

Congestion Control of WebRTC, video streaming over Wireless networks



Contact:

Mohammad Rajiullah mohammad.rajiullah@kau.se
Department of Computer Science,
Karlstad University

Background:

We have a traxxas RC car with a raspberry pi, GoPro Hero cam and a 5G Modem. To communicate with the car over the network for video streaming, issuing control commands, etc., we have developed a Web Real-Time Communication (WebRTC) framework. The framework is built on Pion (<https://pion.ly/>). A foundation library for controlling the servo motors (<https://gobot.io>) are also available.

WebRTC has a challenge to offer smooth real-time services (e.g. video and audio conferencing) over wireless networks. IETF RMCAT (RTP Media Congestion Avoidance Techniques) working group specify several congestion control mechanisms for RTP (real-time transport protocol) flows over UDP, used by WebRTC. Some of these congestion control mechanisms are Network Assisted Dynamic Adaptation (NADA), Google Congestion Control (GCC) and Self-Clocked Rate Adaptation for Multimedia (SCReAM). NADA requires network support and not all of these algorithms are optimized for wireless networks.

Task:

Implement SCReAM <https://github.com/EricssonResearch/scream/> as an interceptor in Pion <https://github.com/pion/interceptor> and evaluate the performance of SCReAM under a 5G network. The task will possibly require adaption of gstreamer/ffmpeg parameters.

The task requires good knowledge about congestion control algorithms, network testing, c and reading standard/scientific documents. It also requires knowledge about golang and h264 encoding/decoding, gstreamer/ffmpeg and webrtc standards.

Bonus task: Implement GCC in Pion and evaluate its performance jointly with SCReAM under the 5G network.

Epic:

The system will build upon previous project(s) : <https://git.cs.kau.se/research/dwr/picar1> that have done the basic work.

The exact epics and sub-task will be discussed and evaluated with the development team. but should consist of the following:

- Implementation of SCReAM in go (started here: <https://github.com/pion/interceptor/tree/feat/scream-cgo-update>)
- Evaluation of CC algorithms (and how different encoding/decoding parameters influence end user results)

Legal and Technical Requirements:

The source code produced in this project should be published under a copyleft license (ie GPL 3).

Focus will be on product readiness and stability. All code must adhere to common best practices and follow standardized language conventions.

Code checks, tests and build procedures should be automated (gitlab-ci) as much as possible and the outcome (build package) should directly be installable in the host OS (Raspberry Pi debian bulleseye).

The CS department at KaU will provide the car, raspberry pies, a public facing server for the webrtc parts and provide the repositories and help with setting up repositories.