Quality by Design

A technical paper prepared for presentation at SCTE TechExpo24

Shafayet Khan
Lead Software Solutions Engineer
CableLabs
s.khan@cablelabs.com

Tyler Glenn
Principal Engineer
CableLabs
t.glenn@cablelabs.com

Paul Fonte
Director, Future Infrastructure Group
CableLabs
p.fonte@cablelabs.com

Table of Contents

List of Figures

1. Introduction

Optimizing network performance is essential for service providers and application developers to ensure the best possible experience for end-users. Traditionally, networks such as DOCSIS® networks, optical networks, and mobile networks have operated in silos, each managed independently. This fragmented approach often results in operational inefficiencies, extended service delivery times, and compromised user experiences due to delayed issue resolution and suboptimal network performance. For application developers, this fragmentation can lead to significant challenges in ensuring their applications run smoothly and deliver optimal performance, as they must account for varying network conditions and limitations. Additionally, problems might not always stem from the network; issues may actually originate within the applications themselves. Identifying these issues is vital for developers, and network insights can be instrumental in pinpointing and resolving such problems.

To address these challenges, CableLabs® introduces the Quality by Design (QbD) (Fonte & Khan, 2024) specification within the Network as a Service (NaaS) framework. QbD is a comprehensive approach aimed at enhancing network performance and user experience through real-time monitoring and automated resolution of network issues. Quality by Design leverages a set of APIs to facilitate two-way communication between applications and the network, transforming applications into network monitoring tools. This innovative approach allows applications to share real-time Key Performance Indicators (KPIs) with the network, providing true visibility into user experience. By correlating application KPIs with network performance data, QbD enables rapid identification and resolution of network impairments and helps determine if issues are rooted in the applications themselves.

End users can experience degraded quality, leading to suboptimal application performance, caused by various factors across the network or within the application itself. QbD addresses this by enabling applications to not only monitor network conditions but also influence network behavior through shared KPIs and network requirements. This proactive approach ensures that applications can dynamically respond to network conditions, thereby enhancing the overall customer experience.

In essence, QbD empowers applications with the capability to actively participate in network management, ensuring a seamless and high-quality user experience. By integrating real-time data sharing and automated responses, QbD offers a robust solution to the challenges of traditional network management, paving the way for more efficient and responsive network services. This approach is beneficial for both network operators and application developers, fostering a collaborative environment that enhances performance and user satisfaction across the board.

Key Features of QbD include**:**

- **Real-Time KPI Collection:** Applications can share KPIs with the QbD API, which triggers customer events based on a quality score. This facilitates the collection of real-time network telemetry, allowing for timely identification of performance issues.
- **Identification of Potential Impairments:** By correlating application KPIs with network telemetry data, QbD helps identify network issues in near real-time. This proactive approach allows for swift diagnosis and mitigation of impairments that could affect user experience.
- **Automated Solutions:** QbD reduces the incidence of excessive network alarms and provides rapid automated responses to address suboptimal application performance. Automation ensures prompt resolution of network issues, maintaining high service quality.

The implementation of QbD within the NaaS framework represents a significant advancement towards unified and efficient network management. By enabling real-time monitoring, analysis, and automated resolution of network issues, QbD ensures optimal network performance and enhances user experience. This comprehensive framework addresses the inefficiencies of traditional network management, paving the way for a more integrated and responsive network infrastructure.

2. Architecture

This section provides a high-level architectural overview of the QbD Network Service. QbD is designed to ensure optimal network performance and quality through systematic monitoring, analysis, and automated resolution of network issues. The key components of the QbD architecture are described below followed by the flow of data within the system.

2.1. Key Components

Customer Premises Equipment (CPE): Customer Premises Equipment (CPE) refers to devices located at the customer's site that connect the customer's network to the operator's network. CPE includes residential and business gateways, cable modems, access points, etc.

Access Network: The Access Network connects the end-user (customer premises) to the primary core or distribution network of a service provider. It serves as the link between subscriber devices and the extensive network infrastructure, enabling users to access services such as internet, telephony, and television. Access Networks include cable broadband such as DOCSIS, Fiber to the Home, wireless technologies such as 4G and 5G, etc.

