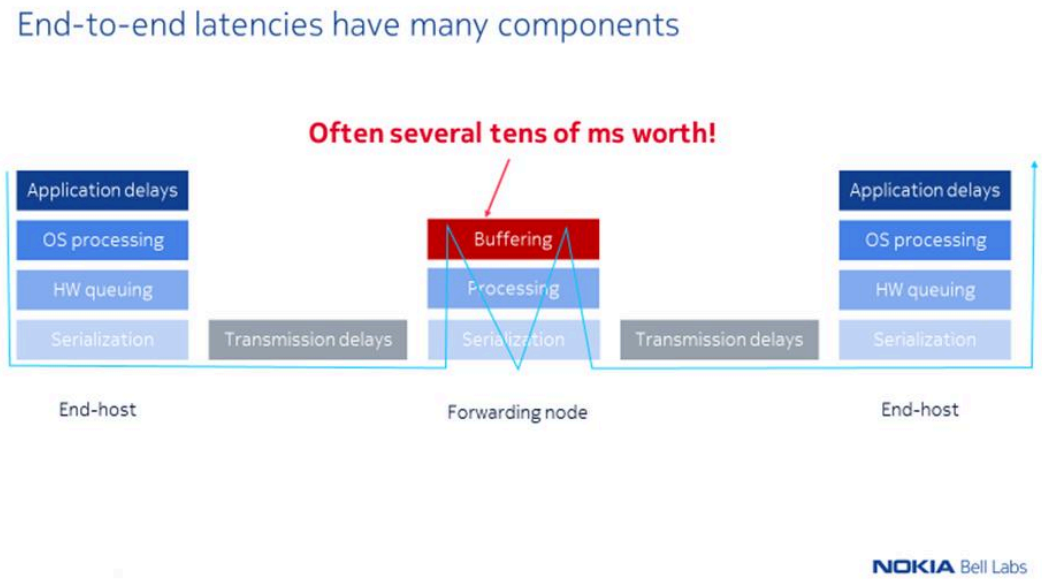
Focusing on latency, not throughput, to provide a better internet experience and network quality

Why is it that, while broadband access speeds have now reached multi-gigabit, we are still having video conferencing issues, VoIP hiccups, gaming lag spikes, and spend so much time staring at a loading icon on our devices?

If COVID-19 has shown us anything it is the importance of universal broadband access. But raw broadband speeds are just not enough to improve the quality of service experience: we need new ways to optimize, monitor and manage networks and applications to deliver optimum latency. Upgrading a broadband connection from 50 Mb/s to 500 Mb/s isn't going to resolve latency woes, so what will? Is the continuous race for more broadband bandwidth still the critical factor? Or should it make way for another metric that is just as important: latency.

In computing terms latency is defined as "the delay before a transfer of data begins following an instruction for its transfer" or, alternatively, the time it takes for a packet to travel from its source to its destination and back again. In reality, what this means for an end-user is the delay between an action (push of a button, etc.), and getting a response to that action.

Over the last 20 or 30 years, we've seen a dizzying rate of evolution of the various communication technologies and protocols in our networks. Unfortunately, although latency requirements were always a factor, they have not kept pace with the rapidly changing nature of real-time communications of internet users. Every new generation of communication technology has brought forth improvements in latency, but always from a siloed (e.g., Wi-Fi, 3G/4G/5G, DOCSIS, FTTH, etc.) point-of-view, with little consideration for end-to-end latency measurements or management. You can't easily fix low latency consistency with just better gaming netcode, or better peering and edge compute, or more efficient video codecs—you need an end-to-end latency strategy and cooperative ecosystem. Data packets must traverse a variety of networks, each of them with a different transmission technology (scheduled transmission, framing, random-access, etc.) and each with different congestion control mechanisms and algorithms (or sometimes none at all).



We have seen pockets of much needed improvement in a variety of domains from a multitude of industry players (IETF, Google, CableLabs, Microsoft, Apple, to name a few).

- Classic TCP: Reno, DCTCP, BBR, L4S, etc.
- Queuing algorithms: RED/WRED, CODEL, PIE, PI2, DualQ, etc
- Wi-Fi: Wi-Fi 6 OFDMA, Wi-Fi 6E 6Ghz clean spectrum, hardware acceleration for packet processing offload, etc.

Yet end-users are still not truly reaping the promised benefits. Why is that, exactly?

Let's take a look at a recently popular and pertinent use case: cloud gaming. I say pertinent because cloud gaming is unique in that it is an application that requires both a high-speed broadband connection and relatively low and consistent latency. Unlike high-bitrate video streaming (4K VOD), which can tolerate relatively high and inconsistent latency due to the deep buffering nature of the streaming protocols and the pervasive use of CDNs, cloud gaming operates on a fixed latency budget that can't easily cope with latency spikes or changes. The infrastructure requirements are significant in terms of edge localized datacenters, server and GPU farms, all for the purpose of providing the lowest latency possible. To better visualize the intricacies of this model, let's consider the following illustration.

