NetQoS
Performance Experts

VoIP: Do You See What I'm Saying?

Managing VoIP Quality of Experience on Your Network

NetQoS, Inc.

Prologue

Network convergence offers the vision of a single network, carrying traffic from many different types of applications (voice, video, and data, to name a few). Many have been enticed by this vision. A large and increasing number of organizations have a Voiceover IP (VoIP) pilot or a full deployment project, either ongoing or planned for a future date. And it's almost impossible to buy a non-VoIP enabled phone system these days. In many cases, VoIP is just the entry point for rolling out new unified communications applications, such as real-time dashboards, video, and presence. Therefore, success is doubly critical for organizations deploying VoIP. Not only is phone service vital to the success of most enterprises, but if VoIP doesn't work well, the results are delays when deploying other, related applications, as well as long-term ramifications that are harder to predict.

A performance-first approach to network management is never more important than when deploying VoIP. In the early days of VoIP technology development, concern centered around call server uptime. Today, VoIP systems have matured to the level that you can move beyond making them work to making them work well. Now the focus has shifted to making VoIP perform well.

The task of making VoIP perform to the highest possible standard increasingly falls to you—the converged network manager. In the past, the network manager was responsible for the network only, and didn't need specialized application expertise. Convergence is changing all of that.

And network managers find that making the VoIP "application" perform well can be a challenge. Users have very high standards for VoIP performance. Enterprises deploying VoIP need to be able to see how their users' quality of experience relates to the quality of service provided by their network. The key question is raised in the title of this ebook: "Do You See What I'm Saying?" Or, in other words – do you have the visibility into the key performance metrics relevant to VoIP and the quality of experience seen by your users?

This ebook equips you, the network manager, with the knowledge you need to be successful in managing VoIP quality of experience on your network–the first step down the path toward truly unified communications. Understanding how VoIP works and what's important to ensure the performance of VoIP is a crucial part of this journey. This book therefore begins with a discussion of the protocols that are used for VoIP communications, then moves on to show you how those protocols affect performance. Once you understand some of the basic facts about VoIP and how it behaves on the network, we'll show you how you can use your knowledge of VoIP to keep it performing at the highest possible level.

How This Book Is Organized

VoIP: Do You See What I'm Saying?: Managing VoIP Quality of Experience on Your Network provides a comprehensive look at a complicated set of technologies. Keeping in mind the complexities you face as you prepare to deploy VoIP, complete a VoIP rollout, or fine-tune your deployment, we'll present the information in a structure that lets you build on the knowledge you already have.

Here is what each chapter covers:

Chapter 1 – Introduction to VoIP Protocols

- Basic VoIP hardware and terminology.
- What are the key VoIP call setup protocols?
- What are the key VoIP conversation protocols?
- VoIP traffic and how it differs from traditional networked applications.
- VoIP bandwidth considerations.

Chapter 2 – VoIP Call Setup Performance

- Key call setup metrics that you should be interested in.
- Call setup protocol details that affect performance.
- Network requirements for call setup.
- Troubleshooting call setup problems.

Chapter 3 – VoIP Call Quality Performance

- Call quality standards and why they are important.
- Key call quality metrics that you should be interested in.
- What impairments affect call quality?
- Where are quality problems likely to occur?

Chapter 4 – The Road to Unified Communications

- VoIP is a foundation for other communications technologies
- What are the new Unified Communications applications?
- How will Unified Communications affect network performance?

Introduction to VoIP Protocols

VoIP has a unique set of performance requirements that make it a challenge for any data network. Understanding the operation of core VoIP protocols is a first step in understanding the performance requirements that VoIP places on your network.

A VoIP phone call occurs in two stages:

- Call setup. This stage is required to set up everything needed to make the telephone connection between the person making the call (the caller) and the person receiving the call (the called party).

- The call itself. The audio conversation needs to be encoded and transmitted across the network.

The hardware required to make a VoIP phone call varies by vendor, but there are several common components that you will see on your network. Let's discuss some of these devices and terminology before diving into the protocols that they use to communicate.

Do you know this already? Many of you may have already deployed VoIP and are working with it every day. If you already have a good grasp on the basic components and protocols, you may want to skip to ahead to VoIP and Network Performance on page 13.

VoIP Hardware

A large number of new devices must be added to your network to support VoIP. Each of these devices will need an IP address and will communicate with other devices on the network. In terms of sheer numbers, IP phones will make up the largest group of new devices on your network.

IP Phones

IP phones connect to the network via an Ethernet interface. Many IP phones receive their power over the Ethernet connection from LAN switches that support the power over Ethernet (POE) protocols. This is similar to the way that Plain Old Telephony Service (POTS) phones can draw power from the Public Switched Telephone Network (PSTN) wiring and still function in the event of a power failure. Some higher-end IP phones have a single-port switch built into the phone, which allows a PC to be connected through the phone switch to the network. IP phones typically run an embedded operating system with its own TCP/IP stack for communications. And IP phones contain the software called a *codec* that is necessary to convert a voice conversation to IP packets.

Many IP phones can also run applications. Web services and protocols such as XML are used to turn IP phones into mini-computing devices.

In addition to "hard" IP phones, there are also "soft" phones, which consist of an application running on your desktop or laptop computer. Using an attached headset or the computer's own microphone and speakers, the softphone provides similar functionality to a traditional phone.

Both hard and soft IP phones register with a call server that provides the call setup functionality needed to make a phone call.

Call Servers

The call server serves as the core of a VoIP system. Its primary function is to provide call processing. Using a number of different protocols, a call server communicates with IP phones, voice gateways, and with other application servers. Sometimes called an IP PBX, the call server provides telephony features similar to a traditional PBX – features like hold and resume or transfer. IP phones register with their call server and communicate with it during the call setup phase of the phone call. The call server runs an operating system such as Microsoft Windows, Linux, or, in some cases, an embedded operating system, and it typically manages the configuration necessary to form the dial-plan for an enterprise. Call servers may be grouped or "clustered" together to provide increased fault-tolerance and load-balancing capabilities.

All the major VoIP vendors have different classes of call servers. Some are tailored to the needs of large enterprises, while others work well in SMB environments. Cisco CallManager is a popular example of an enterprise class call server.

