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 transfers 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 parallelization 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 and quality 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 approaches in implementing the protocol stacks.

Significant amount of resources are utilized during indirect simultaneous call sessions by processing multimedia packets [1]. 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.

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 300 ms.
- Smaller packet loss than 5%.
- Sensitivity to factors that are difficult to objectively measure such as jitter.

The designed application captures data from network in the *network layer* which ensures communication with both endpoints of multimedia session and is running in its own thread. It contains buffer pools for incoming and outgoing data to ensure maximal level of parallelization in each layer of application. Pointers for input and output buffers are passed for further packet processing where are extracted information such as header and payload from the packet, copied data from the memory

to OpenCL data structures and serial implementation of AES key schedule. Parallel processing of payload is executed in precisely 16 work-items while the application follows paradigm of persistent thread implementation.

In the scheme in figure 1 gray blocks visualize serial implementation in C/C++ and yellow blocks are parallel implementation in OpenCL. The arrows represent data flows between and inside of blocks. Thin purple lines depict usage of barriers.

Scheme 1 visualizes only packet encoding, however, decoding is executed in similar manner. 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. Due to it's wide support it has been chosen and used for evaluation and comparison of two approaches for encryption and decryption – serial and parallel.

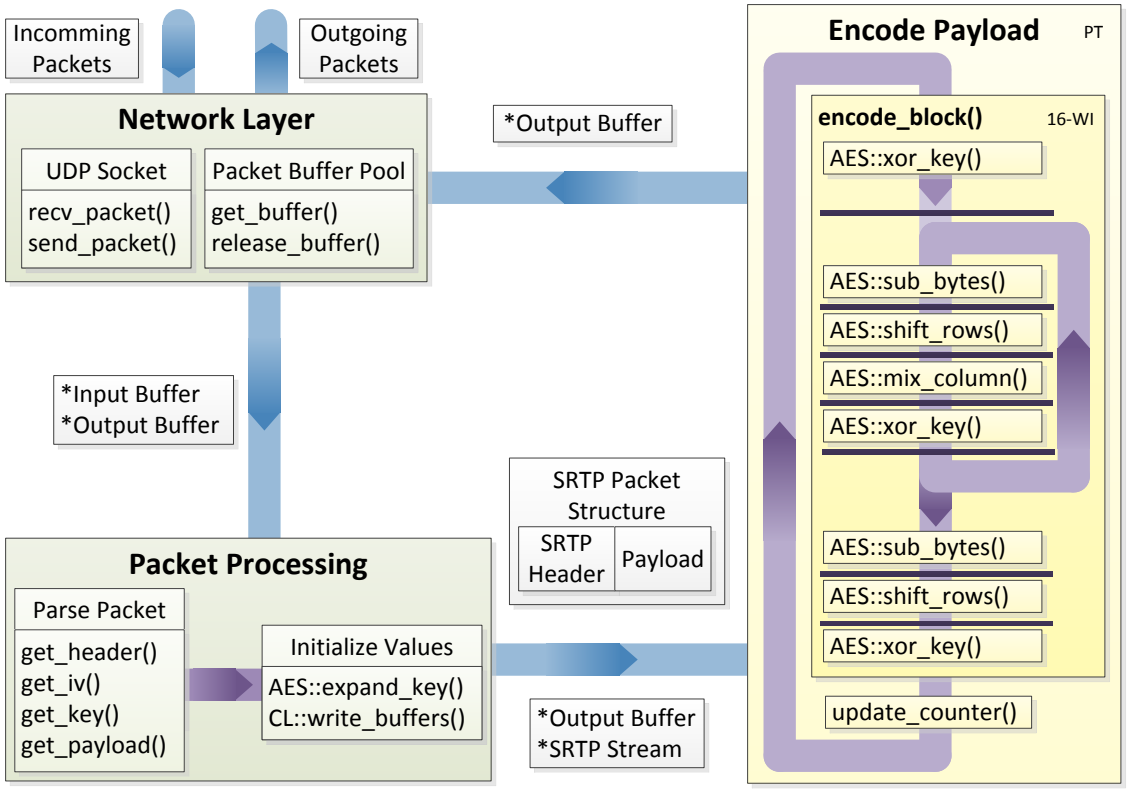

Figure 1: SRTP Parallel Processing Scheme.

2.1 AES

The CTR has been selected for the VoIP sessions due to its invariace for delay and even possible loss of packets and it provides flexibility in SRTP processing for packets received out of order. 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 approach 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 careful 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 [3].

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 G.711 packet's payload it uses one work-item for each AES block cell in a loop that goes through the data.

3 RESULTS

The commercial gateway with optimized hardware can hold around 120 concurrent calls [4]. The evaluation of implementation proposed as backup for this paper is summarized 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, results for 150 concurrent calls for serial implementation were excessively large thus not included in the graph. The thicker part of the column guarantees to contain 95% of the packets which was earlier presented as one of requirements for reliable session.

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.

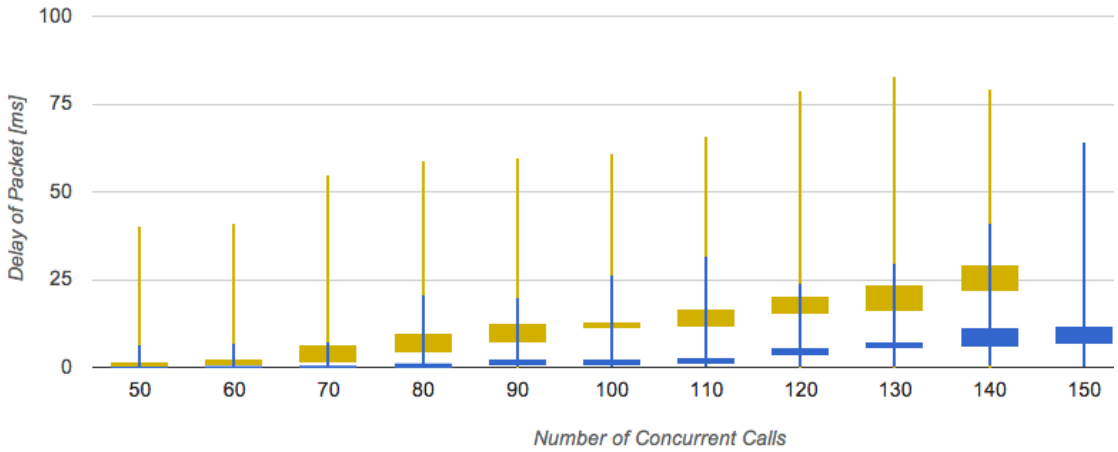

Figure 2: Latencies for parallel implementation in blue color and serial in yellow color.

4 CONCLUSION

Proposed parallel implementation can reduce packet latency caused by encryption to half or third depending on number of concurrent calls. Even though optimizations are far from being complete the results are promising and can bring opportunities for improvement in related tasks, e.g. transcoding.

REFERENCES

[1] A. Alexander, A. Wijesinha, and R. Karne, "An evaluation of secure real-time transport protocol (srtp) performance for voip," in *Network and System Security, 2009. NSS '09. Third International Conference on*, pp. 95 –101, October 2009.

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

[3] K. Gupta, J. A. Stuart, and J. D. Owens, "A study of persistent threads style gpu programming for gpgpu workloads," in *Innovative Parallel Computing*, p. 14, May 2012.

[4] "Siemens Hipath 4000 [online]." `http://www.athlsolutions.com/web/en/Products/tabid/128/ProdID/38/Hipath_4000.aspx`. Accessed 2013-3-3.