Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3142496, 10 pages
https://doi.org/10.1155/2018/3142496

Research Article

Adaptive Video with SCReAM over LTE for Remote-Operated Working Machines

Ingemar Johansson,[1] **Siddharth Dadhich**,[2] **Ulf Bodin**,[2] **and Tomas Jönsson**[1]

[1]*Ericsson AB, Luleå, Sweden*
[2]*Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Sweden*

Correspondence should be addressed to Ingemar Johansson; ingemar.s.johansson@ericsson.com

Received 23 March 2018; Revised 7 June 2018; Accepted 17 July 2018; Published 2 August 2018

Academic Editor: Federico Tramarin

Remote operation is a step toward the automation of mobile working machines. Safe and efficient teleremote operation requires good-quality video feedback. Varying radio conditions make it desirable to adapt the video sending rate of cameras to make the best use of the wireless capacity. The adaptation should be able to prioritize camera feeds in different directions depending on motion, ongoing tasks, and safety concerns. Self-Clocked Rate Adaptation for Multimedia (SCReAM) provides a rate adaptation algorithm for these needs. SCReAM can control the compression used for multiple video streams using differentiating priorities and thereby provide sufficient congestion control to achieve both low latency and high video throughput. We present results from the testing of prioritized adaptation of four video streams with SCReAM over LTE and discuss how such adaptation can be useful for the teleremote operation of working machines.

1. Introduction

Working machines comprise a large set of heavy machinery, of which the majority are typically mobile. These machines perform tasks that require experienced drivers to perform well in terms of safety, productivity, and cost. Safety includes human safety in the direct vicinity of the machine as well as damage prevention of the surrounding environment. Productivity relates to limiting the time consumed for performing the given task, while cost includes limiting the energy consumption and wear and tear on the machine. These aspects are important for the operation of mobile working machines.

Teleremote operation and automation are under development by companies including Caterpillar [1], Atlas Copco [2], Kotmatsu, and Sandvik. Most research in remote control works toward increasing the perception of remote operators, for example, the visualization of different perspectives of the construction machine in its environment by body-mounted fish-eye cameras [3, 4]. The display environment has also been investigated such as in [5], where a dome display with panoramic camera is used.

In [6], the remote control of an excavator over a wireless IP network using a head mount display (HMD) with an end-to-end latency of 180 ms reports a 164% increase in the cycle time of operation time for remote control, compared to manual operation. The amount of tolerable latency for teleoperation varies a lot between different experiments, and the variation in latency (jitter) can be more detrimental to performance than the latency itself [7].

The progress toward the automation of moving working machines is discussed in the following five steps: (1) manual operation, (2) in-sight remote operation, (3) teleremote operation (4), assisted teleremote (semiautomation) operation, and (5) fully autonomous operation [8–10]. Steps three and four require good video feedback, which is challenging to achieve over wireless networks since the bandwidth can vary frequently and unpredictably. For example, reflections from surrounding objects result in multipath propagation, which interfere with receiving devices to cause errors in the received data, making the wireless link to adapt to stronger coding and consequently lower transmission rates. As a result, data get buffered in the networking devices and are delayed even before transmission over the wireless link. If the buffer gets full, all incoming packets are discarded.

Figure 1: Remote-control operation of a wheel-loader. The experimental wheel-loader (left) and remote-control station (right). The operator sees the video streams from six cameras mounted on the machine. The chair is a moving platform that obtains roll and pitch data from an IMU mounted in the machine. GPS and various other signals (RPM, machine speed, etc.) are used to simulate a human-machine interface (HMI) for driver assistance.

Without video streams adapting to lower transmission bit rates, packets are either buffered or discarded. With buffering, a safety risk exists since the video can become more than a second old. Alternatively, when packets are discarded, the resulting video degrades in quality, possibly compromising the task of the remote operator.

Remote operation of heavy working machines is a time-critical application in need of reliable video transmission from multiple cameras over wireless links and the ability to prioritize task-critical and safety-critical video streams over others. In this paper, we examine the use of Self-Clocked Rate Adaptation for Multimedia (SCReAM), first introduced in [11], to adapt video streams to lower bit rates through compression setting in cameras limiting the frame rate and resolution of the encoded video. We demonstrate, for the first time, the functionality of SCReAM with prioritized scheduling of video streams on a small scale experimental mobile platform using a public LTE network.

SCReAM provides a rate adaptation algorithm that can control the compression used for multiple cameras and thereby provide sufficient congestion control to achieve both low latency and high video throughput over wireless networks with varying bandwidth. Further, it can use priorities to control the differentiated adaptation of multiple media streams.

Two alternatives to SCReAM are (1) Google congestion control (GCC) [12] and (2) NADA [13]. A qualitative comparison between SCReAM, GCC, and NADA is presented in [14]. A quantitative comparison between SCReAM, GCC, and NADA is however difficult to perform due to the different implementation status of these proposed algorithms [14]. Moreover, out of these algorithms, only SCReAM supports prioritization among competing streams.

The main contributions of the paper are (1) the integration of SCReAM into the industrial application of the remote operation of a Volvo L180H wheel-loader to offer video adaptation and (2) the evaluation of SCReAM for multiple low-delay video streams sent upstream over LTE, including the use of priorities for the different video streams. The use of SCReAM with prioritized scheduling of video streams over cellular wireless networks such as LTE can be extended to other teleoperated machinery or vehicles.

