Ultra-Low Latency on Ethernet Technology

Atilla Alpay Nalcaci, Florian Wiedner*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: atilla.nalcaci@tum.de, wiedner@net.in.tum.de

***Abstract*—Network latency depicts the total amount of time for a data packet to be captured, processed and transmitted, potentially through multiple devices, from one communication endpoint to another. This measurement of delay is a performance characteristic among telecommunications and cellular communication providers.**

In this paper, we present our research on the implementation requirements of Ultra-Reliable Low-Latency Communication (URLLC) to the current ethernet infrastructure. Further, we analyze commodity software and hardware on the performance of low latency packet processing. Investigations focus on network areas and quality of service provisions and conclude on requisites to support URLLC applications in shared networks. Findings show that any non-specialized network infrastructure requires fine-tuning of communication specifications that is capable of achieving maximum transmission delay of approximately 50 ms with very high achievable network reliability and utilization measurement.

***Index Terms*—5G, ultra-reliable low-latency communication, network latency, packet processing, reliability**

1. Introduction

The latency of a network describes the overall delay in the communication, usually measured in ms (millisecond) and the final result is typically indicated as a round trip delay – the absolute amount of time that is spent for transmitting the information to the target destination and then back to the original sender. It is important to appertain performance optimizations concerning the latency to test system performance emulating under high latency in order to optimize for users with lousy connections.

Ultra-low latency is a service category introduced in 5G New Radio (NR) standard which allows newly emerging services and applications to surpass and resolve the prospective latency and reliability requirements. 5G NR is the global standard for a robust and capable cellular network infrastructure that enables enhanced communication between user endpoints in terms of data delivery, reliability, and transcend user experience on a massive scale [1]. In summary, 5G networks encapsulate the following generic connectivity types: enhanced Mobile Broadband (eMBB), massive Machine-type Communication (mMTC), and Ultra-Reliable Low-Latency Communication (URLLC) [2].

The conception of 5G networks is inclined to interweave with the notion of "ultra-reliable" connectivity, making the implementation process of URLLC rather difficult and restrictive [3]. The trend of ultra-reliable communication guarantees perpetual connectivity of approximately > 99.999% for a given time window [4]. URLLC enables computer networks to process and exchange high volume data packets with eminently low latency between the endpoints. These networks support real-time access and request/response to previse rapidly changing data [2].

The key feature of URLLC is low latency. This is a crucial aspect for devices and/or gadgets which perform over a common network of command nodes that provide query of commands on what needs to be executed next [2]. Performance measurements that are included in the following sections are conducted in the context of tail latency-percentage of response times out of all responses to the input and output requests that the system serves, which take the longest amount of time in comparison with the totality of its response times. With low tail latency, networks are open to optimizations that enables the processing of large amounts of data with minimal latency. Since networks are required to be adaptive to dynamic data entries and alterations, these optimizations have the potential to increase the overall network utilization as well as inaugurate an expeditious method of data transfer.

In this paper, we present our research and analysis for the requirements of URLLC to the current ethernet technology. Further, we analyze what is needed to support URLLC applications in shared networks. The paper is organized as follows. Section 2 represents some background and related work. Section 3 examines the current status and evolution of the ethernet technology. Section 4 gives a brief description and potential sources of tail latency. Section 5 presents thorough information on the prerequisites of URLLC network infrastructure. Finally, Section 6 concludes the paper and provides some literature on various future work.

2. Background and Related Work

In this section, we present the work on supporting URLLC on non-specialized networks and latency measurement methodologies for low-latency systems.

2.1. Latency Measurement

Several studies exist [5]–[7] for achieving highly reliable connectivity with low-latency measurements. The research carried out by Gallenmuller et al. [8] depicts a new methodology for measuring the tail latency of Linux

doi: 10.2313/NET-2022-07-1_12

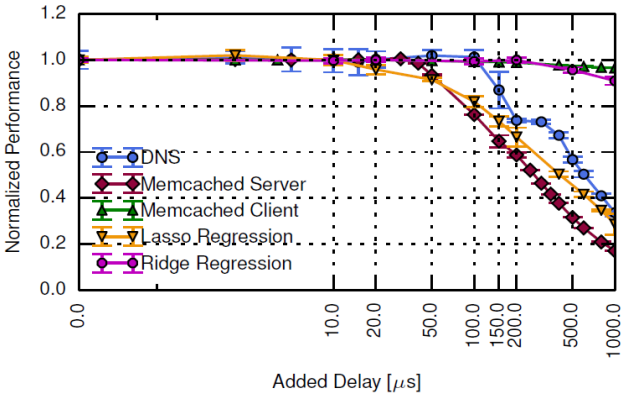

Figure 1: The effect of static latency on different applications. [9]