Voice Gateways

The primary function of the voice gateway is to connect the IP network and the PSTN. Voice gateways must be able to communicate using many different protocols to route phone calls between the IP network and the PSTN. Gateways provide many different connection types to the PSTN, including analog and digital PRI. A T1 PRI connection is one of the most commonly deployed solutions. The basic T1 PRI configuration provides for 23 voice channels or connections to the PSTN. These channels are grouped together to form "trunks." A single voice gateway can support multiple T1 PRI connections.

Voice gateways may provide translation between the PSTN and the IP network. This translation, called *transcoding*, is the conversion between different encoding algorithms. The PSTN uses Pulse Code Modulation or PCMU encoding. This type of encoding, which is also known as G.711, is often used on VoIP networks as well. But if a different type of encoding is used, the gateway must transcode the communications.

Now that we've covered some of the basic VoIP hardware components, let's discuss the protocols that they use to send data across the network. We'll begin by looking at some of the protocols that are used in the call setup portion of a VoIP call.

Call Setup Protocols

The call setup stage of the VoIP call requires protocols that enable dial tone, number lookup, ringing, and busy signals. In other words, a lot has to happen before the call even occurs. Call setup protocols also handle things that happen after the call, including resource cleanup and statistical reporting.

Call setup protocols use either the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) to transfer data during the setup, call processing, and resource cleanup phases of a telephone call. Each protocol uses a well-known port or ports for communication. Some call setup protocols are used primarily for communication between endpoints (IP phones) and call servers, while others allow for communication between call servers and voice gateways handling calls to and from the PSTN.

Call setup messages are sent back and forth between the caller, called party, call server, and/or voice gateway. For calls that travel between the VoIP network and the PSTN, the call server converses with a voice gateway using the appropriate call setup protocol. These messages, which vary in size and number, handle functions like the mapping of phone numbers to IP addresses, generating dial tones and busy signals, ringing the called party, and hanging up.

Figure 1-1 shows a typical VoIP configuration with call setup flows among the different components.

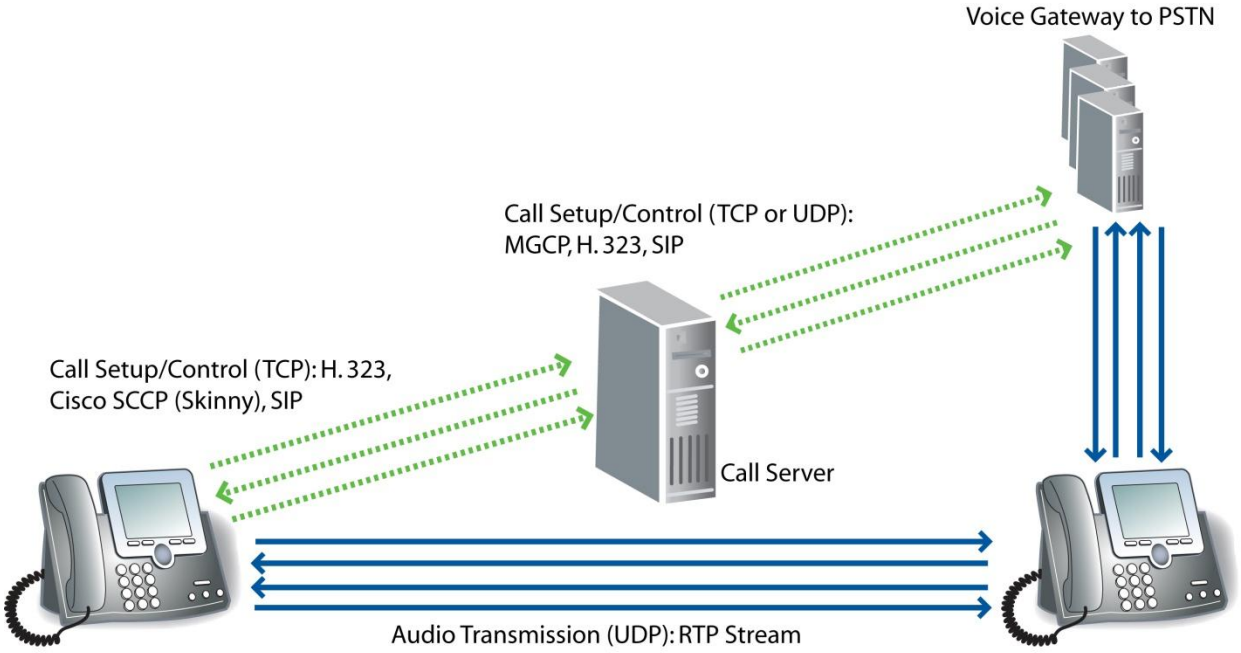

Figure 1-1 – Call setup protocols are used between endpoints, call servers, and voice gateways

Many different call setup protocols, some standardized and some proprietary, are in widespread use in VoIP deployments. We'll describe some of the most popular call setup protocols below:

H.323

The call setup protocol H.323 is standardized by the International Telecommunications Union (ITU). H.323 is somewhat of a legacy. It has been widely deployed and, was among the first call setup protocols used for VoIP calls. In a VoIP environment, H.323 is commonly used by voice gateways to connect the IP network to the PSTN. However, H.323 is actually a family of telephony-based standards that can be used by endpoints for both voice and video conferencing.

H.225 and H.245 are two of the protocols in the H.323 family that are involved in call setup. H.225 uses TCP and port 1720 for communications. H.245 uses TCP as well, but the ports used are dynamically negotiated during the H.225 phase of the setup. Setting up a call with H.323 can require many TCP flows and handshakes. Some implementations of the protocol provide a "fast start" capability to bypass some of the normal handshaking in an effort to speed up call setup performance. Be aware of this fact if you are investigating network performance issues during call setup.

Some additional configuration on the voice gateway is required for H.323 because the gateway maintains the information about how calls are routed. H.323 is a peer-to-peer protocol that lacks centralized configuration.

