



An Analysis of Network KPI variability for various SaaS application in the Internet

JP Vasseur (jpv@cisco.com), Ph.D., Cisco Fellow, Mukund Yelahanka Raghuprasad (myelahan@cisco.com), Vinay Kolar (vinkolar@cisco.com), Ph.D.

October 2021

Up to just a few years ago, applications were located in private data centers, network topologies were fairly static and traffic growth was fairly predictable. Now, networks and associated challenges have rapidly evolved. SaaS-based applications are now used extensively and such applications are “in the cloud” and also move over time in a very dynamic Internet, making the network an even more crucial component of the application experience. Assessing the true user (application) experience remains a strong challenge. Current approach relies on the use of static templates that specify hard bounds for network Key Performance Indicator (KPI) metrics such as the delay, loss and jitter to be satisfied in order to meet the Service Level Agreement (SLA) and make the user “happy”. Other approaches consisting of receiving feedback from the application itself are being envisaged in the context of the “Predictive Internet”. The aim of this paper is to provide an analysis on the characteristics of such network KPIs for a number of SaaS applications.

How can we define the application experience?

Simply put, application experience is synonymous to user satisfaction.

As briefly discussed, the domain of application (user experience) has been studied through the lens of Service Level Optimization or SLO metrics (e.g., delay, loss, jitter) required to meet the application Service Level Agreement (SLA). More specifically, bounds have been specified that should not be crossed to meet the application SLA. Determining such hard bounds for network metric KPI using so-called templates is challenging and subject matter experts often disagree. Furthermore, such templates must be specified for each application.

Methods and granularity for probe measurements become highly critical and are very often overlooked. Indeed, agreeing that that maximum packet loss ratio for a Voice CODEC should not exceed a threshold T is meaningless without specifying how such a variable (packet loss) is computed. Often, systems do use probes sent every x seconds and then report an average value using sliding windows. Others may report a maximum value of a percentile but the granularity of such probes has a great influence on the ability to determine whether the application SLA is met using a static SLA template. Sporadic and transient network phenomenon may impact the application experience while being undetected by a probe-based mechanisms operating at insufficient granularity.

Paths Network KPI metrics to SaaS applications

There is a clear distinction between the Network KPI metrics of the path to a SaaS application and the overall application experience. The overall experience accounts for a number of delays such as delay due to network and due to application processing “in the cloud”. Often both network KPI metrics and application experience are mistakenly used interchangeably. If the path network KPI metrics is not sufficient, the application will be impacted. However, a path not offering the required SLA – as measured by an adopted probing methodology – does not imply

that the application experience will be unsatisfactory. An alternative approach consists in using synthetic tests that emulates user activity

In the next section, we analyse the dynamics of various networking KPIs along path to several key SaaS-based applications and their variability across regions in the world and time.

Network KPI metric Variability across entities

The network KPI metrics for SaaS applications vary based on their deployment. One sees variation by customer networks, geo-locations and time. There is also a significant variation even between different SaaS applications from the same network, geography and time. This section outlines the analysis on loss-fraction and latency on paths to various SaaS applications from various customer-networks.

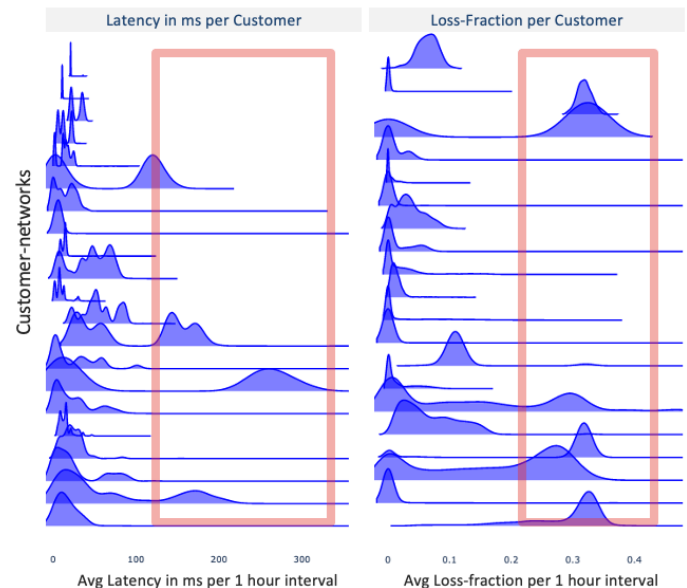


Figure 1: Distribution of loss-fraction and latency for paths carrying SaaS application traffic across different customer-networks. The red box indicates metric ranges that would cause disruptions in user application-experience.

