BUSINESS WHITE PAPER

Exploring Six Essential Metrics for WebRTC Monitoring in Contact Centers

Maximizing Call Quality in Contact Centers through Explanation and Evaluation of Key WebRTC Metrics

spearline®

Now part of CYARA

CONTENTS

Introduction

Contact centers prioritize one thing above all else: communication. Effective communication is the moneymaker that keeps the contact center machine going. It's also the reason why many contact centers are making the switch to WebRTC.

The promise of WebRTC is to instantly connect anyone, anywhere without the need for expensive infrastructure. Data from customers is transmitted without effort, while information can be shared back with them in a way that delights their senses with full-motion video and high-resolution graphics. In addition, sessions may be shared and joined by others as needed. Calls can easily be routed to the most cost-effective and capable data center for every business operation – including remote and offshore workers. Best of all, all of this can happen over the public Internet, with business logic and data housed in the cloud. Thus, there is no need for dedicated communications hardware. Capital expense is kept to a minimum in the WebRTC world.

But communicating over the Internet raises its own set of issues, especially in our post pandemic era, where work from home is the reality for many contact center agents. This has brought in the notion that every employee runs his own home office, versus the pre-pandemic world, where most IT managers had to contend and manage only a handful of office locations. This makes it doubly important to understand and optimize network connectivity and audio quality so your agents can communicate quickly and easily with your customers. This includes taking things into consideration like jitter, packet loss, and round-trip time. In order to get the most out of your system and save you money, there are six pivotal metrics you need to monitor.

This whitepaper is organized as follows: the first three metrics are basic metrics that indicate the network, operating system, and device performance. Namely, round-trip time, jitter, and packet loss. These three metrics contribute to the subsequent metrics: connectivity establishment and connection time, selection time, and wait time. These metrics each impact the user feedback and the objective quality.

Why WebRTC for Contact Center Agents?

The most classic approach to introducing WebRTC in a contact center is by enabling it for the agent. This brings with it quite a few advantages.

Reduced TCO

Using WebRTC reduces the overall cost of the solution. By using less telecommunication phone numbers and relying on the public internet and its connectivity, the total cost of ownership (TCO) of the solution and its ongoing maintenance reduces.

Increased Operational Flexibility

WebRTC reduces the number of moving parts in a contact center and enables agents to use a single device with a single web application to conduct their calls. There is no longer a real need for physical phones or even installable softphone applications. Removing these and adopting a cloud native approach allows the agents to connect from anywhere. The need for expensive office space is reduced and the ability to easily add capacity and agents on peak hours is increased.

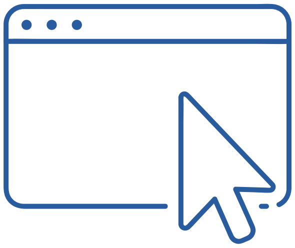
Tighter CRM Integration

As a first class citizen on the web browser, WebRTC can easily integrate with CRM applications, at times even sharing the same browser tab. This reduces the need to switch across tabs and applications and increases the level of integration and information sharing between your contact center application and your CRM (Customer Relationship Management) software.

Why Businesses Use WebRTC

There are many reasons why in this day and age businesses are opting to use WebRTC technology in front of the customer. Here are just a few:

Cross-platform Support

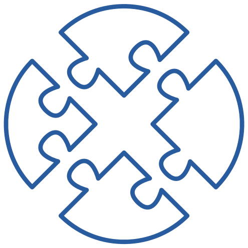


WebRTC lets users communicate directly through multiple means so they can talk to you on their terms, on their preferred platform. It is available on most major web browsers, including Chrome, Safari, Edge Firefox, and on mobile browsers for iOS and Android. Imagine the freedom and flexibility your customers will feel having access to your support teams through their web browser, in-app on their phones, or through a phone call.

Cross-medium Support

Intricately linked to cross-platform support comes the powerful ability to communicate with your users through their medium of choice. For example, with WebRTC, you can give your customers the ability to communicate via video conference, voice call, or through text chat. You can provide these options in the browser or in a mobile app for an immersive, dynamic experience. American Express found great success with WebRTC, with 67% of customers opting to use WebRTC-enabled two-way video.