Core Network: The Core network refers to the central part of the network infrastructure that interconnects various access nodes and headends, facilitating the management, routing, and distribution of high-speed data, video, and voice services. The core serves as the backbone that supports the extensive delivery of services to users via coaxial cable segments, fiber, and wireless solutions such as 5G.

2.2. Physical Topology and Flow

The diagram below represents the physical placement of devices and network infrastructure. It provides a step-by-step flow of how data is shared between the Application and the Network.

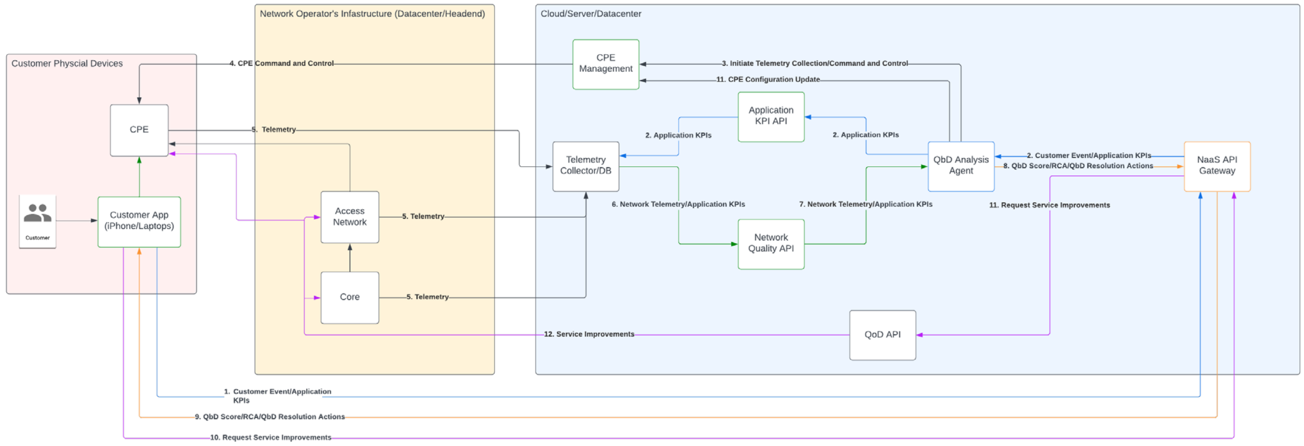

Figure 1 - Physical topology and flow diagram

The QbD framework integrates various devices and network infrastructure components to ensure optimal application performance through systematic monitoring, analysis, and automated resolution. The diagram below illustrates the physical placement of these devices and infrastructure, providing a detailed step-by-step flow of how data is shared between the application and the network.

Step 1: Application KPI Sharing

The process begins with the application sharing its KPIs with the QbD API in real-time or near real-time. These KPIs include metrics such as Latency, Jitter, Packet Loss, and Bitrate. By sharing these KPIs, the application provides critical data that reflects its performance and user experience. The QbD API then passes these KPIs to the QbD analysis agent, which has pre-defined Minimum Quality Threshold profiles to determine whether the application's performance is optimal, suboptimal, or unusable.

Step 2: Application Performance Scoring

Once the KPIs are received, the QbD analysis agent encapsulates the application's performance into a score. This score categorizes the performance into three levels: Optimal, Suboptimal, or Unusable. This categorization is crucial for determining the subsequent actions needed to maintain or improve application performance.

Step 3: Triggering Customer Event

If the application's performance score falls below the optimal range, a Customer Event is triggered. This event initiates rapid network telemetry collection to identify and diagnose the underlying issues affecting performance. The ability to trigger such events in real-time ensures that network issues can be addressed promptly, minimizing the impact on the end-user experience.

Step 4: Real-Time Telemetry Collection by CPE

Under normal circumstances, the Customer Premises Equipment (CPE) collects network KPIs at regular intervals. However, when a Customer Event is triggered, a command is sent to the CPE via the CPE Management service to collect network telemetry data in real-time. This real-time data collection allows for a more immediate and accurate assessment of the network conditions affecting the application's performance.