2. Background

2.1. Teleremote Setup. We have a teleremote system for a Volvo L180H wheel-loader as shown in Figure 1 and described in more detail in [15]. It is determined that, for this current setup, the productivity (ton/hour) for loading gravel in short loading cycles decreased by 42% using remote control compared to manual operation [15]. The main reason for the productivity loss is the lack of perception that the environment surrounding the machine and glitches in the video streams from cameras are identified as a major contributing factor for degraded operator experience. Figure 2 shows the proposed setup in terms of network components.

The remote-control system is built with parts that are commercially available. This poses a few limitations but also creates new challenges to solve. The setup uses Panasonic WV-SBV111 IP cameras with a 104° horizontal field of view. Three cameras in the front and three in the rear side of the wheel-loader provide a complete view of the surrounding areas.

The cameras are easy to access via HTTP commands for sending rate control purposes and have proven to be robust under different environmental light conditions. The transmission mode that works best when mounted on a mobile machine is variable bitrate mode (VBR).

2.2. Challenges and Requirements for Teleremote Operation. The task is to achieve the desired productivity at a reasonable cost while meeting the safety requirements. Safe and efficient teleremote operation requires recreating the machine environment in real time. It is identified in [16–18] that operators make their decisions based on their vision, the sound from the surroundings, and vibrations from the machine. The lack of availability of these basic human sensor systems presents the problem of environmental perception.

With the current camera setup and the remote-control station, depth perception is lost, and the human field of view is compressed and reduced to the display dimensions. The latency in audio and video is critical for teleremote operation. Wireless network jitter can cause many frames to be dropped or played out at an irregular pace, resulting in sluggish or choppy video that can cause operator fatigue. A common

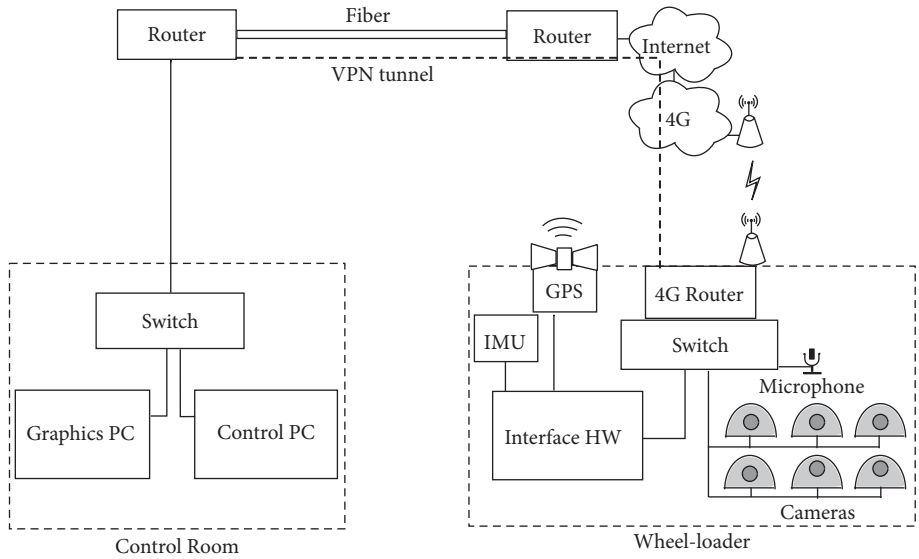

Figure 2: The setup consists of several pieces of network equipment at the remote-control station and the wheel-loader connected by a partially wired and partially wireless network (LTE). From a logical point of view, the 4G router in the wheel-loader connects to a router in the same building as the control station through a VPN tunnel via the backbone network of the service provider (Telia) and the Internet. The 4G base station router is connected to the gateway router for the control station equipment with a 600-meter-long fiber cable. The graphics PC handles the data from cameras and microphone, while the control PC sends the control packets to the wheel-loader and receives feedback data for moving the motion platform and plotting the HMI.

application of wheel-loaders includes loading material and dumping it onto trucks. In this application, there is a risk of collision with the dumper truck if the video stream from the front camera underperforms even slightly.

2.3. Differentiated Prioritization of Streams. The requirements of the teleremote solution on the communication link are foremost low latency, minimal loss, and high throughput. Different tasks and situations require different streams to be prioritized. Therefore, the prioritization of certain streams over others is highly desired.

Prioritization is also needed across streams of different data types such as video and point cloud data. Apart from camera streams and point cloud data, other data (such as monitoring and control) are also transmitted and received by the remote-controlled working machines. In the risk of collisions, the control data should have the highest priority.

Preferably, prioritization should be offered by the network layer, e.g., using the QoS Class Identifier (QCI) in 3GPP Long-Term Evolution (LTE) networks [19]. Telecom operations do not always offer such services, or they are quite expensive. Hence, mechanisms that can prioritize end-to-end independence of intermediate communication links are attractive. Here, the end-to-end prioritization approach refers to any mechanism that resides at the transport layer (TCP/UDP) or above and not in the IP network layer or below, as in the case with QCI in LTE.

In the future, 5G will provide better prioritization mechanism than LTE [20]. Although 5G will offer more stable capacity to the prioritized flows, it may not be immune to all disturbances affecting the radio channel. The ability to prioritize end-to-end will still be valuable as it provides a means to rapidly react to changing stream priorities as well as changing link capacity.