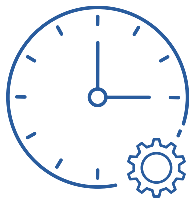
Context-based Support



Context-based support is a unique element your developers can use to create a better experience for your customers. With context-based support, you can pass on information about the user as they initiate the support request. Your agents will have comprehensive information about the user's previous interactions with support, the plan or service they are subscribed to, and more. Even better, your agents can enable screen sharing with the user, so they can see the issue in context at that moment. Connecting with your customers like this will give you an impressive edge over other support operations, and increase your first call resolutions.

Round Trip Time

Round-trip time (RTT) is a critical metric when measuring real-time communications and one of our most viewed dashboard metrics. A low RTT indicates the call center network supports delivery of high quality, reliable customer experiences.



The higher the Round Trip Time, the lower the call quality is going to be.

What is Round Trip Time?

In simple terms, round-trip time is the time it takes for a packet to travel through an IP network, from a sending endpoint to a receiving endpoint and back, not including the time to process the packet at its destination. Many factors affect RTT, like propagation delay, processing delay, queuing delay, and transmission delay.

1. **Processing Delay** is the time associated with the network analyzing a packet header and determining where the packet must be sent. This depends heavily on the entries in the routing table, the execution of data structures in the system, and the hardware implementation.

2. **Queueing Delay** is the time between a packet being queued and it being sent. This varies depending on the amount of traffic, the type of traffic, and what router queue algorithms are implemented. Different algorithms may adjust delays for system preference, or require the same delay for all traffic.

3. **Transmission Delay** is the time needed to push a packet's data bits into the wire. This changes based on the size of the packet and the bandwidth. This does not depend on the distance of the wire, as it is solely the time to push a packet's bits into the wire, not to travel down the wire to the receiving endpoint.

4. **Propagation Delay** is the time associated with the first bit of the packet traveling from the sending endpoint to the receiving endpoint. This is often referred to as a delay by distance, and as such is influenced by the distance the bit must travel and the propagation speed.

Processing, queuing, and transmission delays may vary depending on how many nodes in the network connect the endpoints. The smaller the number of router hops, the smaller the delay. They are typically consistent for a given pair of communicating endpoints. Network congestion, on the other hand, tends to lend a dynamic component to RTT. Propagation delay is a major component of RTT that ranges from milliseconds to hundreds of milliseconds depending on the distance between endpoints. Endpoints that are separated by a few kilometers may experience a different propagation delay than those separated by an entire ocean.

What Causes High Round-Trip Time Values?

In real-time communications, high round-trip time is typically caused by hairpinning. Hairpinning is when media is anchored in a location geographically remote from an endpoint. It can add considerable propagation delay when compared to a peer connection. This is why choosing the placement of your infrastructure can be critical to delivering low RTT and a high-quality customer experience. The further away the media server is from the sending and receiving endpoints, the higher the RTT value and the lower the service quality. This is something to consider when setting up your contact center infrastructure, and if you use a service provider, this may be an important question to raise with them.

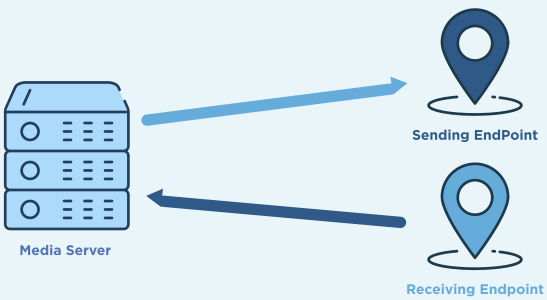


Figure 1: The media server is located further away than necessary from the sending and receiving endpoints, resulting in a high round-trip time.

Additionally, bufferbloat may be a cause of high round-trip time. If there are large buffers in the network, they store a lot of data that must be forwarded. This can lead to large RTTs, as packets that enter this large buffer have to wait until all the previous packets have been processed. In the case of overflow, this may also lead to packet loss.

