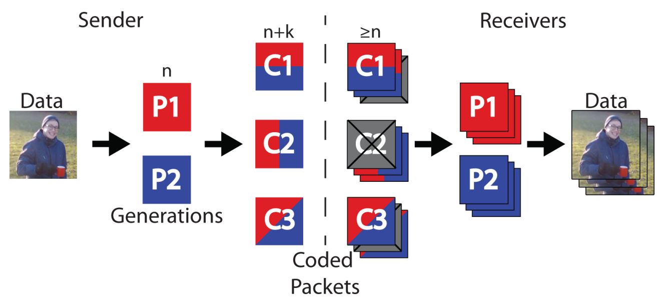

Fig. 4: Usage of RLNC is showed, from coding (left) to decoding (right)

RLNC is flexible since no prior information is needed at the receivers as each coded packet is self contained. This makes information sharing between the receivers easier. If a receiver has less than n packets, it can reobtain its missing packet from another receiver. Therefore RLNC becomes advantageous compared to retransmissions when packet losses are uncorrelated, which is the case for our system [12], [13]. These features come at a cost of an extra data and computational overhead when encoding and decoding which introduces additional system delay. Additionally, the randomness of the coding process can result in linear dependency between coded packets, which can result in situation where n packets are received but decoding is impossible.

C. Video rate

The video rate is the last resource we want to change to gain a reliable live video stream. The video quality should be as high as possible at all times given the application requirements. To change the video rate appropriately, an understanding of how the different parameters in H.264 work and how they affect the nature of the data stream [14]. H.264 consist of I, P and B frames, where only the I and P frame are used for streaming. The I-frame consists of the whole image and is needed when receivers join the stream. A P-frame is the difference in the picture from a previous P-frame or an I-frame. For controlling the bit rate we use variable bitrate (VBR), which allocates as much data needed to every frame. In order to restrain the data size, a target bitrate is specified, which the encoder aims to achieve on average [14]. The bitrate is highly dependent on the change in the video, which can be unpredictable. Because only the necessary data is allocated for each frame the bandwidth is used efficiently.

IV. Mesurement methods

This section contains the descriptions and methods of the tests conducted, in order to understand how and when the different reliability mechanisms should be adapted. To simplify our analysis we make the assumption that the sender has perfect feedback and is able to switch TxR, RLNC and video rate instantaneously. At the receiver we assume no delay is caused by feedback, erasure coding or the video player, and we assume that all receivers receive the same throughput from the sender. Additionally, we assume that RLNC introduces no data overhead or linear dependency. Finally, the throughput examination assumes that every receiver's link to the drone is identical. Based on these assumptions only the channel condition has an effect on the system delay and we model the system as two queues, see Figure 5.

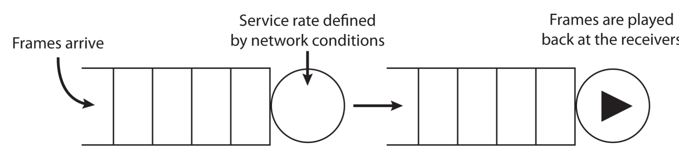

Fig. 5: Simplified system model

When a frame is captured by the drone at a given video rate it is prepared for transmission with additional redundancy. Packets are placed in a buffer and serviced sequentially based on the capability of the transmitting network. When a packet is serviced it is transferred instantaneously to the receivers playback-buffer and ready to be serviced by the receivers video player.

All the tests are performed using the same test equipment. A receiver consist of one Raspberry Pi 1 model B and a TP-link TL-WN722N USB WiFi dongle. The sender is a Raspberry Pi 3 model B which uses the WiFi dongle RT5370. Both the sender and receivers are running Raspbian linux. The sender is configured as an 802.11g access point [2]. The drone used to carry the sender is a Parrot Bebop 2 drone.

Unless otherwise stated the empirical tests are performed with one sender and 4 receivers, where the receivers are placed right next to each other. Multiple receivers are used to account for hardware diversity.