2.4. Video Coding and Compression. Raw video recording is compressed to a great extent in cameras by special encoding to save capacity of communication channels when transferring video over a network. H.264/MPEG-4 AVC standard [21] is a widely accepted video coding standard used in most IP cameras including the one used here. This standard uses both temporal and spatial redundancy to compress the signal. Spatial encoding provides less compression than temporal encoding but comes with an advantage that parts of the encoded signal do not require any other information for decoding. Parts of the video signal encoded using spatial redundancy only are called I-frames, where I stands for intracoded picture.

Temporal redundancy is used for further compression. In temporal redundancy, the current part of the signal is constructed using the previous part as a reference point, and only the difference between these parts is transmitted. The temporal encoding produces P- and B-frames where P stands for predictive and B stands for bipredictive. While I-frames are largest in size, B-frames are the most compressed as they use I- and P-frames as references, both from historical parts and from the future signal.

3. SCReAM

3.1. Protocol Description. SCReAM was originally devised for the end-to-end congestion control of real-time media such as audio and video for WebRTC (Web Real-Time Communication), which is widely used in the gaming community.

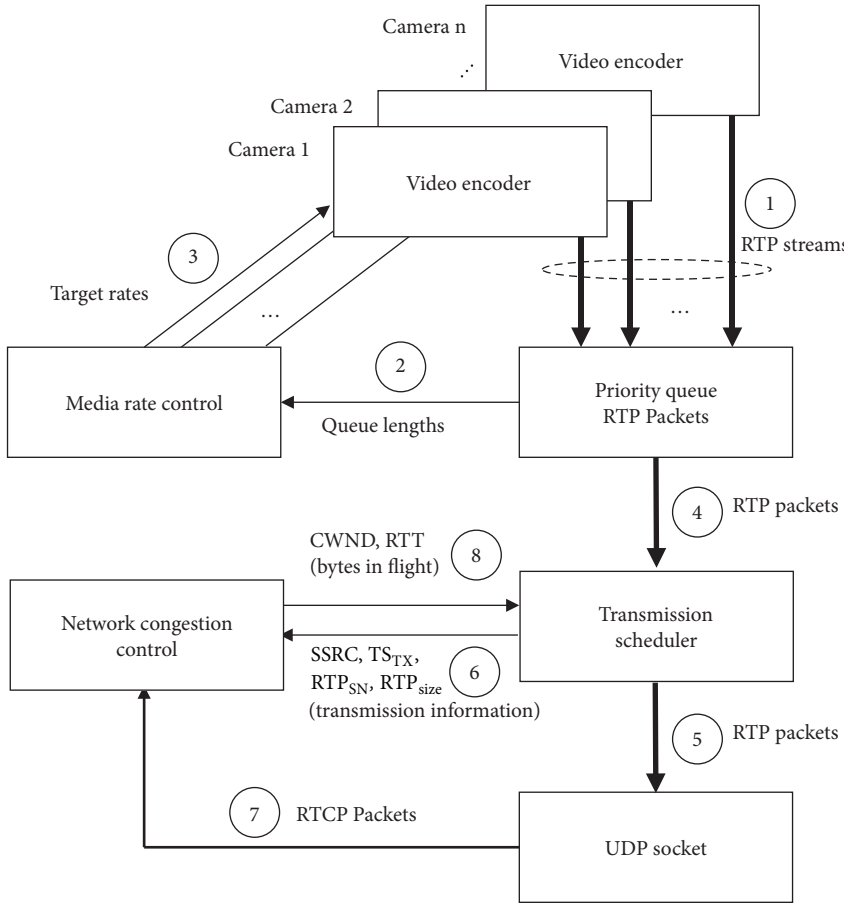

Figure 3: Schematic view of the SCReAM congestion control algorithm on the sender side. The operation of SCReAM is as follows: (1) RTP packets from sources are sent to RTP packet queues component; (2) the queue lengths are reported to the media rate control component, which (3) communicates target bit rates to the media sources. The transmission scheduler (4) fetches the RTP packets from the packet queues component and then (5) sends them to the UDP socket according to its congestion control and packet pacing functions and it (6) updates the network congestion control component on transmission timestamps (TS_{TX}), sequence number (RTP_{SN}), and size (RTP_{size}) of each RTP packet. The received RTCP packets are (7) forwarded to the network congestion control, which then calculates CWND (congestion window) and RTT (round trip time) and (8) updates the transmission scheduler.

SCReAM was developed to give robust congestion control especially for encoded video over cellular links. It can also be applied to other data streams such as audio or slide-share content; however, its main application is in video because video typically needs much higher bandwidth compared to other streams.

SCReAM does not interfere with encrypted media as it does not require inspection of the RTP payload for its function. It only requires access to the RTP sequence number and the SSRC (synchronization source identifier) from the RTP headers that are not encrypted.

SCReAM is a sender congestion control algorithm, which means that all intelligence concerning congestion control and rate control of the media sources is on the sender side. The SCReAM receiver side is very simple in comparison since it only generates timely feedback to the sender. As a simple comparison, the SCReAM sender side is more than 2000 lines on C++ code, whereas the receiver code is only approximately 300 lines of C++ code.

3.2. Protocol Operation. A schematic view of the SCReAM congestion controller on the sender side is shown in Figure 3. The congestion control is divided into two main parts.

(1) **Network congestion control** ensures that the delay over the network is kept as low as possible. This is done by setting an upper limit, CWND (congestion window), on how much data can be in the network. The transmission scheduler uses CWND to determine when to make RTP packets available for transmission and RTT (round trip time) to implement packet pacing, which determines when the RTP packets shall be sent to the UDP socket for transmission. Packet pacing is used to mitigate the problem of multiple RTCP packets arriving very close to each other causing several RTP packets to be transmitted, faster than intended.