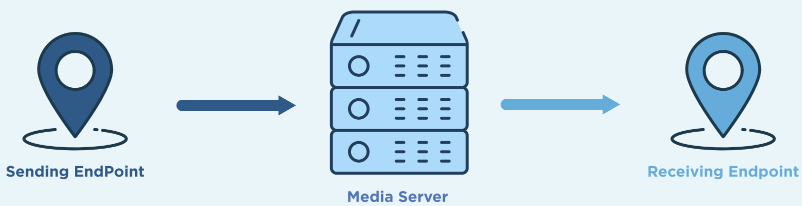


Figure 2: The media server is located between the sending and receiving endpoints, resulting in a lower round-trip time.

Round Trip Time, Delay or Latency?

When talking about Round Trip Time, we cannot ignore two other metrics that are very similar: Delay and Latency.

Latency and Delay are sometimes considered equivalent to round-trip time. Alternatively, some use them to mean the time a packet takes to travel from one endpoint to the other - or in a way - half a round trip.

The reason we use Round Trip Time measurements and not Delay or Latency in WebRTC and other real time communication solutions is that network monitoring tools are very useful at determining the RTT on a given network. It is possible to calculate RTT from the source, since it is able to track the time the packet was sent and computes the difference upon acknowledgement of return. Calculating Latency is a lot harder and usually not worth the additional effort.

Why is Round Trip Time Important for Call Centers?

Putting this into layman's terms, RTT is the delay of the speaker's voice reaching the listener's ear. When communicating with a customer over the phone, this is one of the most important metrics to optimize for.

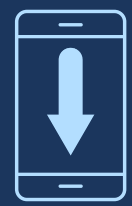
RTT is not noticeable to the average caller unless it is greater than 300 ms. However, above this limit the call center agent and customer may begin to experience noticeable delay on a call. This can easily lead to difficulty communicating, extended time to resolution, call disruptions, and customer frustration. In a business that relies on a good user experience, this metric is absolutely critical to monitor. Depending on your infrastructure, high RTT could be caused by any number of factors, including your service provider. Having contextual, dated values of RTT to compare to your service quality is invaluable. Even more-so, having these values in an easy-to-understand, graphical layout can provide additional clarity to identify when problems occur and why.

How can Contact Centers Improve Round Trip Time?

Understanding the route the media takes from the customer to the agent through your call center is the first step. By understanding the best route and by monitoring and knowing the actual routes taken, you will be able to make routing decisions and reconfigure your network infrastructure. This includes forcing agents or user devices to connect to certain call centers or deploying more infrastructure elements in additional regions.

Jitter

Jitter is an important and widely used metric when diagnosing quality problems with real-time communications sessions. High jitter values contribute to poor audio quality, which can degrade customer experiences and prevent call center agents from communicating effectively.



The higher the Jitter in a call, the lower its quality will be

What is Jitter?

Jitter is the amount of variation in packet delay, which is why it is also frequently called delay variation. It is measured by calculating the average variation in packet arrival times at the receiving node occurring over a fixed interval. At testRTC, the jitter calculation interval is a configurable parameter on the account level.

In VoIP and WebRTC networks, audio packets are transmitted on the network continuously, typically every 20 milliseconds. They inevitably arrive at the receiver with differing delays, even when traveling the same path. Constant delay pacing simply cannot be guaranteed.

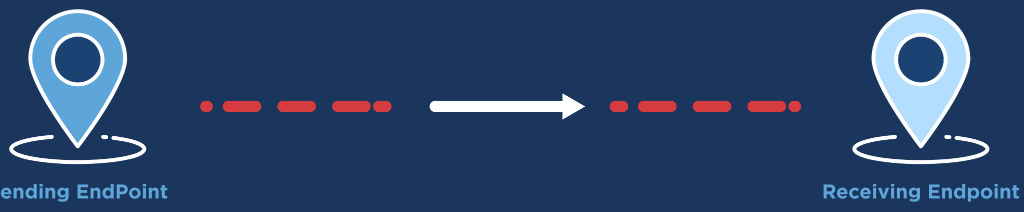


Figure 3: The receiving endpoint getting sent packets with different delays, also known as jitter

This differing delay can be a significant issue for real-time communications media streams, including call centers, because a quality reproduction of the sender's voice requires data to be played at the receiver at a constant rate.

What Causes Jitter?

Two main factors determine the amount of jitter in a media stream: network congestion and wireless networks.

1. **Network congestion.** Jitter is generally caused by congestion in the IP network. The congestion can occur either at the router interfaces or in a provider or carrier network if the circuit has not been provisioned correctly. Having too many devices hooked up to the same system, all being used at the same time, will run out of bandwidth, slowing your connection to a crawl.