Step 5: Collection of Probing Statistics

The triggered Customer Event also initiates the collection of real-time Probing Statistics such as Latency and Packet Loss in the underlying access network. These statistics are stored in the Telemetry Database, providing a repository of data that can be analyzed to identify network issues.

Step 6: Call to Network Quality API

Depending on the type of impairment experienced, the QbD analysis agent makes a call to the Network Quality API to retrieve real-time telemetry data. This step ensures that the most current and relevant data is available for analysis, enabling a more accurate diagnosis of the issue.

Step 7: Aggregation and Data Sharing

The Network Quality API gathers all the aggregated KPIs from the session and shares the data back with the QbD analysis agent. This aggregation of data from multiple sources allows for a comprehensive view of the network's performance and the factors affecting the application.

Step 8: Continuous Monitoring

The Analysis Agent uses the telemetry data from the Network Quality API to continuously monitor application performance and update the QbD score in real-time. This continuous monitoring ensures that any fluctuations in performance are detected promptly, allowing for immediate corrective actions.

Step 9: Analysis and Resolution Actions

The Analysis Agent sends back the determined score, root cause analysis, and appropriate resolution actions to the application. This feedback loop ensures that the application is informed of the network conditions and the steps needed to improve performance.

Step 10: Resolution Implementation

Upon receiving the QbD response with supported resolutions, the application can call the Quality on Demand API to implement actions that improve network and application performance. Examples of these resolution actions include Speed Boost, Packet Prioritization at the CPE, and Profile Management in the underlying access network. By implementing these actions, the application can dynamically adapt to changing network conditions, ensuring a seamless and high-quality user experience.

This structured approach of QbD ensures that applications and networks work together cohesively to maintain optimal performance, benefiting both network operators and application developers by fostering a collaborative environment that enhances performance and user satisfaction.

2.3. Logical Topology and Software Design

The following diagram describes the flow of data and interactions between various system components, from customer devices to the core network and telemetry agents. It provides a conceptual map of how the key components interact with each other.