A. TxR

To understand how the different TxR perform in practice when comparing to the physical location of the drone, we measure the percentage of UDP packets reaching a receiver as a function of the distance and height between sender and receiver. We conduct a number of tests, varying one parameter at a time. Each test has a duration of 1 minute. The different bit rates tested were 12, 18, 24, 36 megabit per second (Mb/s) at distances between 10 and 100 m, in 10 m intervals at a height of 45 cm and 2 m. For each TxR we measure the UDP SDU throughput at the sender in order to determine the actual data rate available for the video data.

B. Video rates

In order to understand the behaviour of different video rates, we analyse videos obtained from the drone using VBR [14]. The target bitrates are set to 5, 10, 20 Mb/s and a large open area is recorded. To perform the analysis of the video, petro [15] (a video analysis tool) has been used.

C. Aerial data collection

For the main source of analysis we use the drone in the scenario previously described in Section II to obtain a trace which shows the packet loss at each TxR simultaneously [16]. The sender can only transmit at one rate at a time. Therefore, the following technique is used to estimate the continues channel behaviour for each TxR: the sender alternates between the four rates 12, 18, 24, 36 Mb/s, sending 20 packets at each rate before switching. Since TxR is changed from user space, we include a delay to ensure packets are send with the correct TxR. The time to send one packet (denoted t_s) depends on the TxR. The slowest rate is 12 Mb/s and assuming a packet size of 1500 B, each packet takes approximately 1 ms to send, as seen in equation 1.

$$t_s \approx \frac{8\frac{b}{B} \cdot 1500\ B}{12\ Mb/s} = \frac{12\ kb}{12\ Mb/s} = 1\ ms \tag{1}$$

This delay is used for all rates to simplify data analysis. Empirical experiments have shown that adding 5 ms to $20 \cdot t_s$ gives a sufficiently large certainty, that all the packets have been sent. Therefore, the sender transmit 20 packets, wait 25 ms, adjust TxR and repeat. This means that out of 100 ms, at most $1/4$ of the time is used to transmit at one rate. We use the drop rate of the 20 packets to estimate the channel behaviour in the rest of the time interval. We show in [17] that this estimate is in fact valid and gives a good representation of the true application layer throughput.

V. Results

We see that the packet loss at a specific TxR does not depend on distance between the sender and receiver given the bounderies of the scenario [18]. Table I shows the UDP SDU throughtput, as measured in [10]. The measured TxR of Table I are not only subject to protocol design limitations, but might also be affected by other traffic on the link e.g. beacons from the access point and interference from other 802.11 devices all of which decreases the application throughput.

TxR	UDP SDU throughput
12 Mb/s	10.296 Mb/s
18 Mb/s	14.958 Mb/s
24 Mb/s	19.305 Mb/s
36 Mb/s	27.174 Mb/s

TABLE I: TxR specified in 802.11g and the equivalent throughput measured at the application layer

As expected the application data rate is lower than the MxR and it is something that should be considered not only when adjusting the reliability parameters but also when examining the system delay.

Segments of three video traces captured by the drone at 5, 10, 20 Mb/s can be seen in Figure 6. Each dot indicate the arrival of a new frame and the height indicate the bitrate needed to service the frame before a new frame arrives. As seen in the Figure 6, the VBR encoding mechanism allocates much of the available bandwidth for the I-frame which is depicted by the spike. The I-frame is a full frame and not incremental differences as the next frames and it has a period of 1 second at 30 fps.

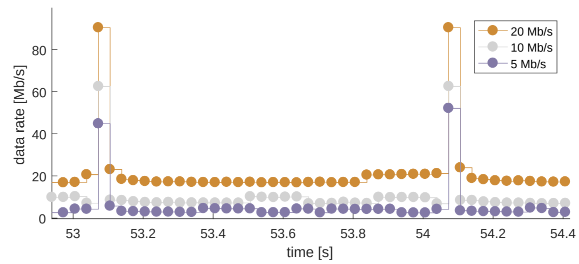

Fig. 6: Traces of different three H.264 encoded VBR videos at different target bitrates