Figure 1 shows the distribution of round-trip latency (left) and loss fraction (between 0 and 1) for paths carrying SaaS application traffic in different customer networks. These are measured from the edge-router to the SaaS advertised endpoints [1]. The network KPI metrics for SaaS paths show a high degree of variability per customer-network. While most of the customers have an acceptable distribution of the metrics, a significant set of the customer networks show metrics which could cause problems to the user application-experience. Note that the cause for such bad network KPI metrics for certain customer networks can be due to plethora of factors such as SPs connected or a site- or router-level problems.

Figure shows the latency and loss metric variability for three popular SaaS applications being measured from multiple edge-routers. Across customers the Network KPI metrics varies for different SaaS applications. Figure 2 shows that the KPI metrics across different SaaS applications vary even within the same customer. This might be due to various other causes such as geo-region of the sites.

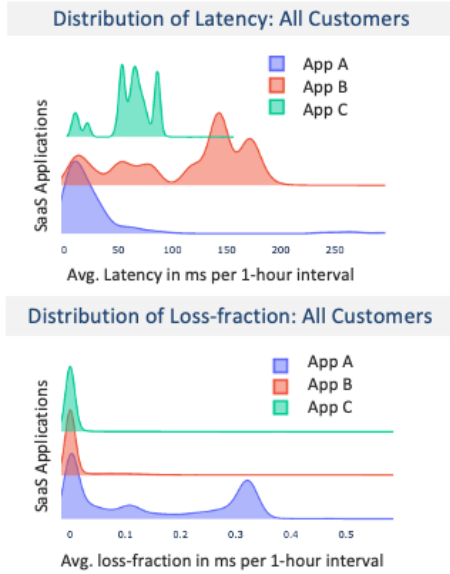


Figure : Distribution of Network KPI metrics for three different SaaS applications seen in customer networks. Shows the variability of Network KPI by SaaS app

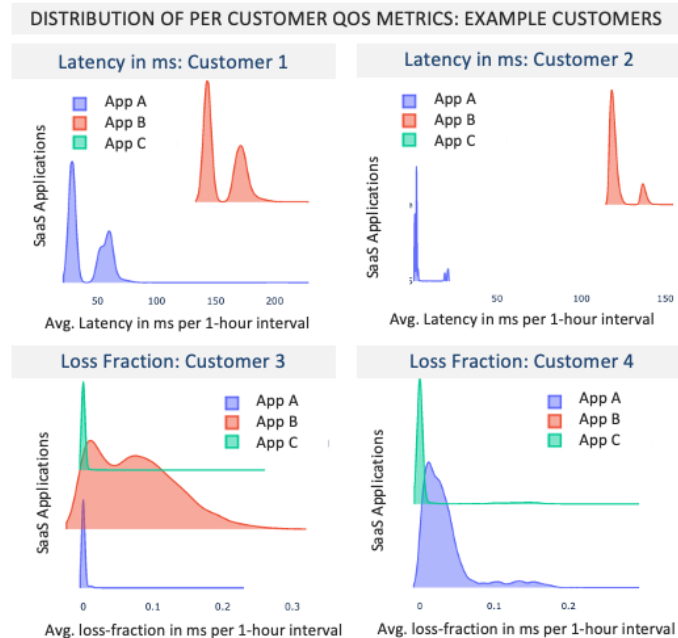


Figure 2: Distribution of Network KPI metrics for example customers. The distributions vary by SaaS application for the same customer-network. Shows the variability of Network KPI by SaaS app.