MGCP

The Media Gateway Control Protocol (MGCP) is another popular call setup protocol. It is standardized by the IETF in RFC 3435. MGCP differs from some of the other call setup protocols in that the endpoints typically do not use MGCP to control the phone call (although this is not prohibited by the protocol). Instead, MGCP is most often used to allow a call server to control a voice gateway connection to the PSTN.

MGCP sends messages between the gateway and call server over UDP port 2427. Because the call server is controlling the gateway, most of the call control intelligence and configuration resides in the call server. With MGCP, the call routing information is configured in the call server instead of in the gateway.

Q.931

The Q.931 protocol is not a canonical call setup protocol, but it supports call setup by defining Layer 3 signaling information for ISDN PRI interfaces. These interfaces are commonly installed in voice gateways to provide connectivity to the PSTN. In an ISDN PRI, a separate TCP session is used to communicate additional information between the voice gateway and call server. This connection is often referred to as "PRI Backhaul." Using the TCP protocol and port 2428, the PRI Backhaul connection provides the ISDN channel signaling information to the call server where the gateway is registered.

SIP

Session Initiation Protocol (SIP) is a call setup protocol developed by the IETF in RFC 3261 (and in many other RFCs too numerous to mention). SIP represents typical data-networking protocols: lightweight, relatively easy to understand and implement, ASCII-based. Many industry observers see it as the future standard for call setup, as vendors such as Cisco, Nortel, and Avaya are providing SIP phone/endpoint support. In addition, Microsoft recently announced the availability of their Office Communications Server, which uses SIP for call setup.

Although SIP can be carried over TCP or UDP, most implementations use TCP and port 5060. Secure SIP uses TCP and port 5061. SIP messages resemble HTTP: they are text-based and generally follow a request-response structure.

Another interesting aspect of SIP is its ability to connect IP networks to the PSTN or other IP networks. Carriers are offering SIP trunking packages to enable an enterprise to connect by means of a session border controller (SBC) device to the PSTN or to another IP network.

Our view is that SIP is a key enabler of unified communications. SIP can be used not only for voice call setup, but also as the protocol for video and instant messaging setup.

Proprietary Call Setup Protocols

In addition to the standardized call setup protocols discussed above, vendors have developed their own proprietary protocols. Here's a list of some of the top proprietary protocols that you may encounter:

Cisco Skinny Client Control Protocol (SCCP)

Cisco phones typically use the Skinny Client Control Protocol (SCCP) for call setup. SCCP, called "Skinny," provides a simple, lightweight call setup protocol for endpoints controlled by Cisco Unified CallManager (now being called Cisco Unified Communications Manager). Skinny passes messages using TCP and port 2000. It can be secured using Transport Layer Security (TLS). The secure version uses TCP and port 2443.

Nortel UNIStim

Nortel phones typically use the UNIStim protocol to communicate with call servers. UNIStim is a UDP-based request/ response protocol and has reliability built in at the application level. UNIStim uses port 5000 for communications.

Call Setup Ports

If you are looking at network traffic for diagnostic or monitoring purposes, or if you are maintaining firewall settings, it might be useful to identify the various VoIP call setup protocols by port number. Table 1-1 lists common call setup protocols and the transport protocol and port number that they use.

Call Setup Protocol	Transport Protocol	Port Number
SCCP	TCP	2000 2443 (Secure)
MGCP	UDP	2427
Q.931/Q.921 (PRI Backhaul)	TCP	2428
H.323	TCP	1720, Multiple dynamic
SIP	TCP or UDP	5060 5061 (Secure)
UNIStim	UDP	5000

Table 1-1 – Call setup transport protocols and ports

No single call setup protocol dominates the present VoIP market. The protocols discussed here (H.323, MGCP, Q.931, SIP, SCCP, and UNIStim) are all used by VoIP equipment in varying degrees. However, the trend is moving slowly toward SIP as the call setup protocol of choice for endpoints and gateway devices.

VoIP Conversation

To be heard by the called party, the conversation portion of a VoIP phone call needs to be converted into packet format, sent across the network, reassembled, and then converted from packet format back to voice. A number of different standards and protocols are available to enable the VoIP traffic to travel across the data network.

Codecs

Codecs encode and decode both ends of the phone conversation to allow the audio signal to be sent and received across the network. Different codecs are in common use, and they have different bandwidth requirements and different characteristics that can impact network performance.

Two of the commonly used codecs have been standardized by the ITU and use the names G.711 and G.729. The codec's job is to take the speech audio signal and transform it into a digital payload that can be sent across the data network. Some codecs, like the G.729 codec, for example, compress the data and require less bandwidth to send the conversation packets between the two phones involved in the conversation. However, the compression is usually "lossy," meaning that some degradation in quality occurs as a result of the compression process. Other codecs, like G.711, do not employ any compression schemes. The upside to the lack of compression is that no additional data loss occurs, but the tradeoff is a requirement for more network bandwidth.

Once the codec has the payload ready, it's up to another protocol, the Real-time Transport Protocol (RTP) to transfer the data to the receiver.

RTP

Unlike call setup protocols, where there is no dominant player, the single protocol that is used almost exclusively for the transfer of VoIP conversations is RTP. (Perhaps the only really notable exception is Skype, which uses proprietary protocols that are beyond the scope of this ebook.) RTP is a protocol that is defined by the IETF in RFC 1889. An application protocol that rides on top of UDP, RTP is widely used for streaming audio and video. Applications that need real-time performance use RTP to send data in one direction with no acknowledgements.

Because a VoIP call is bidirectional, involving two talkers, two RTP streams carry the conversation, one traveling in each direction between the caller and called party. These two unicast streams are referred to as *call legs*. Quality statistics are provided for each call leg and represent the quality of the RTP audio steam at the receiving end.

RTP uses even-numbered UDP port numbers above 16384. Each unidirectional RTP stream typically uses the same destination port. The actual port number used is determined during the call setup exchange between the IP phones. The path that these RTP streams take through the network and the impairments encountered along the way are important factors in determining

the quality of the voice conversations. Figure 1-2 shows the two RTP streams that make up a VoIP call.