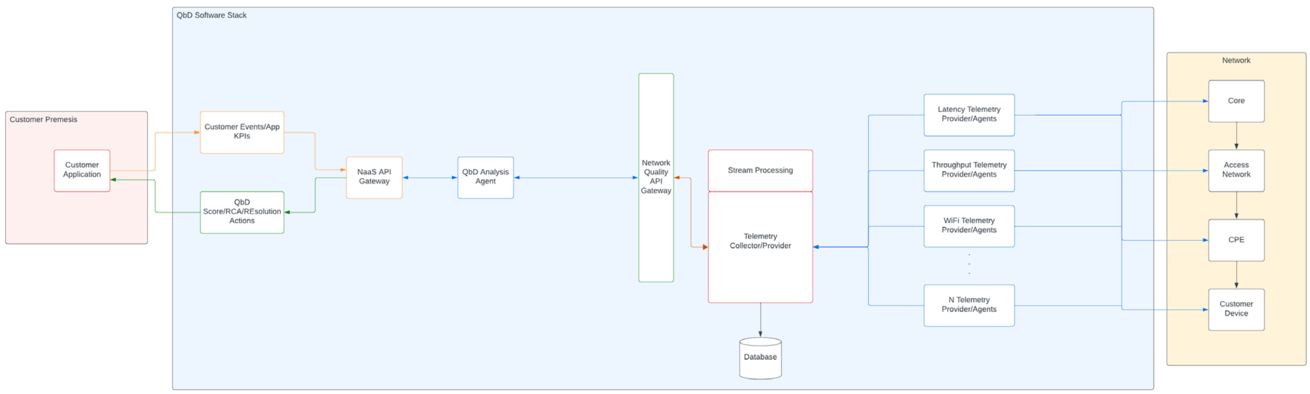

Figure 2 - Logical topology and software design diagram

3. QbD API Flow

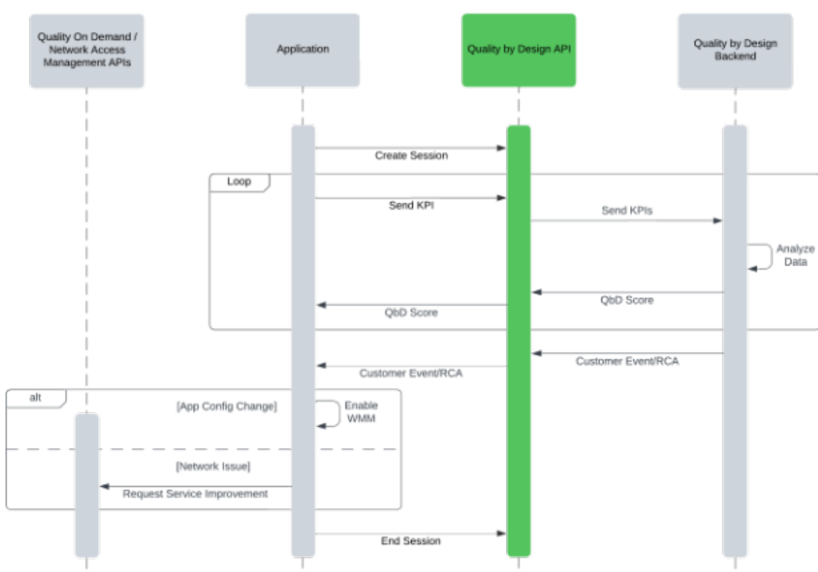

Figure 3 - QbD flow example

The QbD API is a pivotal element in enhancing the interaction between applications and network infrastructure, ensuring optimal performance and user experience. The following story details the API flow, emphasizing its capabilities and functionalities.

Creating a Session: The journey begins with the application establishing a session with the QbD API. This initial step sets the stage for ongoing communication and data exchange between the application and the network. By creating a session, the application registers itself and prepares to share vital performance data with the network operator.

Sending KPIs to the Network Operator: Once the session is active, the application continuously sends its KPIs to the QbD API. These KPIs include crucial metrics such as latency, jitter, packet loss, and bitrate. This real-time or near real-time data sharing enables the network operator to have a comprehensive view of the application's performance, providing true visibility into the user experience. Additionally, the API can take optional requirements from the application as percentiles for Quality of Outcome, allowing the application to specify performance goals, such as ensuring that 95% of packets arrive within a certain latency threshold.

Receiving the QbD Score: As the KPIs are transmitted, the QbD API forwards this data to the QbD Backend for analysis. The backend processes the KPIs, evaluating them against pre-defined Minimum Quality Threshold profiles. Based on this evaluation, the backend generates a QbD score, which categorizes the application's performance as Optimal, Suboptimal, or Unusable. This score is then communicated back to the application via the QbD API.

Receiving Customer Event/RCA: If the QbD score indicates suboptimal performance, the QbD API triggers a Customer Event, initiating a detailed root cause analysis (RCA). The network operator performs real-time telemetry collection to diagnose the underlying issues affecting the application's performance. The results of this RCA, along with the identified root causes, are sent back to the application.

Responding to RCA: Upon receiving the RCA, the application is equipped with actionable insights into the factors causing performance degradation. This information allows the application to respond effectively, making necessary adjustments to mitigate the identified issues. The ability to respond dynamically to network conditions ensures that the application maintains optimal performance and user experience.

Updating App Configuration: One of the critical responses to the RCA involves updating the application's configuration. For instance, if the RCA identifies network congestion as a root cause, the application might enable Wi-Fi Multimedia (WMM) to prioritize multimedia traffic. These configuration changes are implemented to adapt to the network conditions and improve performance.

Requesting Service Improvement: Beyond adjusting its configuration, the application can also request specific service improvements from the network operator. Using the Quality on Demand or Network Access Management APIs, the application can request enhancements such as Speed Boost or Packet Prioritization. These requests aim to optimize the network's performance to support the application's requirements.

Ending the Session: Once the application has implemented the necessary changes and the performance issues have been resolved, it can end the session with the QbD API. This step concludes the interaction, ensuring that the application's performance data is no longer transmitted, and the session is properly terminated.