Figure 3 shows the variability of Network KPI metrics from edge-router to one popular SaaS application with respect to geo-regions. The metrics also vary by the geo-location of the SaaS deployments or the locations of devices accessing the SaaS servers. Average loss in many Asian countries is significantly high (> 20%).

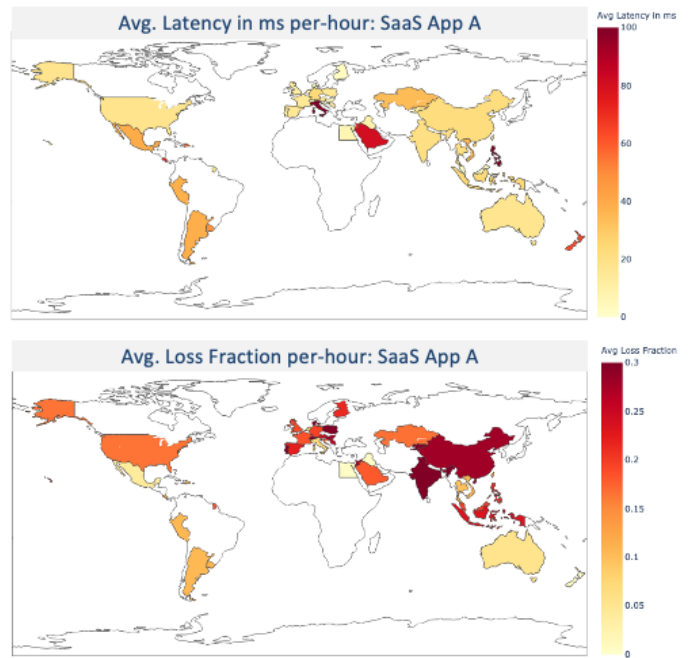


Figure 3: Per-hour Avg. of Network KPI metrics seen across geo-locations for SaaS App A. Shows the variability of KPI by geo-location.

How about time variability?

The Network KPI metrics exhibit different types of time-series characteristics depending on the application, customer-network, geo-location, etc. The time-series characteristics can include periodicity, flat, noisy, flat with random spikes, etc. This section describes the time-variability of KPI metrics by comparing distribution of KPIs at different hours-of-the-day across different days.

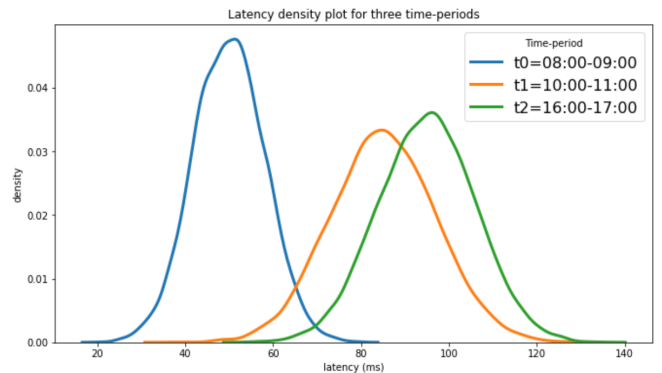


Figure 4: Illustration of the distribution of KPI metric latency measured at three time-periods $t_0=08:00-09:00$, $t_1=10:00-11:00$ and $t_2=16:00-17:00$ hours on different days

For illustration, consider the distribution of latency KPI across three time-periods $\{t_0, t_1, t_2\}$, which are measured at different hours-of-the-day, say, 8:00-9:00 UTC, 10:00-11:00 UTC and 16:00-17:00 UTC, respectively. Figure 4 shows the distributions at these one-hour durations. These distributions can be constructed from raw observed KPI values, or based on modelling these KPI metrics [2]. Distributions at times t_1 and t_2 are more similar to each other, rather than the latency at time t_0 , which is significantly lower. The below metrics capture such dissimilarities in distributions by measuring "Pairwise Distribution Distance": the higher the is distribution distance, the lower is the similarity. The distribution distance can be measured using various non-parametric two sample testing techniques such as KS-stat, Cramer-von Mises or Wasserstein

distance [3]. Pairwise Distribution Distance matrix computes the pairwise distribution distances across all hours-of-a-day for a given day. This is an indicator of the temporal variability and measures the similarity of network distributions across time.