(2) **Media rate control** regulates the target bitrate of each media component, e.g., the different video encoders of cameras. It determines target bitrates from the

queue lengths of the RTP packet queue component. When an RTP queue length exceeds a certain threshold, it is reported to the media rate control, which instructs the source media component to immediately reduce its bitrate by changing to a different encoding.

The network congestion control and media rate control complement each other in mitigating the risk of extensive delays and packet loss. The implementation of the network congestion control is aimed at avoiding overloading the network while the implementation of the media rate control is aimed at avoiding the fact that the sources generate more data than the network can handle.

With SCReAM, the RTP packets from the IP cameras are not directly sent to the outgoing network interface; instead, they pass through sender queues (the RTP packet queues component in Figure 3), which are implemented for each media component to ensure efficient congestion control. This arrangement allows SCReAM to prioritize between data streams and to avoid large network queuing delays. In other words, for the SCReAM implementation evaluated in this paper, the RTP packet queue component can implement priority scheduling to give precedence to some IP cameras over others.

It should be noted that the sender RTP packet queues are most often empty. Their main purpose is to serve as shock-absorbers when the IP cameras output large amounts of data, e.g., due to quick changes in the video picture or when the network throughput drops rapidly and the IP cameras do not respond to the reduced target bitrates notified by the media rate control by decreasing the output bitrate quick enough.

To summarize using Figure 3, the operation of SCReAM follows this; (1) RTP packets from one or more sources are sent to RTP packet queues component, which makes the packets available to the transmission scheduler using a priority scheme. The queue lengths are (2) reported to the media rate control component, which (3) communicates target bit rates to the media sources, i.e., the video encoders of the IP cameras. This completes the operation of the media rate control part of SCReAM.

The transmission scheduler (4) fetches the RTP packets from the packet queues component and then (5) sends them to the UDP socket according to its congestion control and packet pacing functions. When sending packets to the UDP socket for transmission, it (6) updates the network congestion control component on transmission timestamps (TS_{TX}), sequence number (RTP_{SN}), and size (RTP_{size}) of each RTP packet, uniquely identified by the SSRC value (SIP Synchronization Source Identifier). RTCP packets sent from the receiving side (i.e., the media decoders displaying the sent video) carrying the timestamps, sequence numbers, and SSRC of received RTP packets are (7) forwarded to the network congestion control component. Finally, the network congestion control component calculates CWND (congestion window) and RTT (round trip time) and (8) update the transmission scheduler.

3.3. Protocol Behavior. The network congestion control of SCReAM is similar to how TCP behaves; the main difference is that SCReAM does not retransmit lost packets. Similar to TCP, network congestion control is self-clocked. This means that packets are transmitted as long as feedback is received. This prevents the transmission link from becoming overloaded with data, which is good when the throughput decreases rapidly. This helps fulfill the safety requirements for teleremote working machines.

SCReAM, besides being adaptive to packet loss, also detects the increase in network queue delay and adjusts the transmission rate to maintain minimal network delay. This algorithm is based on timestamps of the sender and receiver sides. However, synchronized clocks at the sending and receiving sides are not a necessity. It is necessary though that they use the same clock frequency, or that the clock frequency at the receiver can be inferred reliably by the sender.

SCReAM supports explicit congestion notification (ECN). ECN is a technique that can be used by network nodes to signal when the available throughput decreases and packets start to be queued. The ideal outcome of ECN is near zero packet loss, which is beneficial especially for real-time video that is quite sensitive to packet loss. Furthermore ECN, being an explicit signal of congestion, gives a clear indication of congestion that can be acted upon promptly. This results in reduced end-to-end delay, giving additional stability to congestion control. SCReAM also supports (L4S) low latency low-loss scalable throughput [22], which is a novel technology that can produce very short queue delays.

Packet pacing is implemented to avoid packets being transmitted in bursts. This prevents overflow in network queues due to packet losses. For example, large I-frames generated by video coders can overflow network queues with packet loss as a result. In the worst case, this can cause the video to freeze. Packet pacing reduces the queue build-up in network components and thus reduces the risk of packet loss.

As mentioned above, SCReAM can handle and prioritize between two or more video streams, which is beneficial in applications where differentiated prioritization is needed. This approach also makes the congestion control more efficient, giving better control over the end-to-end delay. Another benefit is that the flow prioritization is straightforward and simple to implement.

The stream prioritization, examined in this paper, is based on credit-based weighted scheduling. Each stream is given a scheduling weight in the range of $[0, 1]$. When one stream is scheduled to transmit an RTP packet, credit is given to the other streams in proportion to the size of the transmitted RTP packet, scaled by the scheduling weight. Credit is accumulated and used by streams to increase their transmission likelihood. When a stream transmits, its accumulated credit is decreased by the size of the RTP packet being transmitted. This means that all streams are allowed to transmit, and their transmission rate depends on their weight in relation to the weight of other streams. For example, if two flows have the scheduling weights 1.0 and 0.5, one flow will receive roughly two-thirds of the available bandwidth, while the other flow will receive a one-third share.