supported off-the-shelf hardware commodities. Furthermore, the research presents a software stack that lowers the overall tail latency of packet processing applications. Latency measurements are made through hardware timestamping for increased precision. The software stack that is presented as a solution has attested to the occurrences of low-latency packet processing on a consistent demeanor. Ensuing case study proved to achieve a forwarding latency of below 25 µs for a non-overloaded Snort IPS.

2.2. Operating System and Hardware

Identifying the software- and hardware-related latency and jitter is one challenge of ensuring low latency while keeping the connectivity uninterrupted, i.e. reliability of the connection. As Stylianopoulos et al. [4] examine, the main objective is to prevent the network interruptions that are directly influential over the user-space applications which are responsible for handling the packet processing flow in its service, to the furthest extent possible. To achieve this, certain kernel options are introduced and later delineated to have contributions to lower and stable network latency. Examples include preparatory configurations of system-level setup options that are namely; Thread isolation which isolates the Data Plane Development Kit (DPDK) cores to prevent common use of these cores by other tasks, disabling of interrupt balancing for disabling the dynamic interrupt distribution daemon to avoid unrelated DPDK interrupts, and disabling Intel turbo-boost technology which introduces high variation to packet processing latency.

2.3. Cloud-based Applications

Cloud-based applications are described as the software which the analogous users access through a shared network, commonly being the internet. The research carried out by Popescu et al. [9] focuses on characterizing the latency of the cloud-based applications' performance. Applications that are used during this research are Domain Name System (DNS), Memcached, STRADS-a scheduled model parallelism distribution framework, and Apache Spark. The methodology is based on devising the host to experience different network latency values by modifying the link that connects the Top of Rack switch (ToR) to the corresponding host. Figure 1 gives an overview of the mentioned applications that are experimented in terms of their additive latency – x-axis is the static latency added in microseconds for round-trip time (RTT) and y-axis is the normalized performance. In particular, the baseline performance of the individual applications is analogized with the ratio of the measured performance at each latency point to analyse the effect of static latency [9].

Experimental conclusions suggest that different injections of controlled network latency have varying impacts on different applications. In particular, latency values are affected to differing amounts, such that even small network delays are found to be influential upon divergent application performance, nearly tens of microseconds.

TABLE 1: Throughput and latency in 1G to 5G [11]

Generation	Data Rate/Throughput (Maximum)	Latency (Minimum)
1G	9.6 kbit s^{-1}	$>$ 1000 ms
2G	2 Mbit s^{-1}	600−750 ms
3G	100−300 Mbit s^{-1} (DL), 50−75 Mbit s^{-1} (UL)	$<$ 10 ms (UP), $<$ 100 ms (CP) (typical values: 40−50 ms)
4G	1−3 Gbit s^{-1} (DL), 0.5−1.5 Gbit s^{-1} (UL)	$\sim$ 5 ms (UP), $<$ 100 ms (CP) (typical values: 40−50 ms)
5G	1 Tbit s^{-1} (over 100 m) $>$ 20 Gbit s^{-1} (DL), $>$ 10 Gbit s^{-1} (UL)	$\leq$ 1 ms

3. Current Status of Ethernet Technology

Presently, ethernet is the most widely used commodity network system that allows the implementation of wired computer networking technologies, most of which are commonly being used in local area networks (LANs) and wide area networks (WANs). Capabilities of the modern ethernet technology allow expeditious data transfer and hard real-time communication. Ethernet is readily scalable, thereby enabling thriving technologies to be easily integrated. Subsequently, as Loeser and Haertig [10] points out, the current ethernet infrastructure is increasingly moving towards the switches–network connection devices that manage the data flow in a given network by transmitting data packets between corresponding hosts. In the context of media-access control, modern ethernet technology uses Carrier-sense multiple access with collision detection (CSMA/CD) to defer data transmissions until the predefined communication channel is not occupied by any transmission. The aforementioned shift to network switches allow the use of traffic shaping strategies by means of implementing the hard real-time distributed systems on commodity networks. Nevertheless, current intuition regarding the collision avoidance limitations yields an increase in terms of processing load and bandwidth allocation over a common network.