This temporal similarity is shown as a matrix as shown in Figure 5 for the loss-fraction KPI. A particular cell in the matrix represents how different the distribution of a given Network KPI metric is from one hour-of-day as indicated on the Y-axis compared to a different hour-of-day (X-axis) on the same given day. This helps in understanding which hours-of-day in a given day had similar distributions for Network KPI metrics.

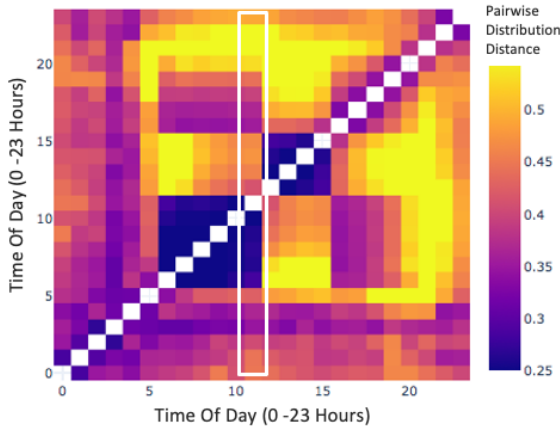


Figure 5: Example block representing the pairwise Network KPI metric distribution distance. Each column represents the distribution distance for a given hour-of-day with every other hour-of-day for a given day. The Network KPI metric being used for this example is Loss-Fraction. For the given example, one can observe two distinct groups hours 05:00 to 10:00 (5-10 on X- and Y-axes) and hours 11:00-16:00 (11-16 on X- and Y-axes), which have very similar distributions in loss-fraction.



Figure 6: Pairwise Distribution Distance matrices and the corresponding timeseries for Loss-Fraction of four example customers exhibiting varied timeseries characteristics. The Loss-Fraction is taken for all paths that carry SaaS application traffic. The matrices offer a visually easier approach to understand the general timeseries characteristics for a larger entity like customer, region etc.

Figure 5 shows a matrix representation to demonstrate the time-variability of a loss-fraction. Each element M_{ij} in the matrix would be *distance* between the distributions of KPI metrics at hour-i with distributions of the Network KPI metrics at hour-j for a given day. For example, hours 05:00-10:00 (as indicated from X- and Y-axis range 5-10), has very small pair-wise distribution distance (indicated by the blue color), signifying that loss values at those times are highly similar.

Network KPI metrics for SaaS paths exhibit varied timeseries characteristics depending on the region, customer-network etc. Figure 6 describes the approach to interpret the Pairwise Distance Matrices to understand the general timeseries behaviour. Figure 6 shows the pairwise distance matrix for loss-fraction across four different days for four different customers (one customer per row of graphs). The right side of the figure shows the scatter time plot of observed loss-fraction.

- Periodic (seasonal) behaviour is represented by distinct blocks of hours-of-day which have low distribution distance. This pattern repeats every-day (or weekday) for entities with periodicity in Network KPI metrics. For example, Customer-1 has an interesting periodic behaviour. Loss is very similar distributions between hours 12:00-00:00 (UTC) on X-axis, and 12:00-00:00 (UTC) on y-axis indicated by the blue block on the top right of "Customer 1" across days. Hence, loss during these hours is similar. However, the yellow block on top-left and bottom-right show that the distribution is very different between 00:00-12:00 UTC and 12:00-00:00 UTC. This suggests periodic variation is loss as shown in the respective time-plot on right side of "Customer 1"
- Noisy timeseries behaviour can be understood with matrices displaying very random high distribution distances. This is evident because, having noisy behaviour generally causes a high distribution distance of any hour-of-day with another hour-of-day. For example, loss for "Customer 4" has not much stable structure across different hours of the day. Note that the actual phenomenon might not be "noisy"; there might be other complex phenomenon that needs to be captured. But from these temporal distribution distances and visual exploration of the similarity across hours, such phenomenon may not be captured yet.
- Flat/Stable behaviour is generally recognizable by matrices which depict low distribution distance throughout (e.g., Customer 3), barring a few sporadic spikes. This is obvious as a stable timeseries essentially gives the same distribution of Network KPI metrics across all hours-of-day.