4. Analysis

The QbD framework is crucial in detecting customer events based on application KPIs. The framework determines the quality and scoring using the Minimum Quality Thresholds. Once performance metrics fall below these thresholds, the Analysis Agent begins its work, correlating application KPIs with Network KPIs to pinpoint issues and provide actionable insights.

4.1. Quality Thresholds

Quality Thresholds are benchmarks predefined by network operators to establish acceptable performance levels for various KPIs. These thresholds are essential for maintaining service quality and ensuring a positive user experience. When performance metrics fall below these predefined levels, a customer event is triggered, initiating immediate investigation and remediation.

To provide a comprehensive evaluation, QbD categorizes applications into four main categories, each with its unique challenges related to performance maintenance:

Video Conferencing: This category focuses on maintaining high quality and low latency in video calls to ensure seamless communication. Challenges include dealing with variability in network conditions and ensuring synchronization between audio and video streams.

Live Video Streaming: Unlike on-demand streaming, live video streaming requires continuous, smooth data delivery to maintain high-quality playback. Challenges include buffering delays, which can cause noticeable interruptions in the stream, with a focus on minimizing buffering time and ensuring uninterrupted streaming even during network fluctuations.

Online Gaming: This category requires low latency and minimal packet loss for a seamless gaming experience. Challenges include managing the high demand for low-latency communication and frequent data exchange, which can be significantly affected by network congestion and instability.

Extended Reality (XR): This category demands high performance to avoid user discomfort in virtual and augmented reality applications. Even slight delays can cause motion sickness and disrupt the immersive experience, with a focus on maintaining high frame rates and minimal latency for fluid and realistic interaction.

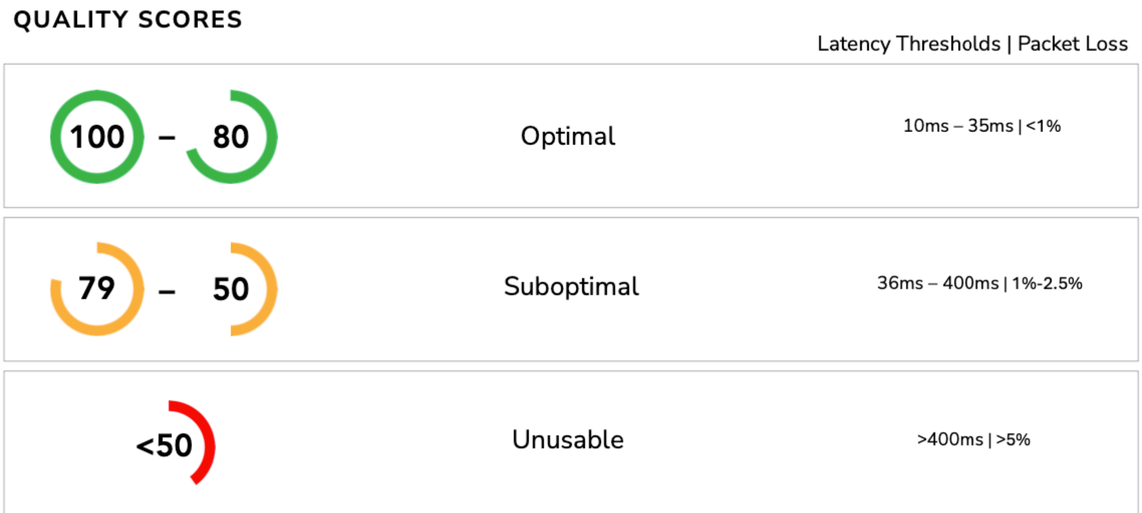

Figure 4 - Minimum quality thresholds for video conferencing scoring

When KPIs indicate that an application's performance has degraded to suboptimal or unusable levels, an event is triggered for further analysis and corrective action. This proactive approach minimizes disruptions to the user experience. Additionally, the QbD API allows applications to specify optional requirements as percentiles for Quality of Outcome. For example, an application might request that 95% of packets must arrive within a certain latency threshold to ensure a satisfactory user experience. Percentiles such as the 90th, 95th, and 99th are commonly used as they provide a higher confidence level in meeting performance objectives.

