DESIGN AND PERFORMANCE ANALYSIS OF PARALLEL PROCESSING OF SRTP PACKETS

Jan Wozniak
Master Degree Programme (2), FIT BUT
E-mail: xwozni00@stud.fit.vutbr.cz

Supervised by: Peter Jurnečka
E-mail: ijurnecka@fit.vutbr.cz

Abstract: Encryption of real-time multimedia data transferes is one of the tasks for telecommunication infrastructure which should be considered in order to reach essential level of security. Execution time of ciphering algorithm could play fundamental role in delay of the packets, therefore, it provides interesting challenge in terms of optimization methods. This work focuses on paralellization possibilities of processing SRTP for the purposes of private gateway with the usage of OpenCL framework, utilization gateway's resources and analysis of potential improvement.

Keywords: AES, SRTP, general-purpose GPU, OpenCL, parallel computations, gateway, VoIP.

1 INTRODUCTION

One of the essential metrics for measuring VoIP gateway's performance is the number of simultaneous calls. It is affected mostly by the computational demands of used communication protocols and number of registered users. While the count of registered users provides very limited room for improvement by the nature of the problem itself, there could be wide variety of aproaches in implementing the protocol stacks.

Significant amount of resources are utilized during indirect simultaneous call sessions by processing multimedia packets. Since security has recently grown to be necessary feature in VoIP communication, and the encryption and decryption processes are designed with the idea of optimization, it is primary scope of interest of this paper.

2 SRTP PROCESSING

Secure Real-time Transport Protocol was designed as an extension over RTP protocol to obtain security and confidentiality for multimedia sessions on application layer of ISO/OSI model. The packet has usual structure cosisting of the SRTP header, authentication extension and only encrypted data payload. The default, and as this paper was written, the only defined cipher is 128bit AES.

In VoIP communication the time has essential impact on the quality of transmitted information, therefore, it is important that ensuring the security of RTP wouldn't increase the latency over the acceptable level. Among typical limitations of real-time communications belong [2]:

- Maximal tolerable latency of round-trip time 300ms.
- Smaller packet loss than 5%.
- Sensitivity to factors that are difficult to objectively measure such as jitter.

The exact size of payload in SRTP packet can differ widely according to the used codec, its bit rate, and sampling frequency. The basic multimedia codec is G.711, which should be supported by every multimedia device and with standardly used 20ms sampling period, the length of payload is 160 bytes.

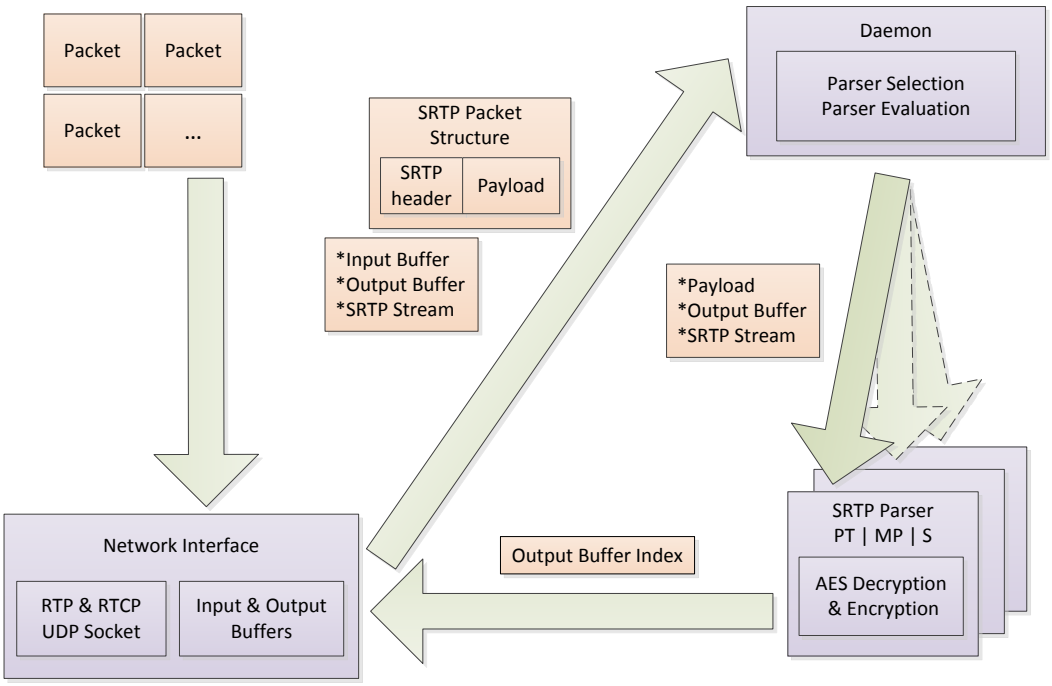

Figure 1: SRTP Processing Scheme: the data from network are captured in the *network interface* running in its own thread. With corresponding *stream*, input and output buffer are passed to the *daemon* which can select one of three types of *SRTP parser* according to the application configuration. Index to output buffer is passed back to the *network interface* and result is sent to the desired address.