Figure 1-2 – Two unicast RTP streams make up a typical VoIP phone call

RTP is a connectionless protocol like UDP, meaning the datagram is sent without acknowledgement and is not retransmitted if lost. RTP datagrams are composed of a payload and header information. The codec on the IP phone creates these datagrams by placing a 12-byte header inside the UDP datagram with a payload containing data produced by the codec. RTP datagrams are typically small relative to other data packets. For some codecs, the payload may be as little as 20 bytes. Several fields in the RTP header are used by the IP phone to help determine quality statistics. It is helpful to understand the structure of the RTP datagrams, which transport the encoded voice information. Figure 1-3 shows the composition of the RTP datagram and header information.

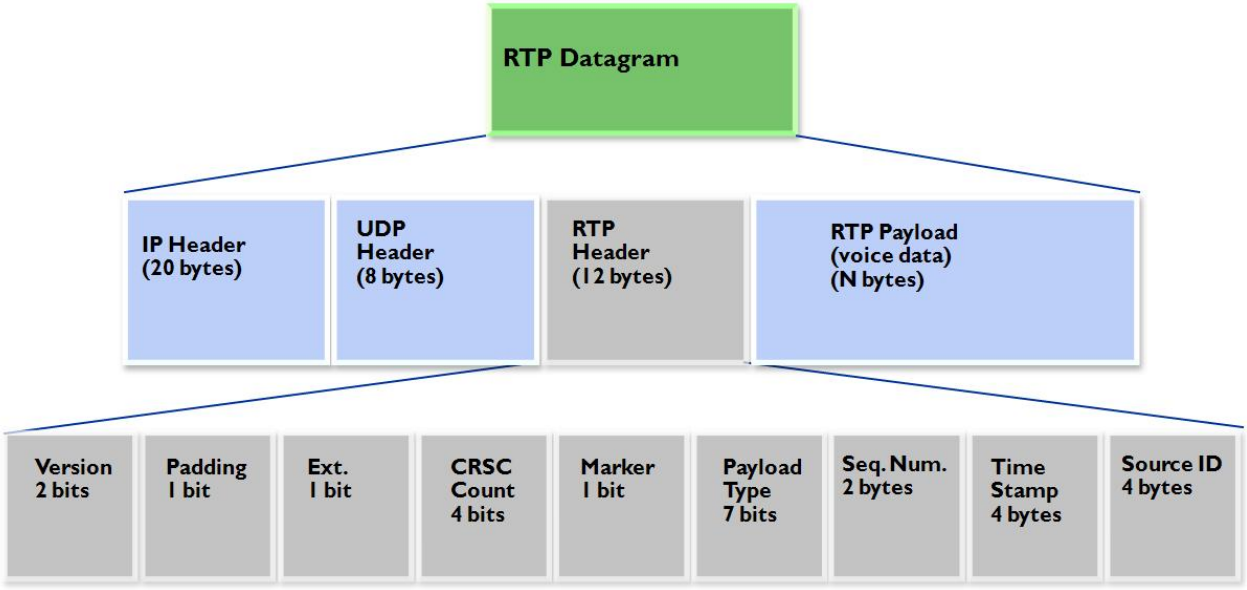

Figure 1-3 - RTP Header Format

As IP phones exchange data during a VoIP phone conversation, the sending phone fills in each datagram header, which contains several important fields:

- **RTP Payload Type** – Specifies the codec that is used. The codec is important so that the receiver knows to apply the same codec to decode the data payload.

- **Sequence Number** – The receiving side uses this field to aid in the reassembly of the data stream. The sequence number is also useful to help determine quality metrics, such as lost, out-of-order, and duplicate RTP datagrams.
- **Time Stamp** – The RTP sender normally sends datagrams at a constant rate; for VoIP, this is usually every 20 or 30 milliseconds. Network delays and congestion can cause variations in the arrival time of datagrams in the same stream, a condition referred to as *jitter*. The timestamp is set by the sender of the datagram, and the receiver can use this information to determine the jitter. If the jitter is too great, the receiver may have to discard the datagram.

RTP has an optional profile called Secure RTP (SRTP). SRTP was defined in RFC3711 and provides data encryption and message authentication. Anytime encryption is used, it's important to understand the performance impact. With recent advances in phone hardware and firmware, encryption overhead is now minimal--sometimes, it's even measured in microseconds. However, for devices like gateways that are terminating a large number of media streams, the performance degradation can be high. Usually, there is a tradeoff in the number of simultaneous streams with encryption vs. without encryption. The number may be lower with encryption enabled because of the processing overhead associated with the encrypted streams.

RTCP

The Real-time Transport Control Protocol (RTCP) is a companion protocol to RTP that provides statistics about the RTP flow for a given call. Originally defined in RFC1889 along with RTP, the latest definition of RTCP is provided in RFC3550. RTCP defines sender and receiver packet types. In most cases, a VoIP phone would use the sender report because it performs both sending and receiving functions during the two-way conversation.

RTCP uses odd-numbered UDP ports. Often a value of 1 is added to the port number of the RTP stream to get the RTCP port number. For example, if the RTP stream is using port 16384, RTCP would use port 16385.

The statistics that RTCP provides are important to help determine the quality of the call and any underlying network metrics that contributed to the quality rating. Figure 1-4 shows the quality information available in the RTCP packet.

RTCP Metrics (condensed)		
Jitter	Lost packets (fraction and cumulative number)	NTP timestamp

Figure 1-4 – The RTCP packet contains a limited amount of quality information

The RTCP packet contains quality statistics for packet loss (fraction and cumulative number), jitter, and a timestamp. You can use the NTP timestamp to compute a round-trip delay value between the sender and receiver. The jitter value is calculated according to RFC1889. This calculation is based on the packet delay variation (D):

$$D(i,j) = (Rj - Ri) - (Sj - Si) = (Rj - Sj) - (Ri - Si)$$

The jitter value (J) is then calculated as follows:

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16$$

This calculation has the potential to hide jitter problems because it is an average derived from the last 16 packets. As an example, let's say that you see 14 packets with 0 ms of delay variation, followed by 2 packets with 80 ms of delay variation. The above calculation would yield:

J (for packet 15) = 0 ms + (80 ms – 0 ms) /16 = 5 ms

J (for packet 16) = 5 ms + (80 ms – 5 ms)/16 = 9.7 ms

After 16 packets, the average jitter value is 9.7 ms, which seems like a very low amount of jitter; however, 2 packets actually had very high delay variation and were likely discarded by the jitter buffer.