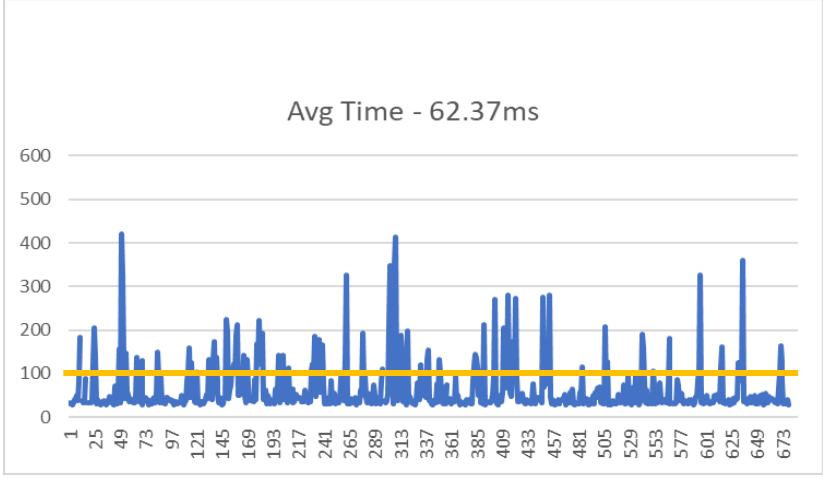
	Median Latency	Jitter Potential (99th percentile)
In-Home Ethernet	1-2 ms	<1 ms
In-Home Wi-Fi (sub 100 Mb/s traffic load, without AQM)	~10-20 ms	~10-4000 ms
xDSL	~20-50 ms	~5 ms
DOCSIS (3.0 under load, with buffer control)	~10 ms	~10-200 ms
LLD DOCSIS (estimated)	~1 ms	~1-5 ms
Access - FTTH	<1 ms	~1-2 ms
4G/5G (RAN latency, non-URLLC based profile, QCI 9)	~15-35 ms	~10-500 ms
IP Transport & Peering	~40 ms (near-edge) ~225 ms (intercontinental)	~5-10 ms ~10-50 ms
Applications & platform	~15-40 ms	~10-500 ms

What's important to remember for these latency sensitive non-queue building applications and services isn't just your average or median latency, but what is the risk for the consistency (or jitter) at the 95th or even 99th percentile. It's really of little value if a technology can deliver ultra-low latency while being easily susceptible to significant variance and spikes. I'd even go as far as saying that the ultimate goal we should strive for isn't ultra-low latency at all; it should be consistent low latency.

Video conferencing can operate very well at one-way latencies of ~150 ms. I wouldn't call 150 ms exactly low latency, but as long as it can remain consistent then the user experience will remain optimal. Many of the real world issues experienced by end-users are from the fact that many equipment and device manufacturers have focused so much on maximum throughputs and ultra-low latency claims, and that not enough attention has been

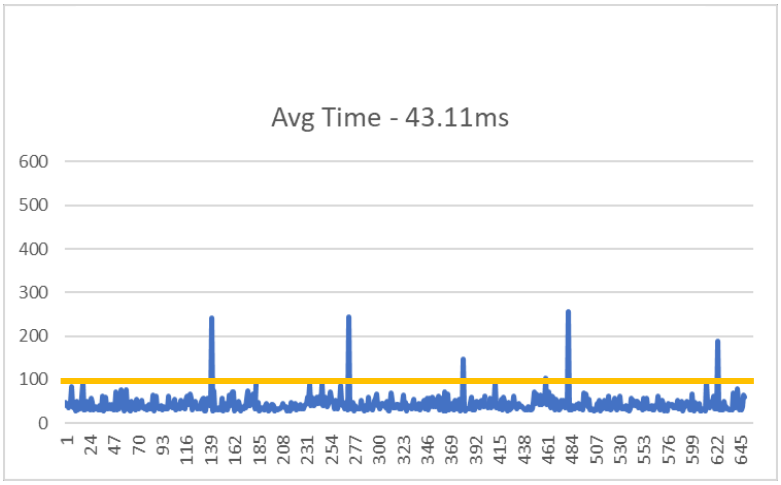
put into the hardening of such solutions when operating in environments that have outside factors affecting the performance of this latency (Wi-Fi interference being a prime example).

Let's consider another example in this quest for consistent latency. A few months ago, I was approached by a major internet provider to understand why they had challenges in providing a good cloud gaming experience, considering they claimed to operate one of the best and fastest fiber-to-the-home services in North America. My team really likes to focus on real world analysis, how things actually work from the point of view of a typical end-user rather than how technology behaves in a controlled lab environment. Lab testing has significant merit, value and need, but can fall short when trying to replicate the idiosyncrasies of the real world. So, we went about doing our testing in an end-user's residence, subscribed to this service provider's broadband service, experienced various sources of interference, traffic patterns, and so forth. The household was served by 1 Gigabit symmetrical fiber-to-the-home, with lots of video streaming, Facetiming and other internet traffic activity going on, while we played and measured latency of an immensely popular first-person shooter game (a 60 player battle royale) using a world-leading cloud gaming provider. We used a standard residential gateway CPE with Wi-Fi 6 capabilities, as they very rarely offer any advanced form of latency queue management beyond what the system on a chip (SOC) vendor offers. You can see the results in the first graph below (*Y axis represents latency in ms, while X axis represents time in seconds*) :