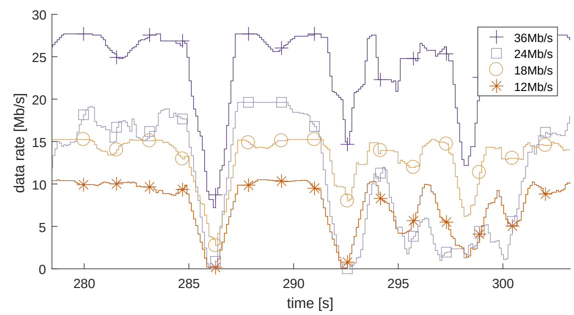

Fig. 7: Available throughput obtained in a trace of different TxR

The next result, seen in Figure 7, are traces showing the estimated behaviour of the different TxR. It shows that the rates are correlated, which means that the errors we observe are not rate specific. The reasons for the decreasing throughput, could be caused by a combination of the following reasons: (i) the sender gets out of range of the receiver, (ii) background noise, (iii) nearby transmission equipment is causing noise or (iv) the drones movement, e.g. acceleration distorts the transmitted signal. Of the mentioned reasons we suspect a combination of reason (i), (ii) and (iv) to be causing the majority of the throughput decrease we see in the traces in Figure 7. Furthermore, it is seen that the highest TxR still offers the highest throughput.

VI. Discussion

It is seen in Figure 7 that the four TxR experience a similar PER over time. Since the PERs are correlated the 36 Mb/s TxR offers the highest throughput and the best adaptation policy is to select the highest TxR. We remind the reader that this policy is suitable for the specific scenario described in Section II. In scenarios with other environments and longer distance between sender and receiver this might not be the optimal policy. This limits our adjustment policy to RLNC and video rate.

RLNC is used to add redundancy to the transmitted video. This allows recovery of the transmitted video at the receiver if the amount of redundancy is equal to or greater than the PER. Since we assume instant feedback, the optimal RLNC policy is to add redundancy equivalent to the PER. The green curve in Figure 8(a) illustrates the available throughput to the video rate of a 8.8 minutes trace. The throughput is based on the estimation presented in Figure 7. Because the trace can fluctuate aggressively, the trace is overlaid by a red line, which is the average throughput over the last second. From this red line, we see that the available throughput rarely drops to 0 Mb/s. From Figure 8(a) and Figure 6 we calculate the time it takes to send a frame from when it was captured till it has reached the receive buffer. This is calculated for each individual frame and can be seen as the blue bars in Figure 8(b,c,d), we denote this as the per frame delay. This delay introduces the playback delay which is the time between a frame is taken until it is displayed at the receivers. A playback freeze occurs when the next frame is delayed beyond the current playback delay. Depending on the playback method these metrics have different impacts.

Continuous playback is playing every frame of the video in chronological order. Whenever a frame is delayed more than the previous frame the playback delay is increased and a playback freeze occurs. The red dotted lines in Figure 8(b,c,d,e) illustrate the playback delay. These freezes can be seen as small vertical boxes on the far right in Figure 8(b,c,d,e). The strength is that every frame of the video is present, the weakness is that there is no upper bound on either the playback delay, or the buffer size. This method accumulates the delay whenever a freeze occurs, which means that the receivers will eventually be heavily unsynchronized with the sender.

Delay bounded playback sets a strict delay requirement and this is the exact amount of time before the receiver will play the video, this is illustrated as the black dotted line in Figure 8(b,c,d,f). Frames delayed beyond this threshold are discarded thereby guaranteeing strict delay and buffer sizes on the sender and receivers. Playback freezes occur every time a frame/series of frames are delayed beyond the threshold. The strength of this method is that the video has a constant playback delay, but the down side is freezes occur in the video whenever frames are not delivered fast enough. The delay bounded freezes are marked as small horizontal boxes under the x-axis of Figure 8(b,c,d,f).

Common for both playback methods is that they will experience playback freezes if the delay of certain frames is large. In Table II are the number of freeze periods and combined freeze time for each video rate. Note that the delay bounded playback is set to 500 ms in our analysis. If the delay bound is set to 1 s less freezes occurs.