2. **Wireless networks.** While a wireless network allows mobility and eliminates the need for wires running through the office, there is a chance that you will experience a degraded network link. WiFi is not necessarily secure or stable enough for our mobile devices to depend on for our phone calls.

Other notable causes:

- Misbehaving hardware
- Hardware capacity on source/destination
- Network capacity
- Bad routing through network/internet (traffic routed in a suboptimal fashion introducing delays, etc)

Congestion and packet loss cause packets to arrive at the receiver with varying amounts of delay, leaving the endpoint to deal with the problem of recreating a continuous listening experience.

Why is Jitter an Important WebRTC Metric for Call Centers?

Endpoints are designed to smooth out minor variations in arriving data by intentionally delaying the playout of incoming packets. Packets enter a jitter buffer that adds an imperceptible amount of delay to the listening experience, but allows the endpoint to create a continuous audio stream for the receiver. However, when jitter becomes excessive, the buffer can be emptied or it can overflow, causing choppy or scrambled audio experiences.

For an industry that relies on high-quality calls, this can be a huge problem. Poor audio quality can result in customer confusion, frustration, or even worse, churn. It also has the potential to extend time-to-resolution and lead agents to miss out on upselling opportunities.

How can Contact Centers Compensate for Jitter?

Jitter problems can be addressed by relieving network congestion or increasing the size of jitter buffers, or both. Network managers need to determine which router(s) is experiencing congestion and increase its bandwidth or change the queuing discipline to expedite real time packets, as appropriate.

Endpoints may be equipped with one of two types of jitter buffers: static and dynamic. Static jitter buffers are implemented directly in the hardware of your system and are configured by your manufacturer, while dynamic jitter buffers are implemented in the software and configured by a network administrator. Dynamic jitter buffers can be increased to deliver a better listening experience for the receiver, albeit with a bit less interactivity in the conversation.

Packet Loss

Packet loss is an important metric for call centers looking to have high quality audio. Depending on the level of packet loss, poor call quality can produce a poor customer experience. What is packet loss and why should WebRTC-enabled contact centers work hard to keep it at zero?

What is Packet Loss?

Packet loss is just as it sounds: transmitted packets have been lost somewhere along their journey across the network and do not arrive at the receiving endpoint. Packet loss can be caused by any number of things: in wireless networks, inadequate signal strength, interference and poor network handoffs cause loss; in all network types, excessive congestion, system noise, hardware failures and software corruption can cause loss. Often, one of these failures can have a ripple effect, making things even worse.

There are multiple packet loss metrics that you should monitor to ensure users receive good audio quality. The raw packet loss metric only tells you how much of the transmitted data is lost in transit. It doesn't tell you whether this lost data is affecting user experiences. To do this, you also need to evaluate post-repair packet loss and discarded packets, which are described in the next section.

The higher the Packet Loss, the more choppy and unintelligible the call will be

What Happens When Packets are Lost?

Fortunately for contact center managers, WebRTC was designed to deliver good quality even when sessions encounter packet losses, as is typical on public internet networks. Two methods are used to recover from packet loss: Forward Error Correction and retransmission.

Forward Error Correction (FEC)

The Opus codec that is part of the WebRTC specification features FEC mechanisms that can recover some or all of the information of lost packets. FEC protects against a packet loss by sending redundant information to the receiver. This redundant information enables the receiver to recreate some of the lost packets.

Sudden bursts of multiple lost packets can be protected by alternate FEC schemes. For example, the sender may transmit multiple FEC packets protecting all packets in a 10 packet window, or all even packets in the window. FEC can adjust and send more redundant data to protect more packets as loss levels change during a session. When loss rates are excessive, the mechanism may be ineffective. In this case, the lost packets are counted as "post-repair packet loss" and audio may be noticeably impaired.

FEC is complemented by concealment mechanisms that help to minimize the audible effects of packet loss. If there is residual loss after repairing packets the receiving system uses error concealment which may result in a slight degradation in audio quality.

FEC adds bandwidth overhead, which can be a worthy trade-off for the additional audio quality it provides. When loss rates spike and FEC has to adjust to protect more packets, the bandwidth overhead also increases.