The pairwise distribution distance matrices can then be used to understand the general timeseries characteristics across entities like SaaS applications, customers, regions etc.,. The distribution of any Network KPI metric for a given hour-of-day is compared with the distribution of that metric over all other hours-of-day where, the distribution is comprised of all the SaaS paths that are associated to the entity we choose to slice by.

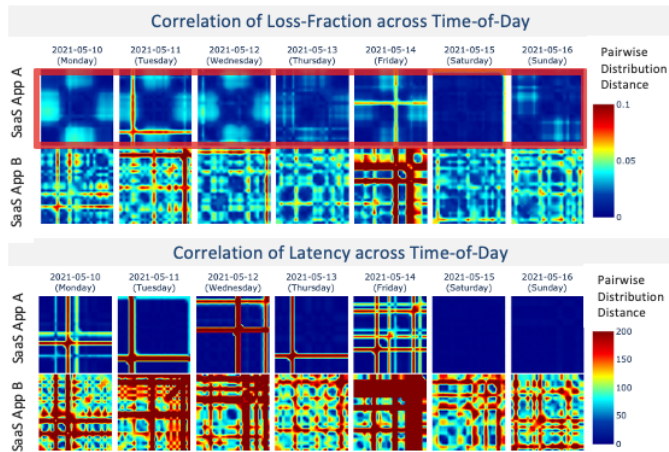


Figure 7: Sliced by SaaS Apps - Pairwise Distribution Distance matrices representing Loss-Fraction and Latency across different days for two example SaaS applications.

Figure 7 shows that, for two different SaaS applications, the Network KPI metrics exhibit two very different characteristics in time. For SaaS App A, there is a significant periodicity for loss-fraction (as indicated by light and dark blue blocks), with a sufficiently stable latency. However, for SaaS App B the behaviours for both loss-fraction and latency is extremely noisy. This shows how varied the treatment of any decision-making engine needs to be for any two SaaS applications.

Similar degree of variability can be seen for time-series behaviour when the distributions are sliced per-customer, as shown in Figure 8 and Figure 9. Periodic patterns can be clearly seen by locating the blocks of highly similar (blue blocks) and dissimilar (yellow/red blocks) hours-of-the-day.

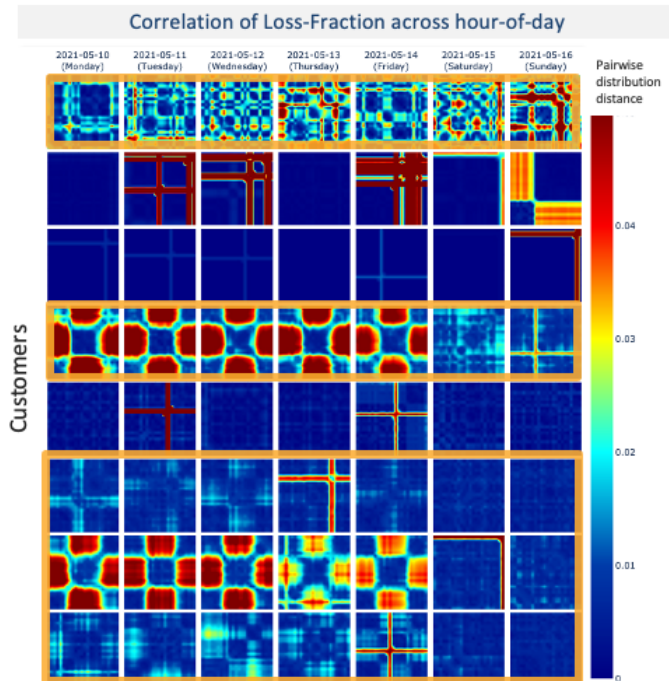


Figure 8: Loss-Fraction variability across time: Pairwise Distribution Distance matrices representing each day in a week. Distribution of loss-fraction is taken across all telemetry seen for a customer on a given hour-of-day and date.