For the 20 Mb/s video we see that if continues playback is

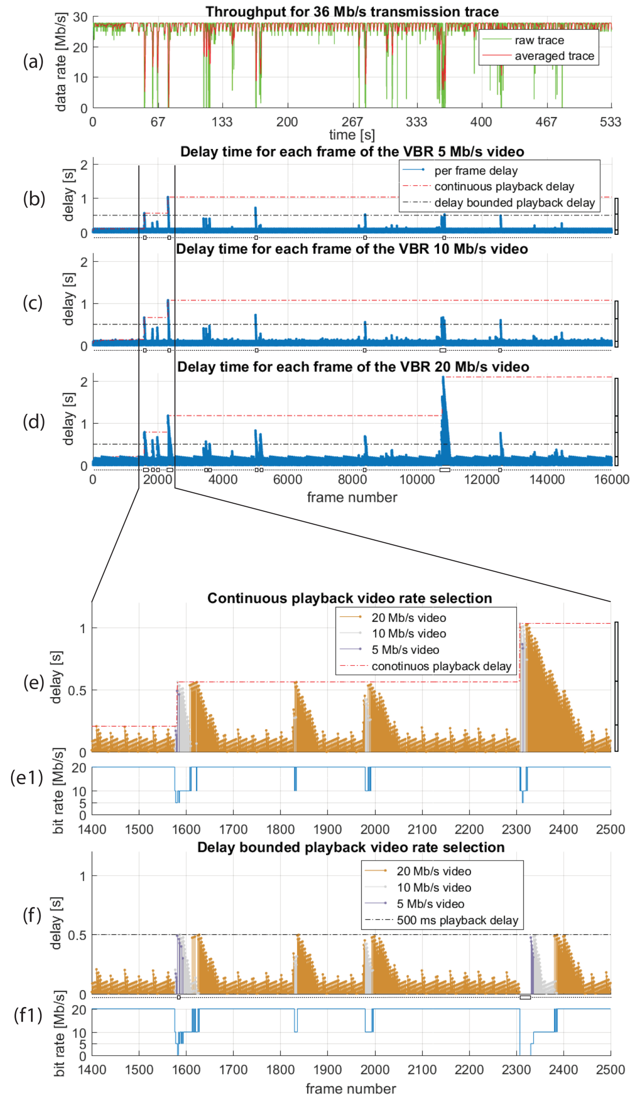

Fig. 8: Correlation between frame delay and the network condition

	Video Rates	5 Mb/s	10 Mb/s	20 Mb/s
Continuous playback	Playback freezes	3	3	4
	Total freeze time	1.032 s	1.074 s	2.102 s
	Accumulated playback delay	1.032 s	1.074 s	2.102 s
Delay bounded playback	Playback freezes	5	6	11
	Total freeze time	1.133 s	2.533 s	15.833 s
	Accumulated playback delay	0.5 s	0.5 s	0.5 s

TABLE II: Comparison of certain metrics in video playback

used, we experience 4 freezes and that the total freeze time is approximately 2 seconds. If bounded playback is used instead we will experience 11 freeze periods which accumulate to approximately 16 seconds. Similar numbers for the other video rates can be seen in Table II.

Even though the 20 Mb/s video experiences greater delay than the lower quality videos these delays only occur for a fraction of the frames recorded. This means that is is possible to use the 20 Mb/s video for the majority of the stream. By adjusting the video rate when frame delay is high it is possible to reduce playback delay and the number of freeze periods in the live video stream.

We propose two ways of adjusting the video rate depending on the playback method used. For continuous playback an initial playback delay is introduced. Whenever a frame is delayed beyond the initial delay we switch to the video rate which causes the lowest additional delay. If the delay decreases, we switch to the highest video rate that does not introduce further delay. An illustration of this policy is seen in Figure 8(e) where the different colours denote which video rate was used on the frame and 8 (e1) shows the video rate. We see that if this policy is applied on the traces it switches incorrectly in some cases, this is because the captured videos are not identical.

For the delay bounded method the policy is as follows: as long as frames from the 20 Mb/s video are delayed less than 500 ms they are played. If the delay is exceeded a lower video rate that still fulfil the requirements is used. If none of the video rates are able to fulfil the 500 ms delay requirement we experience playback freeze. Figure 8(f) illustrates the switching done to fulfil the 500 ms delay requirement and Figure 8(f1) shows the video rate. When the video rate drops to 0 Mb/s a playback freeze period occurs.

These two policies give better performance for both playback methods in terms of number of freezes and freeze time. The methods are inherently different, so choosing a method depends on the desired playback delay and the application of the video.