Retransmission

Retransmissions take advantage of the sequence numbers embedded in each packet in a media stream. When the receiver detects a packet is missing from the sequence, it notifies the sender. The sender determines whether to retransmit the packet or adjust the FEC scheme. Time is of the essence for retransmissions because the replacement packet must arrive at the receiving endpoint before the decoder assembles the packets into audio. The sender evaluates the round trip time and determines whether to retransmit. If the retransmitted packet arrives too late, the receiver discards it, which is counted as a "discarded packet."

For the most part and in most cases, your WebRTC contact center will not be retransmitting audio packets due to packet losses. By default, this just isn't what WebRTC does "out of the box".

Why is Packet Loss an Important WebRTC Metric for Contact Centers?

Ideally, packet loss should be maintained near zero for the best network performance and audio quality. Low loss rates mean FEC and retransmissions consume low bandwidth overhead and the network is running efficiently. When packet loss rates become significant, you should also examine post-repair packet loss and discarded packet metrics. It's the combination of these three loss metrics that can indicate call quality problems that are noticeable to the users.

High post-repair loss rates and discarded packets can be a huge issue for contact center call quality. It causes WebRTC sessions to send more redundant data, which increases bandwidth consumption. It can also result in audible impairments to the session. Imagine if your agent was communicating a crucial point to a customer, and suddenly that point was gone. Moreover, it can cause customer frustration, as audio quality can be poor. Nothing annoys a customer more than when they cannot communicate their frustrations.

How can Contact Centers Reduce Packet Loss?

While WebRTC has mechanisms to overcome and reduce the effect packet loss has on audio quality, you will be better off to reduce packet losses as much as possible - the better the audio is to begin with the higher the resulting quality will be.

There are usually two areas of focus when dealing with packet losses: understanding your infrastructure and understanding the users.

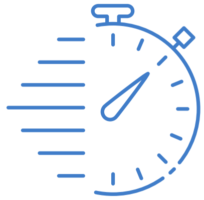
You need to make sure your network infrastructure isn't to blame for packet losses. At times, congested or throttled media servers, data centers and offices are the reason for excess in packet loss. In other times, media servers and other network elements with high utilization and overtaxed CPUs (Central Processing Units) are the source of the issue.

That said, oftentimes the main culprit is going to be the user's device or network. In today's remote working world where both users and agents connect from anywhere, businesses have a lot less control and visibility towards these network connections.

Being able to identify poor network conditions, especially on the agent's side and assisting them in figuring out the root causes and improving their situation can bring with it huge gains in audio quality and a reduction in packet loss.

testRTC offers multiple tools for both the contact center and the agents to gain these insights and to improve audio quality.

Connection Time, Selection Time and Wait Time



Strive to reduce "application" specific measurements such as connection time, selection time and wait time that aren't specifically affected by the network performance itself

Having a clear understanding of the customer journey when engaging with a contact center agent can be a solid indicator of how effective your contact center is. Actually getting to the point of communicating with the contact center agent is not a given - customers have to address questions, be redirected, and may even have wait times in a queue. It's important to break down this process so you can pinpoint problems along the chain and address any underlying lag to get your customers connected as fast as possible.

What are Connection Time, Selection Time and Wait Time?

Connection time, selection time, and wait time measure a chain of events your customer must experience before reaching an agent.

1. **Connection time** is the time it takes to establish the call, from when the customer presses the dial button to when it rings on the agent side.

2. **Selection time** is the time it takes for the customer to navigate through voice commands and be placed in the appropriate queue. Contact centers typically use Interactive Voice Response (IVR), a telephony menu system, to give the customer the opportunity to identify themselves and the reason they are calling. This saves the customer and agent's time, as the IVR automatically directs the call to agent's with the appropriate skills.

3. **Wait time** is the time from when the customer enters a particular queue until they are connected to the agent. If the customer is added to the wrong queue, they will need to be forwarded to the appropriate queue. This can significantly extend the customer wait time.

Why are Connection Time, Selection Time and Wait Time Important WebRTC Metrics for Contact Centers?