Additional work has been done to address some of the limitations in the quality metrics provided by RTCP. The RTCP-XR standard was created for this purpose.

RTCP-XR

RTCP-XR, defined in RFC3611, extends the RTCP report to provide additional quality metrics that are important in understanding the user's experience with a phone call. RTCP-XR defines the VoIP Metrics block, which contains the additional information. Figure 1-5 shows a condensed version of VoIP Metrics with some of the key quality metrics.

RTCP-XR Metrics (condensed)		
Loss rate	Discard rate	Burst density/duration
Gap density/duration	Round-trip delay	End system delay
Signal/noise level	Echo return loss	R-factor and MOS

Figure 1-5 – The RTCP-XR packet contains much more quality information than RTCP

As you'll notice, the RTCP-XR specification contains many more quality-related metrics than RTCP reports. Some of the key metrics and their definitions:

Loss rate – The percentage of packets that are lost by the network. This is the traditional network packet loss metric.

Discard rate – The percentage of packets that are received, but discarded due to early or late arrival. This is the jitter buffer loss metric. We'll explain jitter and jitter buffers more in Chapter 3, where we discuss call quality metrics.

Burst/gap density/duration – These metrics provide an indication of the burstiness of the data loss. This is important because bursts have a greater impact on quality than random packet loss.

Signal/noise level/echo return loss – These metrics provide some quality data for calls that go through a voice gateway to the PSTN.

R factor and MOS – Provides an indication of the quality of experience for the call. The R factor is defined in ITU-T G.107. MOS is the mean opinion score, an estimation of the audio quality of the RTP stream. In RTCP-XR, the MOS is provided in terms of listening quality (LQ) and conversational quality (CQ). MOS-CQ takes into account delay that could impact the conversation quality, whereas MOS-LQ does not factor in delay.

We'll discuss all of the above metrics in more depth later in this ebook.

Proprietary Call Quality Reporting

In addition to the RTCP standards for reporting quality metrics, some vendors may send metrics back as part of the proprietary call setup protocols that they use. These metrics could be as basic as packet loss and jitter, or they could contain enhanced quality of experience information, such as a MOS value. We won't cover these methods in this ebook, but be aware that vendor-proprietary call setup protocols have ways of returning metrics related to the quality of the call.

Now that we've covered the key protocols used by VoIP, let's talk about how VoIP relates to network performance.

VoIP and Network Performance

Because phone calls have traditionally traveled over a separate network, the PSTN, the term *convergence* is often used to describe the process of putting VoIP traffic on the data network, allowing the two types of networks to come together. You are probably well aware that anytime you add a new application to the network, unexpected results often ensue. In this section, we'll discuss the many reasons why VoIP is likely to affect the performance of your existing applications, how those applications behave in comparison to VoIP, and the unique characteristics of VoIP that place new demands on the network.

Let's first talk about some characteristics of non-VoIP applications that can affect their performance on the network.

Traditional Networked Application Performance

There's a group of applications running on your network that can be termed "traditional" networked applications. These applications are usually critical to your business: email, ERP, Web, and database. Many of these applications are client-server based, and others are multi-tiered. They have some common qualities when it comes to their network performance requirements:

- **Traditional networked applications use the TCP protocol.** They use the TCP protocol because it is reliable and connection-oriented. TCP guarantees that any lost packets get retransmitted, and that you receive the data that you sent in the order in which you sent it. A key characteristic of TCP applications is that they are adaptive, meaning they adapt to

changing network conditions. TCP provides a data window that can expand to allow more data to be sent, and contract to throttle back the amount of data that can be sent.

- **Traditional networked applications can tolerate packet loss and delay**. If a packet is lost, it is retransmitted. The application slows down, but for the most part, it is still usable. And if a packet is delayed, you may not even notice it. Consider email: If an email message gets delayed by 30 seconds, do you even know about the delay? Delay, or latency, is a key network performance metric for all types of networked applications. It's important to realize the impact that latency can have on performance. A latency of 100 milliseconds may seem insignificant, but if a transaction is composed of 100 requests/responses, that's 10 seconds of delay.
- **Traditional networked applications have a client-server structure and are transaction-oriented**. A Web browser client requests a page from a Web server. A database financial application updates a record on the finance database server. There's an inherent many-to-1 (many clients to one server) traffic flow model here. The application *transactions* are made up of requests and responses. A simple transaction could be a single request and response, similar to those mentioned above. A more complex transaction could contain many short request-and-response flows, or, in the case of a file transfer, a transaction could consist of a single transmission of a large amount of data. The amount of time it takes to perform a transaction is known as the transaction time. The transaction time metric gives the end user some indication of how responsive the application feels.
- **Traditional networked applications are often written by programmers who aren't networking experts.** Why does this matter? Networked applications can be optimized or de-optimized to transfer data. For example, a word-processing application can generate hundreds or even thousands of network flows for a simple File Open operation. If you are performing this operation over a network with high latency, you will be waiting for the file to load for a long time. Networked application programmers may not be up on the latest workings of the TCP Nagle or delayed-ACK algorithms. Adding to this optimization problem is the release of new operating systems like Windows Vista, with their corresponding tweaks to the TCP stack.

So how do you measure the performance of a traditional networked application? You can look at several key performance measurements that are common to all the TCP applications discussed above. Table 1-2 lists performance metrics for traditional networked applications.

Performance Metric	Description
Transaction Time or Response Time	Time from when a client sends a request and when the client receives the response. A reflection of the user experience for this particular application or network. Measured in milliseconds.
Network round-trip time	Time it takes for a packet to travel between client and server on a network. Measured in milliseconds.
Network connection time	Time it takes a client to confirm a server's

	connection acknowledgement. For applications that use many, short-lived connections, this metric can be an important indicator of latency (and thus, of application performance). Measured in milliseconds.
Retransmission delay	The delay in network round-trip time caused by retransmissions. Measured in milliseconds.
Throughput/Goodput	Throughput is the number of bytes transmitted divided by the amount of time required to transmit. Goodput is the same calculation, but it subtracts the retransmissions. Goodput provides a good indication of the client experience and is an important metric for applications that need to transfer a lot of data. Measured in kilobits or megabits per second.