4.2. Analysis Agent

The Analysis Agent plays a key role in the QbD framework by processing and interpreting data collected from the network and applications. Its functionalities include:

Data Collection and Monitoring: Continuously collecting telemetry data from network and application KPIs to compute the QbD score, reflecting the current performance status.

Scoring and Threshold Comparison: Encapsulating application performance into a score categorizing it as optimal, suboptimal, or unusable. When the score drops below defined thresholds, a customer event is triggered for further analysis.

Root Cause Analysis (RCA): Initiating a root cause analysis upon detecting a customer event to identify underlying issues. It correlates application KPIs with network KPIs to pinpoint the exact cause of performance degradation. For example, if an application experiences high latency and packet loss, the agent examines network telemetry to identify issues such as Wi-Fi congestion or upstream noise.

Resolution Actions: Suggesting appropriate resolutions based on the root cause analysis. These might include enabling WMM to prioritize traffic or adjusting network profiles to mitigate noise interference, ensuring these actions are implemented promptly to restore optimal performance (Smith, 2024).

Continuous Improvement: Using historical data and trends to refine analysis and improve diagnostic accuracy over time. This continuous learning process helps anticipate potential issues and implement preemptive measures.

CableLabs' implementation of the Analysis Agent incorporates a sophisticated grading system that calculates a scalar value, or grade, based on telemetry data from the network and applications. This grading system uses various mathematical models to process input data and produce a score reflecting the health of the network. The structure of the grading system involves:

Data Normalization: Collecting and normalizing performance data, including latency, jitter, bandwidth, packet loss, etc., to match the effective region of the grading system.

Function Application: Using multiple functions to process each set of measurements. These functions can be linear, polynomial, or kernel regression mappings, combined to produce a final score.

Score Calculation: Deriving the final score from the weighted sum of the outputs of the applied functions.

Training and Calibration: Training the grading system using labeled data from various network scenarios, calibrating functions and weights to ensure accurate reflection of real-world network performance.

Additionally, if network operators choose, they can partner with third-party analysis vendors. These vendors can request specific requirements data, such as latency percentile information, to generate Quality of Outcome scores. This collaboration provides deeper insights and optimization recommendations, enhancing the end-user experience and overall network performance.

By leveraging these functionalities, the Analysis Agent transforms raw telemetry data into actionable insights, enabling rapid response to network impairments and enhancing overall service quality.

5. Application and Network KPIs

A core functionality of the QbD API Layer is enabling the sharing of real-time KPIs and other critical data from applications to the network. This continuous flow of information is vital for real-time performance monitoring and optimization, ensuring that both the application and the network can proactively respond to any issues that arise.

This section outlines the key KPIs used to measure and monitor the quality of service in network-based applications. For application developers, they offer guidance on optimizing user experience and implementing robust performance measures. For network operators, they offer critical insights for maintaining network reliability and efficiency, ultimately ensuring a high-quality service for end-users.

5.1. Application KPIs

Understanding the various KPIs is essential for both application developers and network operators to ensure optimal performance and user experience. These KPIs provide critical insights into network and application behavior, enabling proactive measures to enhance service quality. Here, we delve into the primary KPIs—Latency, Jitter, Packet Loss, and Bitrate—and their significance. While these are the foundational KPIs, the QbD API supports application developers in providing additional KPIs as needed to address specific performance metrics.

Latency (Round Trip Time): Latency, or Round-Trip Time (RTT), is the duration required to transmit data packets from the client to the server and back. High latency can occur if the server is located far from the user's location, leading to noticeable delays in data communication. For application developers, minimizing latency is crucial to ensure that applications respond promptly to user actions. By selecting optimal server locations and optimizing performance for various latencies, developers can enhance the application's responsiveness, providing a smoother user experience. On the other hand, network operators focus on reducing latency through network optimization and rapid issue resolution. By strategically placing servers and routing data efficiently, operators can significantly lower latency, thus improving overall network performance.