The evolution of network systems and their specifications has been comprehensive apropos the changing network architectures and radio access network (RAN)

systems [12]. Throughout different generations of network evolution, two principal parameters exist that are rudimentary, namely throughput and latency. While latency signifies the amount of time for data to travel from one communication endpoint to another, throughput denotes the amount of data that has moved successfully between the predetermined hosts. With the drastic evolution of communication technologies, significant architectural changes eventuated, introducing seamless connectivity and mobility properties. In summary, Table 1 shows the values of different generations concerning the conjectural throughput and latency evaluations [11].

Latest generation systems are intended to achieve efficient system development and utilization, in addition to preserving end-to-end connection requirements. Nonetheless, the scope of this research does not cover deployment and optimization fields, as the main focus is the application of URLLC on non-specialized networks.

4. Sources of Tail Latency

The presented rationale is gathered from multiple tuning guides, while also remarking the presence of various studies that aim to overcome ambiguous extents that are arisen from the complication of tail latency.

As already outlined, tail latency, commonly referred to as high-percentile latency, is the percentage of response times from which the response is received that takes the longest amount of time in contrast to the overall response times of the specified server. Maintaining a low margin for tail latency is tricky, especially for large-scale applications that consist of interactive operations. Tail latency is considered to be problematic due to numerous reasons. As outlined by Haque et al. [13], applications with interactive foundations contend in terms of providing complex user functionalities under strict latency constraints. As a result, this creates an unavoidable setting in which tail latency having an impact over user requests pursuant to high degrees of parallelism—a performance metric that indicates the number of operations that can be executed on a server concurrently [13]. Since the totality of a request is not finalized until the slowest sub-request is finished, tail latency is proved to be an arduous challenge for developers.

While tail latency might be an outcome of an application-specific service, there are numerous reasons where tail latency can be introduced to a network, some examples being hardware peripherals, operating system kernel modes or application-level configuration preferences. For instance, as Li et al. [14] points out, buffering has an immense impact on networks that have low traffic rates. This is considered as a primary predicament since URLLC applications generally possess low traffic rates, interpreting an operose situation since such conditions is critical.

Nonetheless, there are numerous studies [10], [13], [15] that explore and mitigate the problem of high latency through modifying certain kernel operations, using various software development tools to enable ISP and P2P user cooperation and implementation of traffic shaping on switched Ethernet.

5. Network Communication Requirements of URLLC

Following explorations and analysis are gathered from various articles that focus on network latency characterization and URLLC performance on commodity hardware. Imperative network specifications and requirements are listed, and use cases are denoted accordingly.

The main purpose of URLLC is to resolve newly emerging latency-critical applications by means of handling the prospective latency and reliability requirements. In principle, network systems that support URLLC applications are capable of supporting real-time access to rapidly changing data by design, thereby allowing the network to be optimal and available to network optimizations in the context of processing high volume data packets with eminently low latency [4]. While the benefits of URLLC on a network are authenticated, particular communication requirements must be established to the network before enabling URLLC supported applications. Preliminary requirements of URLLC services that are prospective to the network infrastructure which the URLLC will be deployed are as follows:

a Low latency: The approximated maximum end-to-end latency requirements for a network with URLLC adaptation, ranging from 1 ms to 50 ms. On average, the conception of URLLC requisites presented by 3rd Generation Partnership Project (3GPP) organizations is an average user-plane radio latency of 0.5 ms, comprising uplink and downlink together. Note that these values are not bounded by an associated reliability value [4].

b High reliability: As stated in section 1, trend of URLLC guarantees a perpetual connectivity ratio of nearly $> 99.999\%$ reliability. As reported by Stylianopoulos et al. [4], URLLC use cases stipulate a reliability measure, ranging from 99.9% to 99.999% of reliability. Note that the depictions are based on a network latency of 1 ms for a transmission of packet size 32 bytes. Thence, the network infrastructure must be capable of sustaining a highly reliable packet delivery margin.

c Low jitter: In spite of the general network specifications, i.e. a network infrastructure which is verified to maintain a latency extremity that is in the acceptable bounds of a system, a certain deviation from the true periodicity of a network is prospectively contingent [4]. This deviation is commonly referred to as "jitter" which describes the variance in latency. High values of jitter connote inadequate network performance and introduces packet loss to the network flow. In particular, communications services that are based on URLLC service category require the average jitter to be $< 50\%$ cycle time [16].