When the sources transmit constantly, for example, simulated video coders, the weighted credit-based scheduling

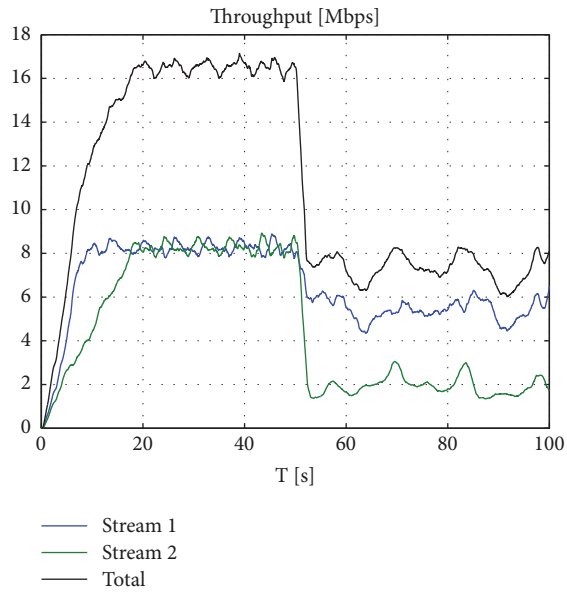

Figure 4: Simulation result showing the prioritization behavior of SCReAM protocol under bandwidth deficiency. At time T=50s, the link bandwidth is reduced from 20 Mbps to 8 Mbps. SCReAM protocol allows distributing the bandwidth to the two streams according to their priority.

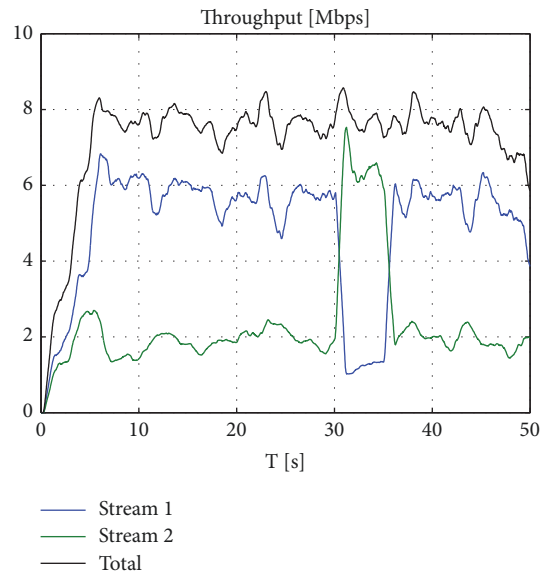

Figure 5: Simulation result showing run-time priority change behavior of SCReAM protocol. The scheduling weights are flipped at $T = 30s$ and restored at $T = 35s$. SCReAM protocol redistributes the bandwidth to the two streams according to run-time changes in their priorities.

works perfectly. However, the actual video coders have both a poor regulatory response for a given target bitrate and a slow transient response for changes in target bitrate. This results in insufficient weighted credit-based scheduling to maintain weighted proportional bitrates of the sources. To handle this, an extra mechanism is introduced in SCReAM that enforces a conservative update of the target bitrate by reducing it 10% for each source that is exceeding its share of the bandwidth. This extra mechanism runs every five seconds.

3.4. Simulation Results. To exhibit the prioritization behavior of the SCReAM protocol, we perform simulation experiments. Figure 4 shows the functionality of SCReAM in a simulated setup in which two simulated video streams are transmitting over a link with a varying bandwidth similar to a wireless link for remote control. Stream one has a scheduling weight of 1.0, and stream two has a weight 0.2. Initially, the link bandwidth is 20 Mbps, and both streams transmit 8 Mbps each. At $T = 50s$, the link bandwidth is reduced to 8 Mbps. Ideally, stream one should receive a bandwidth of $8 * 1.0/(1.0 + 0.2) = 6.7$ Mbps, and stream two should receive a bandwidth of $8 * 0.2/(1.0 + 0.2) = 1.3$ Mbps. Figure 4 shows that this result is nearly achieved. It can also be seen that, in SCReAM, the flow prioritization is only applied when the link bandwidth is limited, which is the rational choice.

The stream priority can also be changed during run time, as shown in the simulated example in Figure 5. In this setup, the link bandwidth is fixed at 8 Mbps. Stream one starts with weight 1.0, while the stream two starts with weight 0.2. The scheduling weights are flipped at $T = 30s$ and restored at $T = 35s$. It can be seen that the corresponding responses in the bitrates of the two streams are quick. This is desired when the wheel-loader shifts into reverse, in which it is needed to achieve better quality streams for the rear camera. However, we have not yet evaluated this feature in conjunction with different video encoders, which can be slow to react to the large changes in target bitrates as their internal quantizer tables may only be changed at new I-frames.

4. Experiment Results

The SCReAM code is available as open source at [23]. This code is used in a setup with a number of commercially available IP cameras, which are rate-controlled via HTTP to get requests from the SCReAM congestion control application.

The SCReAM congestion control software is implemented on a Raspberry Pi 3 PC with Linux Raspbian OS. The solution is prototyped in a small mobile unit (shown in Figure 6), with an LTE modem and 4 Panasonic WV-SBV111 IP Cameras. The LTE link is the public Swedish Telia network, and the location is the main campus of Luleå University of Technology, which means that the LTE access is shared with other traffic in the area. The distance between the LTE base station and the location where the experiment is conducted is ~400m. The LTE base station is configured for channel bandwidth of 20 MHz with 50 Mbps maximum uplink bitrate.

During the experiment, the mobile platform is in constant motion, cameras are sending 25 frames per second and they are configured with different priorities. The front camera has the highest priority 1.0, the rear camera has a priority level of 0.3, and the left and right cameras have a 0.1 priority level. This means that the front camera gets the largest share of the available bandwidth when the network becomes congested.