Jitter: Jitter refers to the variability in packet arrival time caused by network congestion, timing drift, or route changes. This variability can disrupt media quality, causing issues like video pixelation or audio distortions. Application developers must manage jitter effectively, particularly in real-time media applications such as video conferencing and online gaming. Implementing compensatory measures like jitter buffers helps maintain consistent media quality despite network conditions. Network operators, meanwhile, are responsible for monitoring and managing jitter to ensure stable data transmission intervals. By proactively addressing network congestion and timing inconsistencies, they can maintain a smooth data flow, thereby reducing disruptions in media quality.

Packet Loss: Packet loss measures the percentage of data packets discarded during transmission due to network congestion, hardware issues, or latency. This loss impacts media quality by causing gaps or distortions in transmitted data. Application developers must handle packet loss efficiently by deploying error correction techniques and optimizing protocols. These measures ensure that applications remain functional and maintain high quality even under poor network conditions. Network operators play a crucial role in identifying and mitigating the root causes of packet loss. By improving network infrastructure reliability and resolving hardware issues promptly, they can reduce packet loss, enhancing the overall quality of service.

Bitrate: Bitrate denotes the amount of data transmitted per second in audio or video streams and is a key determinant of media quality. Higher bitrates typically offer better quality but require more bandwidth. Application developers must balance bitrate to optimize media quality and bandwidth utilization through dynamic bitrate adaptation. This approach ensures that applications deliver high-quality media while efficiently using available network resources. Network operators, on the other hand, manage bandwidth usage to prevent congestion and ensure consistent service quality. By understanding and controlling bitrate demands, operators can maintain a balanced network load, supporting high-quality media transmission alongside other network services.

5.2. Network KPIs

Network KPIs are essential for monitoring and managing Wi-Fi and DOCSIS network performance, helping network operators and application developers diagnose and resolve issues efficiently. These metrics are collected at longer intervals under normal conditions but are gathered more frequently when a customer event is triggered. This approach ensures real-time or near-real-time telemetry data without adding complexity to the CPE when applications are performing optimally. Additionally, Network KPIs can be extended to include other access networks based on the requirements of the network operator. The following are a small sample of In-home Wi-Fi and underlying DOCSIS access network KPIs:

5.2.1. Wi-Fi KPIs

Airtime Utilization: Airtime utilization measures how much of the Wi-Fi channel's capacity is being used. High utilization can lead to network congestion and reduced performance. For application developers, understanding airtime utilization helps optimize application performance and user experience under varying network conditions. It guides developers in optimizing their applications to cope with high channel utilization. Network operators rely on airtime utilization metrics to monitor and manage Wi-Fi network load, taking proactive measures to balance the load and mitigate congestion.

Noise: Noise levels on each channel significantly impact Wi-Fi performance. High noise levels can cause interference and reduce data rates. Application developers can use noise level information to optimize app performance, ensuring users get a consistent experience even under varying noise conditions. Network operators use noise level metrics to identify and resolve interference issues. By monitoring noise levels, operators can select cleaner channels or take corrective actions to improve signal quality.

Bitrates: Bitrates represent the data rates at which devices are connected. This metric helps diagnose issues with network speed and performance. For application developers, knowing connection bitrates is crucial for adapting media quality or data transmission rates to match network capabilities. Network operators evaluate bitrate data to understand the network's actual performance, identify bottlenecks, and enhance network throughput by optimizing configurations.

Error and Retransmission Rates: Monitoring packet error rates and retransmission counts helps identify problems with signal quality, interference, and network congestion. Application developers can use this information to handle scenarios where high error and retransmission rates might degrade application performance, ensuring robust error correction mechanisms and efficient data transmission strategies. Network operators detect and diagnose network issues through these metrics. High error rates and retransmissions indicate areas needing optimization to reduce network load and enhance data delivery success rates.

5.2.2. DOCSIS KPIs

DOCSIS KPIs focus on monitoring the quality of DOCSIS networks, which are crucial for broadband connectivity. These metrics help network operators and application developers diagnose and resolve access network issues efficiently.