Additionally, low traffic rates are also another aspect that is considered essential to the notion of URLLC services. However, event-based applications which augmented to function in a dynamic environment are not

TABLE 2: Example of low latency and high reliability use cases and their requirements [1]

Scenario	End-to-end latency	Reliability
Discrete automation—motion control	1 ms	99,9999%
Electricity distribution—high voltage	5 ms	99,9999%
Remote control	5 ms	99,999%
Discrete automation	10 ms	99,99%
Intelligent transport systems—infrastructure backhaul	10 ms	99,9999%
Process automation—remote control	50 ms	99,9999%
Process automation—monitoring	50 ms	99,9%
Electricity distribution—medium voltage	25 ms	99,9%

both latency and throughput critical. Standard use cases of these applications foster an approximate broadband speed of $< 50\,\mathrm{Mbit\,s^{-1}}$, a comparatively low traffic rate as contrasted with modern networks [1].

An example of URLLC use cases and requirements [1] are depicted in Table 2. Examples are made with respect to predefined industrial applications that benefit from the utilization of a URLLC network infrastructure. As formerly indicated, end-to-end latency values are in the range of 1 ms to 50 ms, in addition to the eminent reliability percentages. Per contra, maintaining the scope on network design and overall system performance.

6. Conclusion and Future Work

Ultra-reliable and low-latency communication is a substantial service category for providing reliable connection segments to applications that retain stringent latency and reliability measures. The main features of URLLC, low latency and high reliability in particular, enables a primary usage scenario for 5G network infrastructure. In order to sustain a network system that supports URLLC applications, the analogous network infrastructure must be fine-tuned in terms of sustaining high reliability for the correspondent application channels and a latency measurement of 50 ms extremity. Furthermore, certain studies exist for enabling and testing URLLC on Wireless Access systems and Cloud-based application data centers.

In particular, Popovski [17] provides a framework that can be utilized for scheming ultra-reliable wireless network systems, and analyzing accordingly. Previous work of the same research depicts the building blocks for the appliance of URLLC in wireless network access. Continually, the following research is aimed to provide further information on techniques and principles for URLLC wireless access. The research further annexes investigations by introducing a detailed discussion on communication-theoretic principles of URLLC. Subsequently, concepts of latency and reliability are expressed as coupled, from the perspective of an application that has a predefined latency constraint. At length, reliability of a communication is defined under the probability that the measured latency does not exceed this predefined latency constraint.

Additionally, Popescu et al. [9] present quantitative results regarding test benchmarks of cloud-based applications. Results suggest that applications that are of different complexity and distance to the corresponding data centers are affected by network latency to differing amounts. Findings are auspicious with respect to sustaining a cloud-based ultra-low latency network environment for the upcoming future studies that are fundamental to this area.

References

[1] Z. Li, H. Shariatmadari, B. Singh, and M. A. Uusitalo, "5G URLLC: Design Challenges and System Concepts," pp. 1–5, 2018, accessed: 2021-12-08. [Online]. Available: http://dx.doi.org/10.1109/ISWCS.2018.8491078

[2] P. Popovski, C. Stefanovic, J. J. Nielsen, E. de Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A. S. Bana, "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," vol. 67, no. 8, pp. 5783–5786, 2019, accessed: 2021-11-18. [Online]. Available: http://dx.doi.org/10.1109/TCOMM.2019.2914652

[3] P. Popovski, J. J. Nielsen, C. Stefanovic, E. de Carvalho, E. Strom, K. F. Trillingsgaard, A. S. Bana, R. Kim, D. M. Kotaba, J. Park, and R. B. Sorensen, "Ultra-Reliable Low-Latency Communication (URLLC): Principles and Building Blocks," pp. 1–7, 2017, accessed: 2021-12-05. [Online]. Available: http://dx.doi.org/10.1109/MNET.2018.1700258