Table 1-2 – Traditional networked application performance metrics

Underlying these high-level application metrics are the basic network performance metrics of **packet loss** and **latency**. If your network has high packet loss, applications will have to retransmit data, thus affecting performance and adding delay. Under conditions of high latency, applications that use many transactions are the most severely impacted.

Many times, network administrators attempt to throw more bandwidth at network performance problems, only to find out that the underlying issue was latency – which the bandwidth doesn't necessarily address. Over a decade ago, Stuart Cheshire wrote an interesting article titled, "It's the Latency, Stupid." Cheshire points out the problem with latency using the modem technology of the time. The networking technology has changed since then, but the latency problem still exists. As mentioned earlier, 100 ms of latency may not sound like much; however, couple that with hundreds of back-and-forth transactions and it can add up to many seconds of delay.

Networks have been tuned over the years to handle traditional networked applications like email, Web, and other business-critical applications with excellent performance. These and other transaction-oriented applications rely on TCP to handle any network issues like congestion, delay, or lost or out-of-order packets. Unfortunately, this type of handling is radically ill-suited for the real-time performance characteristics of VoIP. If an email message is delayed by a few seconds, it is likely that no one will notice. However, if part of your voice conversation is delayed by a few seconds, you may hang up the phone. Let's take a look at why VoIP traffic has different performance requirements.

What Makes VoIP Traffic Different?

VoIP comprises a diverse collection of often new technologies that are helping transform the way we communicate, at work and at home. And it has a unique set of performance requirements that make it a challenge for the infrastructure supporting any data network. Understanding what makes VoIP different from the other traffic running on your network is the

first step in understanding what's needed for optimal network performance when VoIP traffic is added to the mix on your data network.

Real-time VoIP Characteristics

Unlike the data traffic generated by traditional TCP-based network applications, VoIP traffic has real-time characteristics.

- **VoIP applications use the RTP protocol.** VoIP applications must send data immediately and don't have time to wait for retransmissions. That's why they use RTP. As we explained above, RTP rides on top of UDP. UDP is a connectionless protocol, which means that it doesn't include a mechanism for retransmission or reordering of data. Instead of sending a packet and waiting to make sure it was received before sending another packet, RTP applications typically send packets at a fixed rate to avoid congestion. Not only do they not resend data when loss occurs—they don't even slow down. If an RTP packet is lost, it's *lost*, and there's no chance to retransmit it. And if a group of consecutive packets is lost, entire portions of the conversation can be wiped out.
- **VoIP applications send small data packets.** VoIP data packets are typically small in comparison to other types of data packets. As explained earlier in this chapter, VoIP codecs are tasked with producing data packets from a voice conversation. The payload size for these packets can be anywhere from 20 bytes (for G.729) at the low end up to 160 bytes (for G.711) at the high end. Using fixed data rates, the codec typically sends these packets out every 20 milliseconds. Without proper quality of service (QoS) mechanisms active on the network, these small VoIP packets may get blocked behind a larger data packet waiting to be sent on a slower-speed WAN link.
- **VoIP applications have strict delay requirements.** Because voice conversations are interactive in nature, they are not at all tolerant of long delays. If delay is significant enough (usually more than 150 milliseconds), the phone conversation may feel as if you are talking over walkie-talkies. A VoIP packet may be delayed for several reasons:
  - Packetization delay occurs when the speech is encoded by the codec and the packet is created.
  - Network delay includes time to transmit the packets, buffer and queue them if needed, and move the packets from hop to hop through the network.
  - Jitter buffer delay is added by the receiver when buffering packets to reduce the impact of variations in packet arrival times.

  When these factors are combined, delay can easily become significant. As a result, minimizing the overall delay is an important consideration for maximizing the quality of interactive VoIP calls.
- **VoIP applications require QoS.** The PSTN did a great job of guaranteeing the resources needed for a call and maintaining them throughout that call. After the initial signaling, a path through the PSTN and all associated resources is reserved for each call. By contrast, an IP network does not usually operate this way. In an IP network, the resources are shared by many different users. Congestion at a router caused by another user's large file download can impair the quality of your VoIP call. There is a QoS mechanism known as RSVP that can provide guaranteed resource reservation, but the result is such high overhead that it is not widely implemented. Nevertheless, a QoS scheme that provides prioritization for VoIP traffic is *required* for every VoIP deployment.

The underlying network metrics that affect VoIP call quality are **packet loss**, **jitter**, and **latency**. We will discuss these metrics in Chapter 3. Because of the unique characteristics of VoIP traffic, a new set of demands is placed on data networks. Realize that, if you have good network performance for traditional networked applications, it is not a guarantee that performance will be good for VoIP applications. The real-time characteristics of voice create very strict requirements for network performance. There is some balance required of the network.

An inappropriately configured network could very easily cause VoIP phone calls to have poor quality: calls could be delayed, dropped, or they could just plain sound bad. These call quality issues would quickly affect the end-user experience because telephone usage is very user-intensive. You could end up getting a lot of calls to the help desk to fix the problem--or at least, you would if the phones were working. However, you would not want to make your VoIP traffic perform great at the expense of other applications that are just as critical to your business. When deploying VoIP, you also need to consider the impact that VoIP can have on other applications.

The Impact of VoIP on Other Applications

Just as with any new application deployment, when you add VoIP traffic to the network, you need to take a careful look at the performance, and not only the performance of the VoIP traffic. You need to carefully measure the impact the VoIP traffic is having on your other enterprise applications. And as cool new, integrated Unified Communications applications roll out, providing such services as instant messaging, voice mail, video, and conferencing, you have to consider the network performance characteristics of those applications as well. Good QoS policies must be in place to give priority to VoIP traffic over the TCP applications, which aren't very delay-sensitive.