We see that for that our 10-minute gaming session had an average latency (from the gaming computer to the cloud gaming server) of ~62 ms. At face value, 62 ms would indicate a great experience, considering that cloud-gaming latency budgets are about 100 ms, on average. But when you look at the distribution of the latency measurements, we can clearly see significant spikes due to a variety of factors such as Wi-Fi congestion, queuing delays, etc. Each time our latency spiked over the 100 ms threshold, it forced the cloud gaming provider to adjust the quality and bitrate of the video stream to try and compensate. Yet we had only been using ~100 Mb/s of bandwidth, on a 1 Gb/s connection. Sure, throttling down the bitrate did provide a reprieve, but we can see that the root cause of the latency issue wasn't network congestion: having a Gigabit internet connection wasn't going to provide a better experience than someone with 1/10th the broadband capacity.

So what happened when we introduced a next-generation queuing algorithm (in this case the Bell Labs developed PI2), one designed for use cases such as this? The second graph below shows the results for the same CPE device, under the same network load, in the same time period and network conditions. You are looking at a significant ~30% improvement in the average latency. But more importantly, the latency is now very consistent with very little jitter above 100 ms . The spikes seen are caused by the Wi-Fi scanning process taking one of the dedicated antennas offline (*Y axis represents latency in ms, while X axis represents time in seconds*).



Industry interest and focus on latency improvements has grown steadily over the last few years, yet we are still held back with different schools of thought on how to address the old "bufferbloat" problem. It's not that there is a lack of potentially viable solutions—on the contrary—but what is holding the industry back is the will and fortitude to look at the problem as an end-to-end quality of service issue. Queuing algorithms alone aren't going to fix this. Neither are IP/TCP stack enhancements, nor the natural evolution of communication standards and protocols.

- Are SOC vendors open to providing the necessary low-level packet processing access?
- Are webscale platform vendors willing to jointly adopt technology and protocols that can be used and accessed by application builders?
- Are equipment vendors ready to invest in the necessary R&D to provide latency improvements across all forms of access technology (Wi-Fi, DOCSIS, FTTH, 5G, etc.)?
- Are application providers (gaming companies, video conferencing services, etc.) willing to support industry- and standards-based latency improvement protocols and technologies?

I'd like to think so. There is too much at stake to try and ignore this or go at it alone.

There are already excellent technology candidates such as L4S with (an admittedly small number of) real world results that can easily be implemented and replicated. Technology that can span the various networks, that provide application awareness, and more importantly some form of control and management. It's time to move away from simple ping tests, or speed tests as a benchmark for good quality of service experience. They may tell you if you have enough bandwidth to play a given game, but in no way can they tell you what the quality of that gaming experience will be.

Which brings us to my final discussion point: how does one go about properly measuring latency? The answer most often depends on which part of the ecosystem you fall into. Measuring IP packet latency on a given network technology (e.g., Ethernet or Wi-Fi) is well established, with a rich and active ecosystem of service assurance vendors ready to "test" your latency. Application providers will provide ping or round-trip time measurements for your viewing pleasure, but without any way to tell where latency spikes or issues may be coming from.

I think you see what I'm getting at: we need technology that application and network providers can jointly use to not only measure and monitor, but also manage and configure latency. Proper congestion feedback mechanisms that both application and network nodes can share and understand. Proper Advanced Queuing Management (AQM) protocols that can also measure and monitor, provide manageability and configurability, and expose their real-time metrics to those same applications. There are trade-offs when it comes to offering the best latency

and/or the best throughput, and the means to manage and tune the end-to-end platform for given use cases is what we should strive for.

So many technologies and concepts like digital twins, true real-time high-definition AR/VR, level 5 autonomous driving, will be stymied if the industry can't rally together to solve this. For me personally, I'd gladly take a 1-point increase in Kill/Death ratio on my favorite first-person shooter with better latency consistency. For others, it may mean being able to do a 1-hour video conference session over Wi-Fi without any form of disruption, or maybe not having to look at spinning icons while we wait for every little request or action we make over the internet. We have hundreds if not thousands of incredibly smart and talented engineers working on these issues and concepts, yet when I ask them how this will improve my FPS, or K/D ratio, all I get back are blank stares.

So, I leave you with this: who else is willing to see past their own technology domain to try and address this, to really make a difference in the quality of service experience of end-users, since I hope we can all agree that throwing more bandwidth at it isn't working anymore, and domain level solutions are not the answer either.

Regards,

Gino Dion (gino.dion@nokia.com)

Koen De Schepper (koen.de_schepper@nokia-bell-labs.com)

Olivier Tilmans (olivier.tilmans@nokia-bell-labs.com)