Due to it's wide support it has been chosen and used for evaluation and comparison of two aproaches for encryption and decryption.

2.1 AES

The counter-mode for AES has been selected for the VoIP sessions due to its invariacy to delay and even possible loss of packets. This provides flexibility in SRTP processing for packets received out of order as well.

Each byte in the block of AES cipher can be computed separately. The local dependency of bytes is limited to the distinct steps of the algorithm which requires usage of barriers for local synchronization.

2.2 Persistent Thread

For massive parallel applications the obvious aproach would be to utilize as much of machine's power as possible to gain the largest speed-up in every single execution. However, the aim of this paper is to minimize large delays for multiple sessions which requires rather carefull allocation of resources. Persistent threads is special type of programing paradigm combining both, the possible gain of mapping the program for parallel computation and considerate usage of resources [1].

Since the initialization of computational kernel can consume significant amount of time compared to the actual execution, larger kernel reusing its resources for multiple similar computations could render the initialization negligible trading off portion of parallelization.

This approach has been chosen for packet parsing, while instead of mapping 160 OpenCL work-items on the entire G.711 packet's payload it uses 16 work-items in a loop that goes through the entire packet.

3 CONCLUSION

The commercial gateway with optimized hardware can hold around 120 concurrent calls[1]. The implementation proposed as backup for this paper evaluation can be sumarized in following graphs of distributed packet latencies. Measured was round-trip time latency of each packet during 50 to 150 concurrent calls that all lasted 20 seconds. The range of latencies seem to be unreliable however more than 95% of the recceived packets falls between the smaller range of the column which is visualized thicker.

The tests were all done on the machine with processor intel i5 2500k with HD3000 graphics chip running OpenSUSE 12.2 and OpenCL version 1.2.

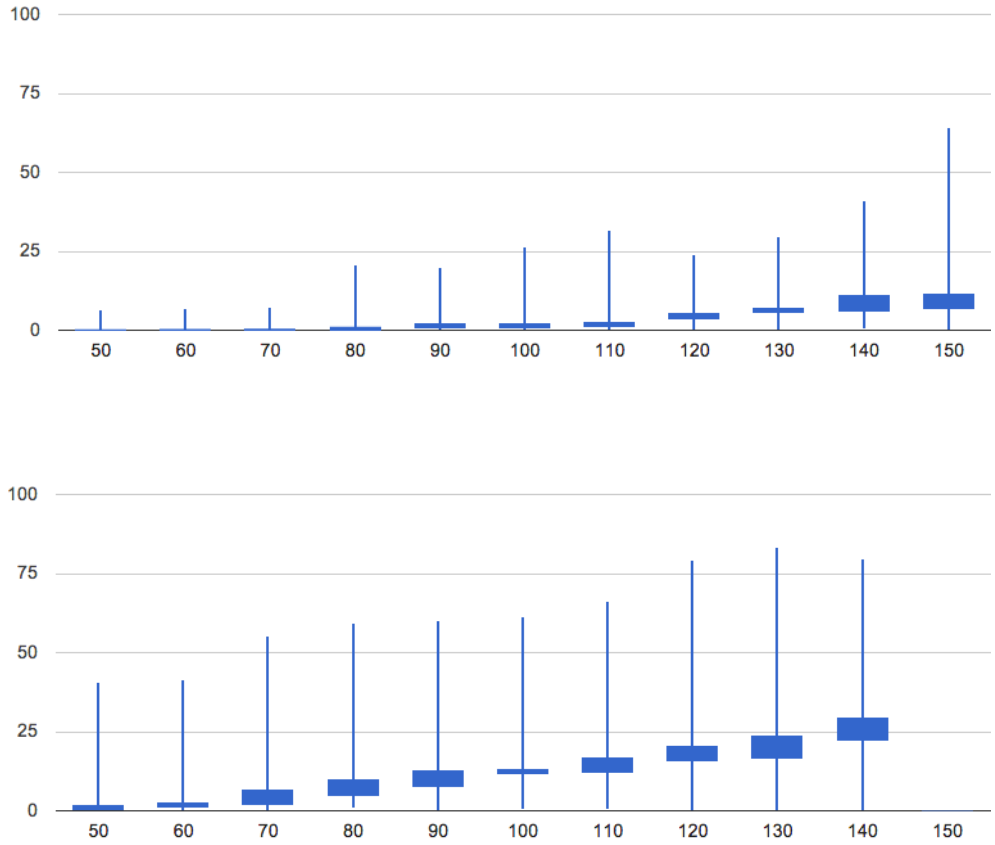

Figure 2: Top graph visualizes delay for prallel processing using OpenCL, bottom graph is for serial processing.

REFERENCES

[1] Kshitij Gupta, Jeff A. Stuart, and John D. Owens. A study of persistent threads style gpu programming for gpgpu workloads. In *Innovative Parallel Computing*, page 14, May 2012.

[2] C. Perkins. *RTP: Audio and Video for the Internet*. Addison-Wesley, June 2003.

[1] `http://www.athlsolutions.com/web/en/Products/tabid/128/ProdID/38/Hipath\_4000.aspx`