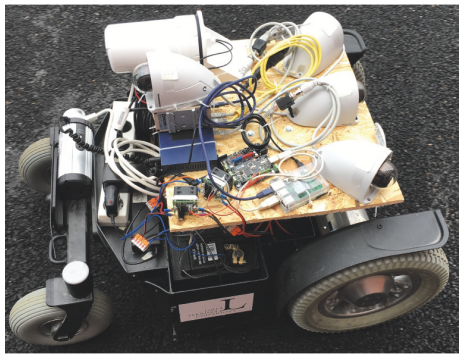

Figure 6: Mobile platform used for testing SCReAM with LTE. One camera is facing front, while two cameras are mounted facing left and right, and one more camera is facing the rear side.

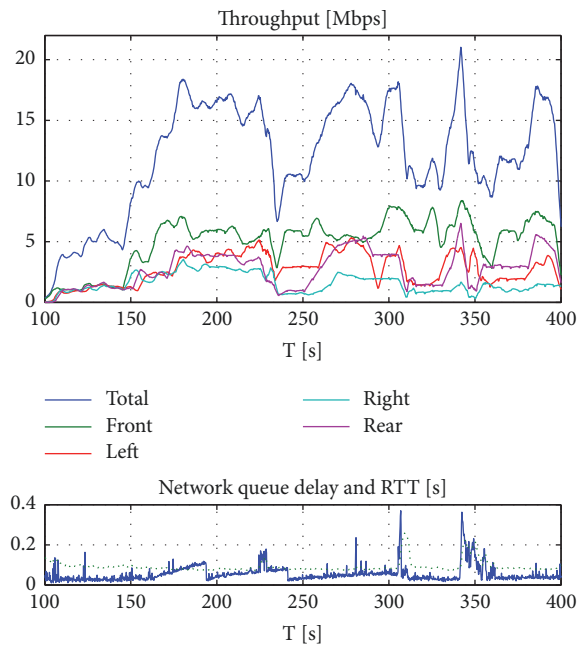

Figure 7: Experimental results using SCReAM over LTE on a mobile platform with four cameras with different priorities. The top plot shows throughput for the four cameras as well as the total throughput. As expected, the front camera obtains the maximum share of the available bandwidth. The bottom plot shows the average queuing delay in blue and the round trip time (RTT) in green. A low value of RTT is maintained throughout the experiment, except at T=305.

Figure 7 shows the experimental results used to evaluate stream prioritization behavior of SCReAM over LTE. The throughput and queuing delays in the sender queue of each stream were logged in the setup. The available throughput varies over time, and the role of SCReAM congestion control is to avoid high values of the queuing delays in the network when throughput decreases. It can be seen that this goal is achieved in many situations, except for a few cases where the throughput drops very quickly.

Figure 7 shows the queue delay in a slowly increasing trend, which decreases to low values at $T = 190$, 240, and 315s. This anomaly is due to clock drift in the Raspberry Pi 3. The outside temperature when this experiment was run was -20° Celsius, and this caused the Raspberry Pi to alter its internal clock frequency slightly, resulting in a clock drift. The SCReAM congestion control has a built-in mechanism to detect and repair this anomaly, which is visible at $T = 190s$, 240s, and 315s. This correction mechanism removes the issues that clock drifts can cause on SCReAM performance.

The stream prioritization works fairly well, and the front camera always obtains the largest share. However, it can be seen in Figure 7 that the prioritization has not been perfect since the rear camera, although having a higher priority, does not show a consistently better performance over the left camera. The main reason is that the IP cameras rarely deliver the target bitrate set for each IP camera by the SCReAM congestion control. The actual video bitrate is heavily dependent on what the camera lens sees; for example, the static image can give almost zero bitrate; however, panoramic sweeping due to a turn can give a bitrate that is close to or sometimes even higher than the target bitrate. All this, in combination with varying network throughput, makes perfect stream prioritization difficult to achieve.

Figure 8 shows a close up of what happens when the throughput drops at $T = 305s$. The throughput drops from 20 Mbps to below 10 Mbps in less than a second. It is clear that the front camera reduces the bitrate very little, whereas the other cameras reduce the bitrate much more. The queue delays increase up to ~400ms and later come down to small values, after a short while. While it is possible to make the congestion control react faster to increase in queuing delay, this will make it more sensitive to delay jitters that can occur in LTE networks. ECN (explicit congestion notification) [24] is a network enhancement technology that gives a more clear congestion signal to the SCReAM protocol and therefore can provide better control over the network queue delays. ECN is not enabled in this experiment, however.

Figure 8 also shows the sender queue delay for each of the four IP cameras. As mentioned before, the sender queues are implemented to avoid network queuing delays growing in an uncontrolled way when, for example, the network throughput drops. The sender queuing delay for the front camera is much lower compared to other cameras. The fast queuing delay drop, which can be seen as the jagged look of the plots, is because of the packets being discarded in the sender queue when they are delayed for too long. Here as well, ECN support can improve the performance as it gives a clear signal to reduce the bitrate.