VII. Conclusion & Future work

Plain 802.11 multicast is inadequate for live video streaming. Additionally, live streaming impose requirements on the video playback delay. Increased reliability for continuous and delay bounded video playback can be achieved by jointly adapting TxR, RLNC and video rate based on PER. Our investigation is based on experimental data obtained while using a drone in an open field, which we analyse using a simple delay model. We show that for this particular case, TxR adaptation yield no improvement on the reliability, thus picking the highest rate is the best choice. Our recommendation is using a policy to adapt RLNC and video rate based on PER. For continuous playback, we propose picking the highest video rate, which introduces the least amount of playback delay. Thereby a high video rate is maintained for the majority of the time while minimizing the playback delay and freeze periods. For delay bounded playback the delay is constant. Here, we choose the highest video rate, which does not exceed the delay bound. Frames delayed beyond the delay bound are dropped. This policy minimize cumulative freeze time and the number of freeze periods.

As these policies are based on a simplified model, further investigation is required before applying them in real applications. Real implementation of RLNC would introduce coding delay and overhead. One would have to ensure that when switching video rate the newest frame should be compatible with the last frame transmitted. Furthermore, the policies are only based on feedback from a single receiver. How to weigh the feedback from multiple receivers is still an open problem since this is specific to the scenario. A real feedback system would also introduce delay.

Future investigations should focus on implementing this policy in a test bed, such that the model assumptions are eliminated. This enables investigation of how factors, which are unaccounted for in the current model, effect the performance of the proposed policies. Another interesting aspect is expanding the capabilities of the receivers, e.g. by allowing cooperative sharing among receivers.

References

[1] Lockheed Martin Corporation, "Indago UAS," http://www.lockheedmartin.com/us/products/procerus/indago-uas.html, 2016, online; accessed 28 November 2016.

[2] "Ieee standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part ii: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, pp. i–67, 2003.

[3] W. Yan, S. Yu, and Y. Cai, "Reliable multicast with network coding in lossy wireless networks," *Int. J. Communications, Network and System Sciences, 2010, 3, 816-820*, September 2010.

[4] O. Alay, T. Korakis, Y. Wang, and S. Panwar, "Dynamic rate and fec adaptation for video multicast in multi-rate wireless networks," August 2009.

[5] Y. Park, Y. Seok, N. Choi, Y. Choi, and J. M. Bonnin, "Rate-adaptive multimedia multicasting over ieee 802.11 wireless lans," vol. 1, pp. 178–182, Jan 2006.

[6] N. Stefano Paris, F. Gringoli, and A. Capone, "An innovative rate adaptation algorithm for multicast transmissions in wireless lans," *IEEE 77th Vehicular Technology Conference*, June 2013.

[7] Y. Bejerano, J. Ferragut, K. Guo, V. Gupta, C. Gutterman, T. Nandagopal, and G. Zussman, "Scalable wifi multicast services for very large groups," *IEEE ICNP'13*, 2013.

[8] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.

[9] Jason Robert Carey Patterson, "Video Encoding Settings for H.264 Excellence," http://www.lighterra.com/papers/videoencodingh264/, 2012, online; accessed 22 November 2016.

[10] R. L. Bruun, M. Eriksen, R. S. Mogensen, and K. W. Mortensen, "Measuring the maximum throughput of 802.11 mac sdu," *Worksheet portfolio*, November 2016.

[11] M. López Aguilera, J. Casademont Serra, and J. Cotrina Navau, "Ieee 802.11g performance in presence of beacon control frames," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2004.

[12] R. L. Bruun, M. Eriksen, R. S. Mogensen, and K. W. Mortensen, "Packetloss and correlation for udp wireless multicast," *Worksheet portfolio*, November 2016.

[13] ——, "Correlation animation tool," *Worksheet portfolio*, November 2016.

[14] ——, "Motivation for h264 source coding," *Worksheet portfolio*, November 2016.

[15] Steinwurf, "petro," https://github.com/steinwurf/petro, online; accessed 22 November 2016.

[16] "Flight video," https://youtu.be/Is5HdCOMl0E.

[17] R. L. Bruun, M. Eriksen, R. S. Mogensen, and K. W. Mortensen, "Estimating channel conditions at different rates," *Worksheet portfolio*, November 2016.

[18] ——, "Range test," *Worksheet portfolio*, November 2016.