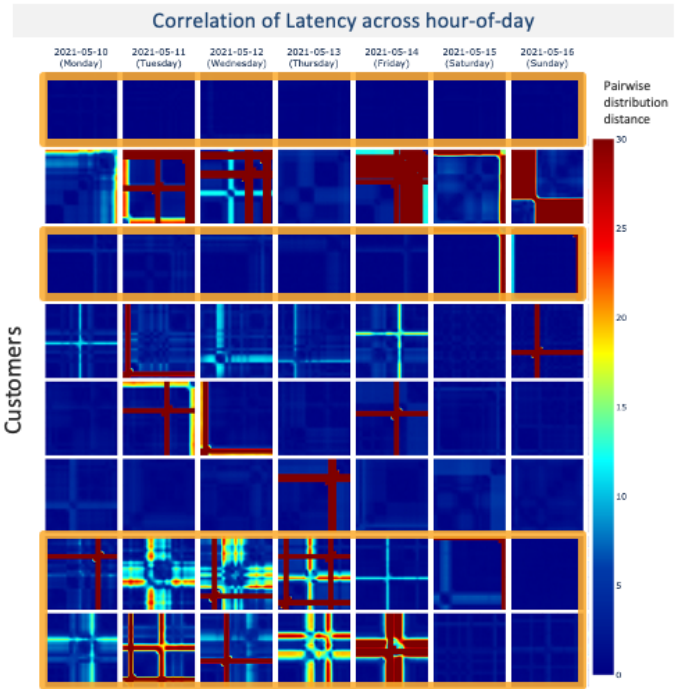


Figure 9: Latency variability across time: Pairwise Distribution Distance matrices representing each day in a week. Distribution of latency is taken across all telemetry seen for a customer on a given hour-of-day and date.

The time-series behaviour does vary for each customer-network. While few of the customers exhibit stable Network KPI metrics with occasional fluctuations, few others exhibit significant periodicity and a high degree of noise.

Measuring the application experience: Thanks to application feedback

Application feedback is the ground truth all scientists look for. In such a scenario, the application itself provides feedback about the user experience taking into consideration Layer-7 (L7) metrics not available at the routing or transport layers. That being said, the application layer may itself make use of heuristics on lower-level metrics so as to measure the user/application experience. For example, in voice systems, *concealment time* measured at the voice-codec is known to be a strong indicator of the user experience; certainly more accurate than the aggregated delay or loss of the path between users. Other systems may make use of Anomaly Detection (AD) strategies to detect L7 anomalies used as a proxy of the user satisfaction while other strategies specify a hard bound for L7 tasks to be accomplished (e.g., layer 7 transaction time).

In this section, we conducted an in-depth analysis collecting up to 1 billion records coming from voice and video systems where each record reports several hundreds of parameters related to voice and video quality for voice/video around the world. This data is gathered from more than 30 million user sessions across 73000 customer networks per day. The data provides more than 500 metrics for every user every minute. This includes voice metrics such as:

1. Identifier columns such as user-id (hashed), voice server and call identifier, along with timestamp
2. Concealment time (CT): CT is the time-period when the decoder inserts dummy voice packets since the original data packets were lost [4].
3. Maximum consecutive CT: This is the maximum consecutive time where the decoder conceals the lost packets.

4. End-to-end round-trip time: This is the latency between the user-device, to voice/video server, to another user.
5. Loss, Jitter and bitrate (tx and rx): End-to-end loss and bit-rate
6. User to Voice/Video-server Loss and jitter between user-device and voice-server.
7. Device statistics: User device statistics such as OS, browser, type, type of network connection and CPU and memory

An interesting observation is that voice/video quality reporting done by some metrics called the UES (User Experience Score) constantly varies across region and time. Figure 7 and Figure 8 shows multiple snapshots of the percentage of voice/video calls with a "non perfect" MoS score (UES<4). It can be observed that such percentage drastically varies over time, even during a single day.