5. Discussion

The mobile platform setup (Figure 6) is a down-sized realization of the wheel-loader remote-control setup (Figure 2). The traffic variability in the public LTE network makes it impractical to do repeatable experiments. Nonetheless, the simulated results show the expected prioritization behavior and also the run-time prioritization change behavior. The experimental results confirm the intended behavior of prioritization works and that the highest priority stream is

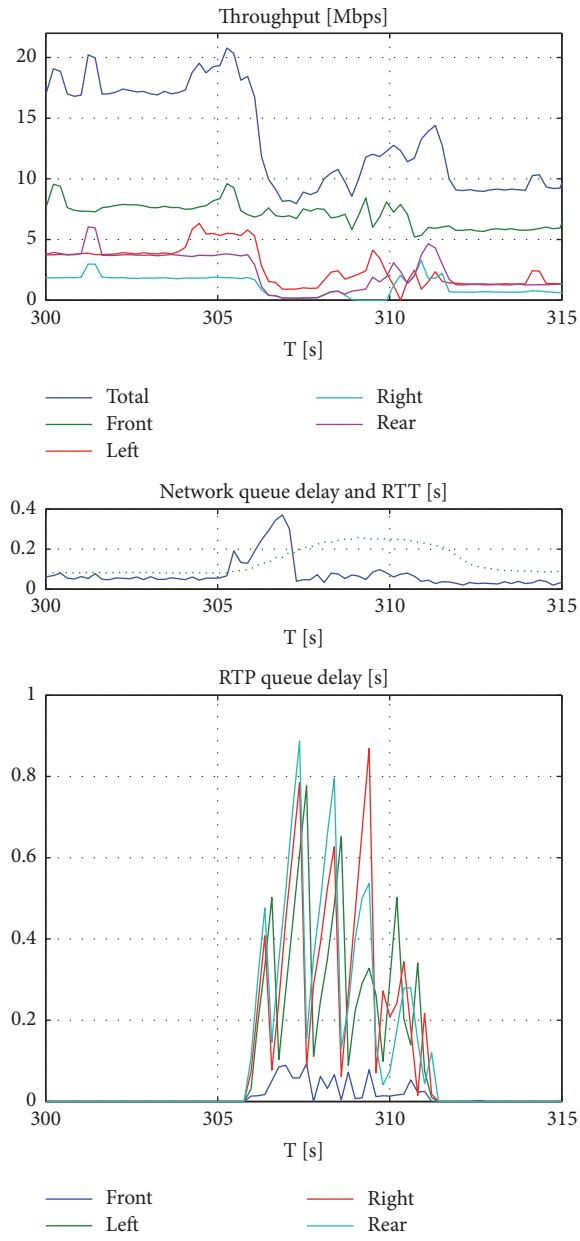

FIGURE 8: Close up of the experimental results at $T = 300 - 315s$. The top plot shows the throughput of all cameras and the total throughout, which suddenly drops to half at $< 1s$, starting at $T = 305$. The middle plot shows the average queuing delay in blue and the round-trip-time (RTT) in green. The bottom plot shows the queuing delay for each stream. The peak queuing delay for the top front camera is 8 to 9 times lower than that of other cameras.

significantly less affected by bandwidth drops compared to less prioritized streams.

SCReAM applies packet pacing to avoid packet loss. The problem is not fully solved when the I-frame is delayed in the sender queue, which can cause additional end-to-end delay. An operator will then experience video freezes for a split second. I-frames are the main source of the problem as they are generally quite large and can cause packet loss if the link bandwidth is limited. A lost I-frame can also cause a short freeze in the video stream as successful decoding of the following P-frames depends on the I-frame.

To solve this, video coders in IP cameras can be optimized to enable better performance for wireless remote-control applications. The most important modification can reduce the impact of I-frame generation. The techniques for this are already available in a x264 encoder under the name "periodic intra refresh", where the refresh image is sliced column-wise, and one column is refreshed per frame. This technique spreads the I-frame over time, without any additional delay; therefore, the large bitrate increase is reduced, which is common with normal I-frame generation. The proposed use of periodic intrarefresh is not recommended in certain cases such as scene-cuts (common in movies), but scene-cuts are not expected to occur in remote-control applications.

The access to video coding algorithm parameters in IP cameras is limited to setting the I-frame interval, video quality, and target bitrate. For further improvements, the ability to fine tune encoder settings in IP cameras and video coders is needed. Neither the used IP camera nor any other cameras or video coders that have been investigated provide the parameters needed to make a more accurate tuning of target bitrate, which is needed for remote-control applications. However, assuming a camera/video coder with all of the available parameters, modification to the presented solution is minor.

ECN is a network support function that improves the stability of congestion control. It is recommended to use industrial LTE routers with ECN support; one example of an LTE router that supports ECN is a Cisco IR829 router. Furthermore, LTE-operated networks should be configured to support ECN. The video in the remote-control application is transported in the uplink direction, which is sufficient to ensure that the ECN bits are not cleared by network nodes.

QoS support can be beneficial, especially if the remote-control application is operated in a public cellular network. In this case, it is important to ensure that the solution also works in heavily congested network conditions.

6. Conclusion and Future Work

The remote operation of mobile working machines requires a minimal loss and low latency video transmission mechanism to guard against the occasional pitfalls of the wireless link. Loss or excessive delay of an I-frame is the reason for sluggish and choppy video stream, which is unsuitable for remote operation application.

SCReAM provides the mechanism to detect congestion and adjust the sending bit rates of the cameras providing congestion control without the need of packet acknowledgment for each received packet. It uses fields in the RTP header to do this. Additionally, SCReAM supports the prioritization of certain streams over others. If the network bandwidth decreases, this feature can help ensure the safe operation of the remote-operated machine by prioritizing the most important data stream; for example, the control signals or the camera pointing to the moving direction are prioritized over other less important streams like side cameras.