By analyzing each segment of the customer journey, you can target trouble areas and more easily identify potential customer delays. This initial part of the customer journey is critical. Having a positive experience from the start improves customer moods and makes them feel appreciated and heard faster. If agents report customers are frustrated by the time they reach them, you'll want to monitor these metrics and identify the reasons why.

If you have extended connection time, you may need to look at other metrics.

If you have high selection time, you may need to take a look at optimizing your IVR so that it asks more targeted, effective questions and chooses an agent queue more efficiently.

If you have high wait times, you may need to hire more agents to reduce times to something more sustainable. Alternatively, if some agents are swamped with work while others have none, you may need to make optimizations to your IVR. For example, if your IVR is sending individuals to the wrong queues, you should look into ways to improve it. Lastly, if some agents are still getting limited customers, it may be time to redirect additional agents to other, more popular sectors.

Connectivity Establishment

Contact center agents need to reliably connect to the cloud service that delivers their calls, no matter where the agent is located or what network they are connected to. Unless the agent's computer happens to be directly connected to the internet (nearly impossible), their connectivity depends upon being able to traverse a NAT (Network Address Translation) device. In this case, special protocols are used to identify the addresses that the endpoints must use when sending packets to each other.

The higher your Connectivity Establishment is, the more calls get successfully connected and handled

With WebRTC, connectivity establishment is brought on by Interactive Connectivity Establishment (ICE) using STUN and TURN to help WebRTC endpoints traverse a NAT device. If ICE fails to do its job, endpoints can't communicate and your agent won't be able to initiate or receive calls.

What is Connectivity Establishment?

Connectivity establishment describes the stages of address negotiation that occur during the call establishment operation (either at the beginning of the call or upon reconnection), including:

- Waiting: needs the IP address of the remote party.
- In-progress: checking connectivity with the remote party's IP address.
- Succeeded: connectivity established.
- Failed: no connectivity established (or connectivity lost) with the IP address pair.

We separate these into failed calls, established calls, and reestablished calls. For our purposes, we consider failed calls to be when no connectivity is established.

1. **Failure**. In the case of no connectivity, you can check your ICE data table to identify whether your agent received any host, server reflexive, peer reflexive or relay candidates. A configured TURN server can help navigate around pesky NATs and Firewalls, in those cases the active ICE candidate pair should be of relay type.

2. **Established, but fails**. When there is connectivity, but network disruptions cause the call to fail, this is an established but failed call. Endpoints will subsequently choose a different candidate pair based on priority in the succeeded state from the set of candidate pairs and induce an ICE restart. The new candidate pair's nominated field should be set to true to indicate it is the most recently used candidate pair.

3. **Reestablishment of calls**. Reestablishment of calls is often caused by handovers, like when candidate pairs change from the mobile network to WiFi. You can use this to identify disruptions in call quality from handovers. In this case, an ICE restart will most likely be necessary.

Some valuable ICE data worth taking a look at includes timestamp data, local address, remote address, RTT, state, priority and bytes sent and received.

Why is Connectivity Establishment an Important set of WebRTC Metrics for Contact Centers?

Connectivity establishment is invaluable when diagnosing call quality issues. If your agents are having trouble establishing connections or losing established connections with the customer, this is where you want to look first. The ICE data table gives multiple different metrics for the local and remote candidates, so you can see if the issue is on the cloud provider side or agent side.

As a contact center, you want to maintain high quality audio. But even more than that, you want to be able to resolve call quality issues as quickly as possible. It is inevitable that call quality issues will crop up, so you need to have the tools at your disposal to resolve them efficiently. Connectivity establishment gives you this ability. You can resolve your call quality issues faster, so you can keep your customers happier.

User Feedback and Objective Quality

Many contact centers solicit feedback from users about call quality just before a call ends. By gathering user feedback, you get a direct indication of call quality from the user's perspective. However, this isn't always reliable. Not only do many customers avoid giving feedback, but they also tend to give a biased answer regarding call quality based on other factors like the result of the call, whether they liked the call center agent, how the conversation flowed, and more.

Look at both subjective user feedback and objective quality metrics to optimize and improve call quality in your call center over time

What are User Feedback and Objective Quality?