Jim McQuaid (Sr. Product Manager at NetQoS), recently did an interesting video blog post titled "Nice Guys Finish Last – The impact of voice/video on data applications." In this video whiteboard session, Jim explains a key point: That the differences in the UDP and TCP protocols can have an interesting impact on the performance of all applications when VoIP traffic is added to the network mix.

Specifically, the adaptive nature of TCP can result in cases where TCP application performance degrades when faced with additional UDP traffic. Within conditions of congestion, TCP plays nice and backs off: the window sizes shrink, and less data is sent and received. But UDP, and VoIP in particular, continue to send data with no regard to congestion. Even though VoIP applications send data at a fixed, relatively slow rate, if enough calls are present, each with two unidirectional streams, the VoIP UDP traffic might create congestion and crowd out the TCP application traffic, creating longer transaction times and impairing the performance of TCP applications.

The goal in managing a converged network is to tune it so that many types of application data traffic can coexist and perform well. QoS mechanisms are necessary, as well as visibility into the underlying metrics that affect end-user quality of experience. And taking a step back from those metrics, knowledge of overall network performance is another significant factor in the success of a VoIP deployment. Figure 1-6 shows an example of how key metrics like transaction time can be baselined. Comparing these baseline measurements before and after

VoIP traffic is added to the network can give you an indication of the performance impact of VoIP traffic on other applications.

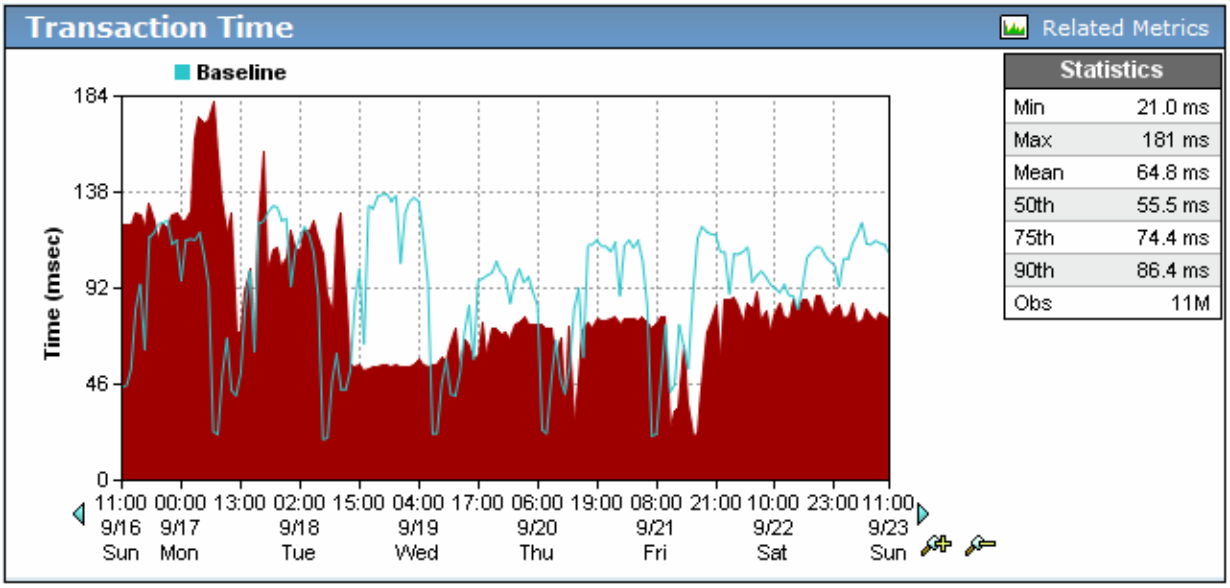

Figure 1-6 – Baseline transaction time for critical applications.

QoS is important, but you shouldn't ignore the possibility that congestion might become an issue. VoIP traffic is going to consume additional bandwidth--maybe more than you expect. Despite the smaller packet size, a VoIP call using the G.711 codec requires ~87 kbps of bandwidth. This is due to header overhead, which belies the fact that the data rate for G.711 is 64 kbps. And every call has two call legs. It's therefore important to consider the bandwidth usage of VoIP, especially if you have VoIP traffic traversing WAN links.

VoIP Bandwidth Considerations

How much bandwidth is consumed by VoIP? How much is consumed by other applications? Those are good questions to answer. Let's start with VoIP. Earlier in this chapter, we discussed the RTP protocol and the contents of the RTP header. While the RTP header is required to support the real-time nature of the protocol, the accumulation of RTP headers can add a lot of overhead, especially considering the relatively small size of VoIP codec payloads.

Here's an example of the header overhead of RTP, UDP, and IP headers on each VoIP packet:

RTP (12 bytes) + UDP (8 bytes) + IP (20 bytes) = 40 bytes

The typical payload size for G.711 is 160 bytes, which means that RTP, UDP, and IP headers are adding 25% overhead. When you add in layer 2 headers dependent on the physical media, this overhead increases even more.

Real Bandwidth Required by Each Codec

Codec advancements have led to a new generation of codecs. G.711 and G.729 are still the most widely used codecs, but some of the newcomers tout better quality for less bandwidth.

Some of the new codecs, such as RTAudio, provided with Microsoft Office Communications Server, are variable-rate codecs, meaning they can adjust the data rate based on underlying network conditions. With variable-rate codecs, more bandwidth is used to achieve better quality. In conditions of network congestion, the codec may throttle back the sending rate to use less bandwidth. The Bandwidth Usage column in Table 1-3 shows the actual bandwidth usage you can expect for some common codecs.

Codec	Data Rate	Bandwidth Usage
G.711u	64.0 kbps	87.2 kbps
G.711a	64.0 kbps	87.2 kbps
G.729	8.0 kbps	31.2 kbps
RTAudio	29.0 kbps	45.0 - 74.0 kbps

Table 1-3 – Bandwidth required by different codecs.

It's not too difficult to do the math and calculate the approximate amount of bandwith consumed by VoIP, but this number could be ever-changing as calls come and go. Is there a way to determine this from the network? Yes! There's a technology called Cisco IOS® NetFlow, or more generally, IPFIX, which allows network devices (routers and switches) to collect information about protocol usage and bandwidth consumption. Figure 1-7 shows an example of the protocol data available from NetFlow.