The experimental results with a remote-controlled mobile platform running SCReAM over LTE show that the prioritization of different streams works as expected in most situations. SCReAM supports changing stream priorities in run time. Teleremote applications require run-time changes in priorities for both safety and productivity. However, this feature has not yet been evaluated, and further study is needed to see how this performs in a working machine environment with video encoder behavior taken into account.

L4S is a future enhancement of ECN that can provide very low queue delays. Future studies will investigate the performance of SCReAM with ECN and L4S and demonstrate its usefulness in this context.

Data Availability

The code from this work is uploaded at https://github.com/EricssonResearch/scream with the extension with ECN and L4S. The data collected during the evaluation of SCReAM for the remote-control application can be made available upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was completed with support from the program "Fordonsstrategisk Forskning och Innovation, FFI", Vinnova Grant 2017-01958. The authors thank Robert Hedman and Olov Sehlin for their work in setting up the mobile platform to make it possible to run the SCReAM experiments.

References

[1] M. Glover, "Caterpillar's Autonomous Journey - The Argument for Autonomy," in *Proceedings of the SAE 2016 Commercial Vehicle Engineering Congress*.

[2] A. A. Dobson, J. A. Marshall, and J. Larsson, "Admittance Control for Robotic Loading: Underground Field Trials with an LHD," in *Field and Service Robotics*, vol. 113 of *Springer Tracts in Advanced Robotics*, pp. 487–500, Springer International Publishing, Cham, 2016.

[3] S. Iwataki, H. Fujii, A. Moro, A. Yamashita, H. Asama, and H. Yoshinada, "Visualization of the surrounding environment and operational part in a 3DCG model for the teleoperation of construction machines," in *Proceedings of the 8th Annual IEEE/SICE International Symposium on System Integration, SII 2015*, pp. 81–87, Japan, December 2015.

[4] W. Sun, S. Iwataki, R. Komatsu, H. Fujii, A. Yamashita, and H. Asama, "Simultaneous tele-visualization of construction machine and environment using body mounted cameras," in *Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics, ROBIO 2016*, pp. 382–387, China, December 2016.

[5] C. A. James, T. P. Bednarz, K. Haustein, L. Alem, C. Caris, and A. Castleden, "Tele-operation of a mobile mining robot using a panoramic display: An exploration of operators sense of presence," in *Proceedings of the 2011 7th IEEE International Conference on Automation Science and Engineering, CASE 2011*, pp. 279–284, Italy, August 2011.

[6] C. L. Fernando, M. Y. Saraiji, Y. Seishu, N. Kurio, K. Minamizawa, and S. Tachi, "2P1-D08 Study on Telexistence LXXXI : Effectiveness of Spatial Coherent Remote Driving Experience with a Telexistence Backhoe Under Hazardous Conditions," *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, vol. 2015, no. 0, 2015.

[7] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1231–1245, 2007.

[8] B. Frank, L. Skogh, R. Filla, A. Fr, and M. Alaküla, "On Increasing Fuel Efficiency By Operator Assistant Systems in a Wheel Loader in," in *Proceedings of the International Conference on Advanced Vehicle Technologies and Integration*, pp. 155–161, 2012.

[9] J. M. Roberts, E. S. Duff, and P. I. Corke, "Reactive navigation and opportunistic localization for autonomous underground mining vehicles," *Information Sciences*, vol. 145, no. 1-2, pp. 127–146, 2002.

[10] S. Dadhich, U. Bodin, and U. Andersson, "Key challenges in automation of earth-moving machines," *Automation in Construction*, vol. 68, pp. 212–222, 2016.

[11] I. Johansson, "Self-clocked rate adaptation for conversational video in LTE," in *Proceedings of the the 2014 ACM SIGCOMM workshop*, pp. 51–56, Chicago, Illinois, USA, August 2014.

[12] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Analysis and design of the google congestion control for web real-time communication (WebRTC)," in *Proceedings of the the 7th International Conference*, pp. 1–12, Klagenfurt, Austria, May 2016.

[13] X. Zhu, M. Ramalho, S. Mena, P. Jones, J. Fu, and S. DAronco, "Nada: A unified congestion control scheme for real-time media," in *Internet-Draft draft-ietf-rmcat-nada-07, IETF Secretariat*, vol. 2018.

[14] L. D. Cicco, G. Carlucci, and S. Mascolo, "Congestion Control for WebRTC: Standardization Status and Open Issues," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 22–27, 2017.

[15] S. Dadhich, U. Bodin, F. Sandin, and U. Andersson, "From tele-remote operation to semi-automated wheel-loader in," in *Proceedings of 2nd European Conference on Materials*, 2018, in press.

[16] A. Hemami, "Fundamental analysis of automatic excavation," *Journal of Aerospace Engineering*, vol. 8, no. 4, pp. 175–179, 1995.

[17] P. J. Lever, "An automated digging control for a wheel loader," *Robotica*, vol. 19, no. 05, 2001.

[18] U. Andersson, *Automation and Traction Control of Articulated Vehicles*, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 2013.

[19] H. Ekström, "QoS control in the 3GPP evolved packet system," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 76–83, 2009.

[20] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[21] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–142, 2006.

[22] B. Briscoe, K. D. Schepper, and M. Bagnulo, "Low latency, low loss, scalable throughput (l4s) internet service: Architecture, Internet-Draft draft-briscoe-tsvwg-l4s-arch-02," in *IETF Secretariat*.

[23] J. Ingemar, Scream, https://github.com/EricssonResearch/scream (2018).

[24] D. Black, "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation," RFC Editor RFC8311, 2018.