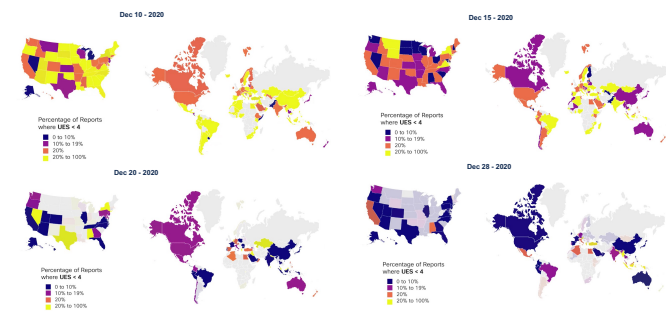


Figure 7: Variability of Voice/Video experience over regions and time

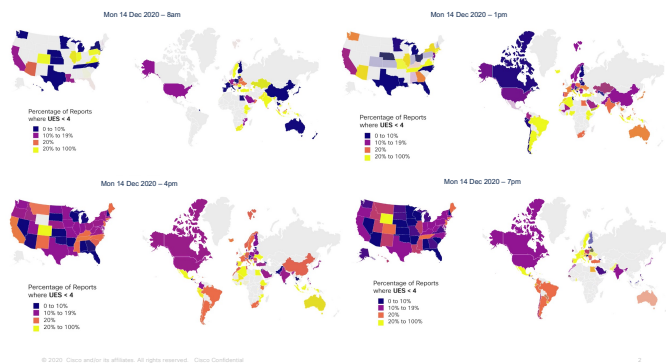


Figure 8: Variability of Voice/Video experience within in single day

But why? There are several reasons that could lead to bad voice/video user experience; problems on the network (path Network KPI to SaaS varies over time and region), problems on the application-server side and problems on the user-device side.

Figure 9 shows the user experience score distribution from different Service-Providers (SP) to the respective voice/video servers. The SP is the service-provider to which the client is connected and the SPs are tagged at a city-level, i.e., SP is a combination of SP-name and city. This is done because the same service provider may provide different network experience – in different cities – which in turn might affect user-experience. Each distribution is coloured by the 10th percentile (the worst few measurements) of the UES seen for that SP. The median and 10th percentile is also marked. It can be seen that some SPs provide bad UES with most UES scores in the range of 1-3 (top, red-coloured SPs), and others (bottom blue-ish distributions) provide consistently good UES.

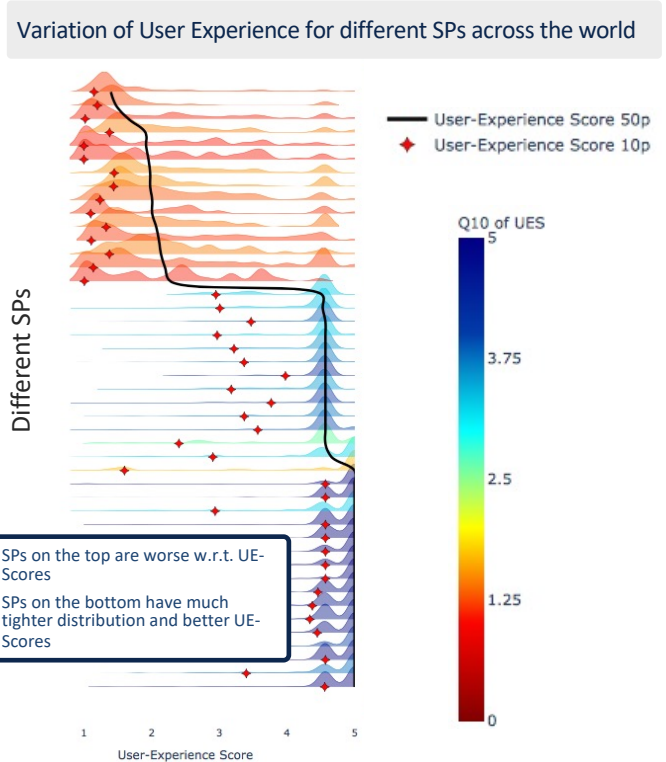


Figure 9: Distribution of User Experience Score for voice connections from different Service Providers (at a city-level)