Upstream Received Modulation Error Ratio (RxMER): RxMER is a key metric indicating the quality of the received signal in DOCSIS systems. For application developers, understanding upstream signal quality is essential for robust error handling and fault-tolerant mechanisms. Network operators monitor RxMER to identify problems and optimize upstream channels, ensuring efficient fault detection and maintenance in the access network.

Forward Error Correction (FEC): FEC in DOCSIS systems employs LDPC codes for error correction. This metric is crucial for monitoring link quality and identifying bit errors. Application developers need to understand error correction capabilities to handle transmission errors effectively. Network operators use FEC metrics to maintain service reliability by identifying performance degradation, ensuring the robustness of the access network.

Signal to Noise Ratio (SNR): SNR measures the ratio of average signal power to average noise power, indicating communication quality. Application developers optimize performance based on signal quality, particularly in scenarios requiring high data integrity. Network operators ensure robust communication by selecting appropriate modulation schemes and maintaining service quality, especially under varying noise conditions.

Bit Error Rates (BER): BER measures the rate of errors in transmitted data, reflecting data transmission integrity. For application developers, designing error detection and correction mechanisms is critical for application reliability. Network operators monitor BER to minimize errors, addressing early signs of degradation and improving channel quality within the access network.

6. Conclusion

The QbD framework marks a significant leap forward in network and application performance management. By integrating real-time data collection, automated analysis, and proactive resolution mechanisms, QbD effectively addresses the shortcomings of traditional network management approaches. A key factor in the success of QbD is the adoption of its API by application developers, fostering seamless interaction between application developers and network operators.

Application developers can use the QbD API to share real-time KPIs with the network, which in turn provides a clear view of the user experience. This real-time data exchange allows for the swift detection of performance issues and the prompt initiation of corrective actions, often before users even notice a problem. This collaboration ensures that applications can dynamically adapt to changing network conditions, maintaining optimal performance and user satisfaction.

Quality Thresholds, set by network operators, define acceptable performance levels for various application categories. When performance metrics dip below these thresholds, the Analysis Agent steps in to correlate application KPIs with network KPIs, conducting a thorough root cause analysis to identify the exact issues. This proactive approach ensures that problems are promptly and appropriately addressed, minimizing disruptions and maintaining a high-quality user experience.

The Analysis Agent, with its advanced grading system, converts raw telemetry data into actionable insights. This continuous monitoring and real-time feedback loop enable rapid responses to network impairments, significantly enhancing overall service quality. Additionally, the QbD API's flexibility allows for the inclusion of additional KPIs tailored to specific application needs, offering a comprehensive and adaptable framework for performance optimization.

By facilitating real-time data sharing and automated responses, QbD provides a robust solution to the challenges faced by traditional network operators and application developers. This approach benefits network operators by offering deeper insights and optimization recommendations while empowering application developers to engage actively in network management. The collaboration fostered by QbD not only boosts performance but also significantly enhances user satisfaction, paving the way for more efficient and responsive network services.

In conclusion, the QbD framework ensures a seamless and high-quality user experience by unifying network and application performance management. This innovative approach represents a collaborative effort between network operators and application developers, ultimately benefiting end-users by delivering superior service quality and reliability. As the network landscape continues to evolve, QbD stands as a vital framework for maintaining optimal performance and customer experience.

Abbreviations

QbD	quality by design
NaaS	network as a service
API	application programming interface
KPIs	key performance indicators
CPE	customer premises equipment
DOCSIS	data over cable service interface specification
RxMER	received modulation error ratio
FEC	forward error correction
SNR	signal to noise ratio
BER	bit error rates
WMM	Wi-Fi multimedia

Bibliography & References

Fonte, P., & Khan, S. (2024, August 22). *Quality by Design Streamlines Network Management for Improved User Experience*. (CableLabs) Retrieved from https://www.cablelabs.com/blog/quality-by-design-streamlines-network-management

Smith, L. (2024, March 28). *Enriched Wi-Fi Performance Through Wi-Fi Multimedia*. (CableLabs) Retrieved from https://www.cablelabs.com/blog/quality-by-design-streamlines-network-management