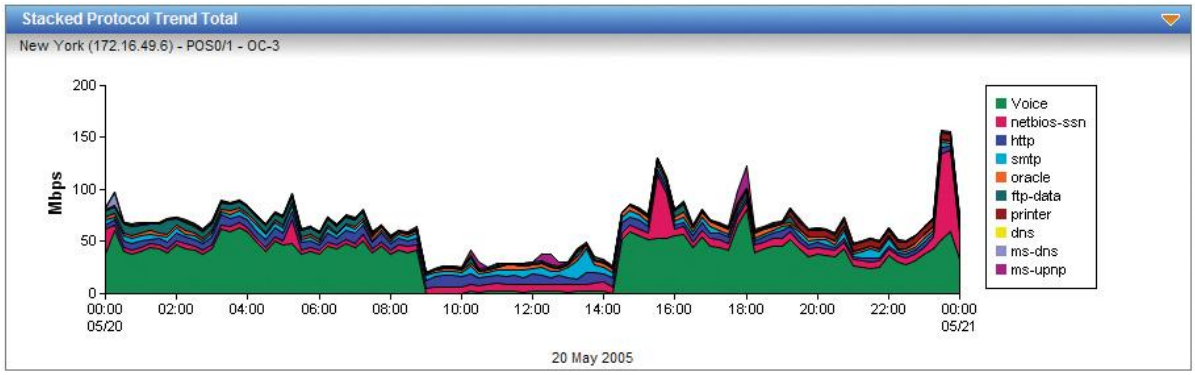

Figure 1-7 – Bandwidth breakout by protocol, including VoIP traffic

NetFlow plays a key part of the management picture by providing the data to perform link traffic analysis. You can see whether, for example, the VoIP traffic is being crowded out by the Web traffic, or the opposite case as well. NetFlow can also provide information about the QoS markings for different protocols. This information is useful in troubleshooting cases where the

VoIP traffic is not marked with the correct QoS bit settings. VoIP bandwidth is something to consider as part of any performance management strategy.

VoIP Network Performance Visibility

Technology enables a lot of the things we do every day of our lives, and it also adds a little fun. You can get away to a far-off place in a matter of hours as long as it has an airport. But technology always has limitations. Take flying, for example: Visibility is a key requirement for operating an airplane. No planes can take off if the pilot can't see well enough. Even in a high-tech airport, you still need to be able to see what's going on around you. If you can't see, you can't fly.

The same analogy applies to network performance, and to VoIP performance especially. If you don't have good visibility into what is happening on your network, you are likely to be grounded before your deployment really takes off. On a converged network, visibility into VoIP network performance is a key requirement for effective network management. Do you see what I'm saying?

At the end of the day, VoIP will provide a strong test for the underlying network. Even if you have a high-performing network, some tuning will almost certainly be needed to support VoIP at the level required for high-quality calls.

Chapter Summary

In this chapter, we've introduced the VoIP protocols and the basics of VoIP network performance. Most of the time, we don't think too much about the underlying protocols when we use traditional TCP network applications, but in the case of VoIP, the protocols themselves have some interesting characteristics that can affect the performance.

Let's briefly summarize a recommendation for how to handle VoIP traffic on your network. You can't treat voice traffic the same as traditional networked applications. Why?

- Different protocols – RTP vs. TCP
- Different packet sizes – small VoIP vs. large data
- Different QoS requirements – delay sensitive for VoIP
- Different data rates – relatively low, constant for VoIP; elastic for TCP
- Different user expectations – PSTN-shaped, high quality for voice, vs. data, where delays might not be noticeable

The quality of experience that your users perceive from a VoIP system is largely shaped by their perception of the availability and call quality that the system provides. In terms of the phone system, availability can be summarized with the following basic criteria:

- Do you get a dial tone when you pick up the phone?
- Does the call connect within a reasonable amount of time?

In the next chapter of this ebook, we'll examine these questions and how they are related to VoIP call setup performance.

About NetQoS VoIP Monitor

NetQoS VoIP Monitor is a network-based call setup and call quality monitoring product that tracks the call quality users experience, provides alerts on call performance problems, and isolates performance issues to speed troubleshooting and MTTR. NetQoS VoIP Monitor is integrated with the NetQoS Performance Center so you can monitor VoIP quality of experience while managing network quality of service, from a single, Web-based console. With NetQoS VoIP Monitor, you can assess the performance of your Cisco® Unified Communications Manager (CallManager) IP PBX by tracking, evaluating, and reporting on key metrics without deploying server agents or probes. For every call, NetQoS VoIP Monitor reports on user call quality and the underlying network performance metrics associated with it. NetQoS VoIP Monitor also breaks out performance data from the IP and Public Switched Telephone Network (PSTN) legs of calls that pass through voice gateways traveling to endpoints in the PSTN. This data lets you know if it is your network that is responsible for less-than-optimal call quality.

About NetQoS

NetQoS is the fastest growing network performance management products and services provider. NetQoS has enabled hundreds of the world's largest organizations to take a Performance First approach to network management—the new vanguard in ensuring optimal application delivery across the WAN. By focusing on the performance of key applications running over the network and identifying where there is opportunity for improvement, IT organizations can make more informed infrastructure investments and resolve problems that impact the business. Today, NetQoS is the only vendor that can provide global visibility for the world's largest enterprises into all key metrics necessary to take a Performance First management approach. More information is available at **www.netqos.com**.

NetQoS Global Headquarters
5001 Plaza On The Lake
Austin, TX 78746
Phone: 512.407.9443
Toll-Free: 877.835.9575
Fax: 512.407.8629

NetQoS EMEA
1650 Arlington Business Park
Theale Reading, RG7 4SA
Phone: + 44 (0) 118 929 8032
Fax: + 44 (0) 118 929 8033

NetQoS APAC
NetQoS Singapore Representative Office
Level 21, Centennial Tower
3 Temasek Ave., Singapore 039190
Phone: + 65 6549 7476
Fax: + 65 6549 7001

Website: www.netqos.com
E-mail: sales@netqos.com