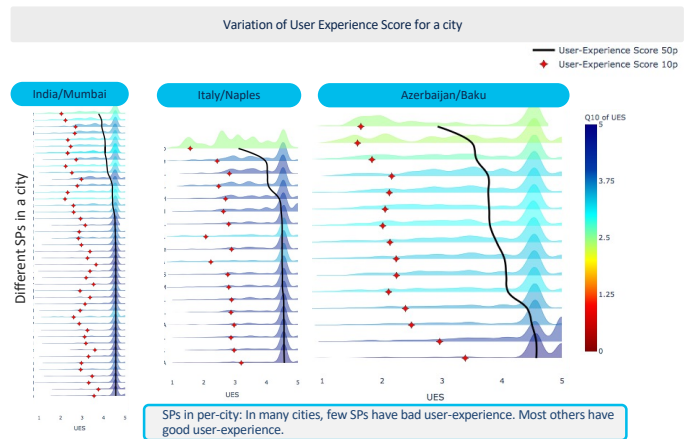


Figure 10: Distribution of User-experience Score for all SPs in three cities

Figure 10 shows UES distribution across multiple SPs for three example cities: India/Mumbai, Italy/Naples and Azerbaijan/Baku. Even individual cities have significant difference in UES based on the SP to which the client is connected. Hence, if an edge-router is connected to multiple SPs, the UES may depend on which SP the traffic goes through.

Microsoft O365 feedback

Microsoft Office 365 (O365) is a SaaS provider that provides multiple productivity tools such as Exchange, SharePoint and Teams video conferencing products. A novel approach has allowed for the gathering of application feedback from O365 using a set of discrete labels collected at regular intervals that were then correlated with the network path measurements. Note that such labels may then be used by a reactive system adapting the routing paradigm to the application feedback or a



proactive system to proactively route traffic according to forecast so as to maximize the user experience (as discussed in [5]).

The data provided by O365 includes a discrete user experience (such as Good, Degraded, Bad), along with end-to-end metrics such as different percentiles of loss and latency.

Further revisions of this document will provide extensive analysis for O365 and other applications.

Conclusion and Next Steps

Although these studies have been performed over months of telemetry and thousands of paths, results need to be taken with a grain of salt as the inferences are not definitely conclusive; the Internet is highly dynamic and so is the cloud. Thus, the results may greatly vary over time. That being said, these results have shown a great deal of variability in several networking KPI influencing the application experience, per geography, Service Provider and Time. This implies that routing systems making use of networking KPI centric templates will need to constantly change and accordingly use reactive or proactive approaches such as in the case of the Predictive Internet (see [5]). On-going analyses is being conducted to analyse the variability of application centric KPI (sometimes generally referred to as MoS), for different applications, geography and time that will be published in further revisions of this document.

Acknowledgement

I would like to express my real gratitude to several key contributors I have been working with for a number of years: Gregory Mermoud, PA Savalle, Michal Garcarz with whom many ML/AI innovations gave birth to novel innovations for networking (Wireless Anomaly Detection with root causing, Self-Learning Networks, detection of spoofing attacks. ...). I would like to acknowledge the work of several engineers: Nathan Buckles, Tim Evens and Cullen Jennings to mention a few who had a major contribution to this work. Needless to say, that a close collaboration with a number of customers in the world has allowed for such an innovative work.

References

- [1] Cisco, "Cloud onRamp for SaaS," [Online]. Available: https://www.cisco.com/c/m/en_us/solutions/enterprise-networks/sd-wan/infographics/cloud-onramp-for-saas.html. [Accessed 12 01 2021].
- [2] V. Kolar, J. P. Vasseur and M. Y. Raghuprasad, *Large-scale Internet Path modelling and applications*, 2020.
- [3] O. Thas, "Methods Based on the Empirical Distribution Function," in *Comparing Distributions*, Springer, 2010, pp. 123-160.
- [4] V. P. Bhute and U. N. Shrawankar, "Speech Packet Concealment Techniques Based on Time-Scale Modification for VoIP," in *International Conference on Computer Science and Information Technology*, 2008.
- [5] J. P. Vasseur, *Towards a Predictive Internet: A new world, with new challenges*, 2021.