User feedback is feedback directly from the user about the quality of the call's audio streams. It's usually collected by an IVR system that enters the call after the agent interaction is complete. Along with measuring NPS, agent responsiveness, and other factors, these surveys can collect feedback on overall audio quality. Many cloud-based contact center services can also collect agent's feedback from within the agent's app.

testRTC's tools can assist in the collection of user feedback while at the same time using the collection of all audio metrics in WebRTC, which are objective in nature. These audio metrics are then used to create a single quality score along with a MOS score for the whole system. These aggregated values can be broken down to specific agents, specific interactions and even to the minute level of media streams and their specific audio quality metrics.

By evaluating user feedback and objective quality values in conjunction, you can more effectively identify when call quality is truly poor and dig deeper to find and address these issues.

What Causes Low User Feedback and Objective Quality Values?

Low user feedback ratings can be caused by any number of factors. Ideally, all customers will respond and evaluate the call's audio quality. However, it doesn't always happen this way. First, customer feedback is limited because some individuals may not want to respond. Second, the rating may be influenced by other things going on in their life, how the conversation with the call center agent went, or even whether they are in a rush. Third, each customer may assign a different score to calls that deliver the same quality. These factors make user feedback extremely subjective and a poor indicator of call quality. Typically, low user feedback scores correlate better than high scores.

Objective quality, however, provides a common yardstick for measuring call quality based on technical data collected at each endpoint and overcomes the socio-economic biases of qualitative user-feedback. It combines several metrics such as jitter values, throughput values, delay variations, packet loss, and concealment metrics, audio variations, audio output, and sync time for audio to form an accurate and direct indicator of call quality.

Objective quality values allow call center infrastructure engineers to quickly identify when there is a problem, so they can dig deeper into other metrics and find a solution.

Why are User Feedback and Objective Quality Important WebRTC Metrics for Call Centers?

User feedback is a critical metric for contact centers because it gives a direct link to how the customer felt about the call. Direct feedback from the customer is the quickest indicator of call quality. If user feedback is consistently negative, it is a sign something is very wrong - whether that be with agent conversations, call quality, or some other factor. It's crucial to monitor this metric and use it to guide your contact center operations. Without engaging with and making improvements based on this metric, you may find that customers are wary to interact with your contact center. This can result in customer frustration, missed business opportunities, brand degradation, and potential customer churn.

Objective quality gives your operations team a deeper understanding of whether calls are of high quality. This metric gives you insights before your users become so frustrated they flood you with negative user feedback. You can address poor call quality immediately and prevent customer frustration and churn. This will also aid your contact center agents, as without poor quality calls, they are able to complete calls quicker, keep the customer happier, and maintain a smoother conversation. This opens up more business opportunities and quicker time to resolution with fewer transfers, resulting in higher customer satisfaction.

Conclusion

Communication is key for contact centers. It is critical for contact center agents to be able to communicate effectively with customers so they feel heard and can have their needs met. WebRTC is prime for enabling this communication.

WebRTC is able to instantly connect anyone, anytime, anywhere, through the channels they prefer. Information can be shared through their favorite medium, which improves customer satisfaction. On the infrastructure side, capital expense is kept to a minimum, since with WebRTC, there is no need for dedicated communications hardware.

With all of these benefits comes the need to optimize audio and video quality. By tracking and improving your output on these six key metrics, you can ensure a better customer experience combined with business and resource savings.

Spearline Is Joining Cyara's Solution Family!

The award-winning Cyara Automated CX Assurance Platform enables companies to deliver better CX with less effort, cost, time, and risk. Cyara supports the entire CX software development lifecycle, from design to functional and regression testing, load testing, and production monitoring, ensuring enterprises can build flawless customer journeys across voice and digital channels.

Spearline is a technology company dedicated to helping businesses achieve optimal communication quality and deliver outstanding customer experiences. Our proactive testing platform, Voice Assure, monitors worldwide toll-free, toll, and premium rate numbers for audio quality and connectivity. This ensures that our clients can identify and address issues before their customers are impacted. In addition, our testRTC product set offers comprehensive WebRTC testing, monitoring, and support for web-based communication. Our diverse product range enables us to serve a wide range of businesses across all industries by helping their customers to connect seamlessly with them.

Customer Smiles. **Delivered at Scale.**

Learn more at www.cyara.com

LinkedIn.com/company/Cyara

Twitter.com/GetCyara

Facebook.com/Cyara

YouTube.com/Cyara