[4] C. Stylianopoulos, M. Almgren, O. Landsiedel, M. Papatriantafilou, T. Neish, L. Gillander, B. Johansson, and S. Bonnier, "Industry Paper: On the Performance of Commodity Hardware for Low Latency and Low Jitter Packet Processing," pp. 1–5, 2020, accessed: 2021-11-18. [Online]. Available: http://doi.org/10.1145/3401025.3403591

[5] AMD, "Performance Tuning Guidelines for Low Latency Response on AMD EPYC 7001-Based Servers - Application Note," 2018, accessed: 2021-11-12. [Online]. Available: http://developer.amd.com/wpcontent/resources/56263-Performance-Tuning-Guidelines-PUB.pdf

[6] J. Mario and J. Eder, "Low Latency Performance Tuning for Red Hat Enterprise Linux 7," 2017, accessed: 2021-11-12. [Online]. Available: https://access.redhat.com/sites/default/files/attachments/201501-perf-brief-low-latency-tuning-rhel7-v2.1.pdf

[7] E. Rigtorp, "Low latency tuning guide," 2021, accessed: 2021-11-12. [Online]. Available: http://rigtorp.se/low-latency-guide/

[8] S. Gallenmüller, F. Wiedner, J. Naab, and G. Carle, "Ducked Tails: Trimming the Tail Latency of(f) Packet Processing Systems," pp. 1–7, Oct. 29, 2021, accessed: 2021-11-12. [Online]. Available: http://dx.doi.org/10.23919/CNSM52442.2021.9615532

[9] D. A. Popescu, N. Zilberman, and A. W. Moore, "Characterizing the impact of network latency on cloud-based applications' performance," vol. 2, no. 914, pp. 3–16, Nov. 2017, accessed: 2021-11-12. [Online]. Available: http://dx.doi.org/10.17863/CAM.17588

[10] J. Loeser and H. Haertig, "Low-latency Hard Real-Time Communication over Switched Ethernet," pp. 1–3, 2004, accessed: 2021-11-12. [Online]. Available: http://dx.doi.org/10.1109/EMRTS.2004.1310992

[11] A. Slalmi, H. Chaibi, A. Chehri, R. Saadane, G. Jeon, and N. Hakem, "On the Ultra-Reliable and Low-Latency Communications for Tactile Internet in 5G Era," vol. 176, pp. 3853–3862, 2020, accessed: 2021-12-08. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050920318925

[12] O. T. Eluwole, N. Udoh, M. Ojo, C. Okoro, and A. J. Akinyoade, "From 1G to 5G, What Next?" vol. 45, Aug. 2018, accessed: 2021-12-10. [Online]. Available: http://www.iaeng.org/IJCS/issues_v45/issue_3/IJCS_45_3_06.pdf

[13] M. E. Haque, S. Elnikety, Y. h. Eom, R. Bianchini, Y. He, and K. S. McKinley, "Few-to-Many: Incremental Parallelism for Reducing Tail Latency in Interactive Services," pp. 1–4, 2015, accessed: 2021-12-05. [Online]. Available: http://dx.doi.org/10.1145/2694344.2694384

[14] J. Li, N. K. Sharma, D. R. K. Ports, and S. D. Gribble, "Tales of the Tail: Hardware, OS, and Application-level Sources of Tail Latency," pp. 1–10, 2015, accessed: 2021-12-05. [Online]. Available: http://dx.doi.org/10.1145/2670979.2670988

[15] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPs and P2P Users Cooperate for Improved Performance?" vol. 37, no. 3, pp. 31–34, 2007, accessed: 2021-11-12. [Online]. Available: http://dx.doi.org/10.1145/1273445.1273449

[16] L. Xia, X. Hou, G. Li, Q. Li, L. Sun, W. Rui, J. Erfanian, S. Tatesh, B. Liu, A. Chan, B. Tossou, A. G. Serrano, B. Sayrac, G. Wannemacher, A. Kadelka, A. Frisch, J. Sachs, D. Patel, and R. Sabella, "5G E2E Technology to Support Verticals URLLC Requirements," Nov. 18, 2019, accessed: 2021-12-14.

[17] P. Popovski, "Ultra-Reliable Communication in 5G Wireless Systems," pp. 1–4, 2014, accessed: 2021-11-20. [Online]. Available: http://dx.doi.org/10.4108/icst.5gu.2014.258154