Evaluating Data Networks for Voice Readiness

by John Q. Walker and Jeff Hicks
NetIQ Corporation

Contents

Our focus is on organizations that deploy and troubleshoot voice over IP (VoIP) networks. These groups may be professional services or pre- and post-sales engineering groups. A big problem with VoIP is the quality of calls, which is most affected by convergence with existing data network traffic. In most scenarios, the data network will require tuning in order to achieve acceptable quality for voice traffic. When changes are made to the network, network administrators will need a way to ensure that they will continue to get the same or better quality. This paper describes practical steps for building an assessment of whether a data network is ready for VoIP. Our software, *Chariot*, is designed so that personnel with little training and no additional VoIP equipment can quickly make useful VoIP-readiness assessments.

Introduction

The opportunity to use data networks for telephone conversations is appealing. The technology to do this is commonly known as Voice over IP or IP telephony, and has become widely available during the past few years [1]. Is the data network before you ready for this new type of traffic, that is, will the IP phone users be satisfied with the quality of their telephone conversations?

Our focus during the past six years has been building software to measure network performance. We have worked with the industry leaders in numerous deployments of VoIP, and learned many lessons that can be applied by anyone considering VoIP. This paper discusses practical steps for determining whether an existing network is ready for VoIP.

The steps are easy to follow, and they're all done before investing in any voice equipment or doing any deployment. You create a test that simulates the data flows of VoIP traffic, you run the test periodically on the live network while capturing appropriate network performance measurements, and then you analyze these measurements, comparing your results against values which indicate acceptable performance. These steps show the suitability of the network to support a single telephone conversation. If one conversation can be supported with good quality, you can repeat these steps with additional traffic to find the network's capacity to support multiple conversations.

The parameters involved in making voice work well in a network are different from those involved with making traditional business transactions work well. With the VoIP-readiness assessment we describe here, you can determine the status of your real network without any voice hardware. You can discover whether the network's ready – and if it's not, make it ready – without actually purchasing and deploying call gateways, IP PBXes, IP phones, and so on.

Determining Readiness

Voice quality testing has traditionally been subjective, which involves picking up the phone and listening to the quality of the voice. The leading subjective measurement of voice quality is MOS (Mean Opinion Score) as defined in the ITU (International Telecommunications Union) recommendation P.800 [2]. However, asking people to listen to calls over and over can be difficult and expensive to set up and execute.

There has been a lot of work recently to provide an objective measurement. Standards have developed such as the E-model, PSQM (Perceptual Speech Quality Measure), and PAMS (Perceptual Analysis Measurement System). The ITU recommendation G.107 [3,4] introduced the E-model. The E-model provides an "R factor," derived from various delays and equipment impairment factors. Once an R factor is obtained there is a defined mapping to an estimated MOS score. The ITU recommendation P.861 [5] introduced the PSQM measurement, which provides another objective measurement of voice quality. British Telecom introduced PAMS, which bears some similarity to PSQM. The PSQM and PAMS measurements send a reference signal through the network and then compare the reference signal with the signal that is received on the other end of the network. Several traditional voice measurement tools have implemented PSQM and PAMS measurements.

We used a modified form of the E-model, as documented in ITU G.107. This can easily be converted to an estimated MOS score. We set up tests that generate VoIP traffic between two points in a data network. A test runs periodically for a day, say once every 15 minutes for 24 hours. Each time a test is run, measurements are collected on the one-way delay time, the number of packets lost, the number of consecutive packets lost (known as burstiness), and the amount of variability in the arrival time of the packets (known as jitter). These measurements (and the way they vary over the course of a day) capture the important aspects of voice quality: how the two people at the two telephones perceive the quality of a voice conversation. Our modification of the original E-model algorithm takes into consideration jitter, packet loss, burstiness, and the codec.

Creating the Test

Implementing a telephone conversation on a data network involves doing the call setup – that is, doing the equivalent of getting a dialtone, dialing a phone number, getting a ring at the far end (or a busy signal), and picking up the phone at the far end – and the telephone conversation. There are several protocols for doing the call setup and takedown, such as H.323, SIP, and Megaco. They use TCP, a connection-oriented network protocol, to encapsulate the call setup and takedown phases. The exchange of actual encoded voice data occurs after the call setup (and before the call takedown), using two data flows – one in each direction. Each of these two data flows uses the Real-time Transport Protocol (RTP)[6]. RTP is widely used for streaming audio and video; it is designed to send data in one direction with no acknowledgment. The header of each RTP datagram contains a timestamp—so the receiver can reconstruct the timing of the original data—and a sequence number—so the receiver can deal with missing, duplicate, or out-of-order datagrams.

Our focus here is on the voice conversation, the two RTP streams, since they are the important elements in determining quality of the voice conversations. Let’s look at the composition of the RTP datagrams, which carry the voice packets.

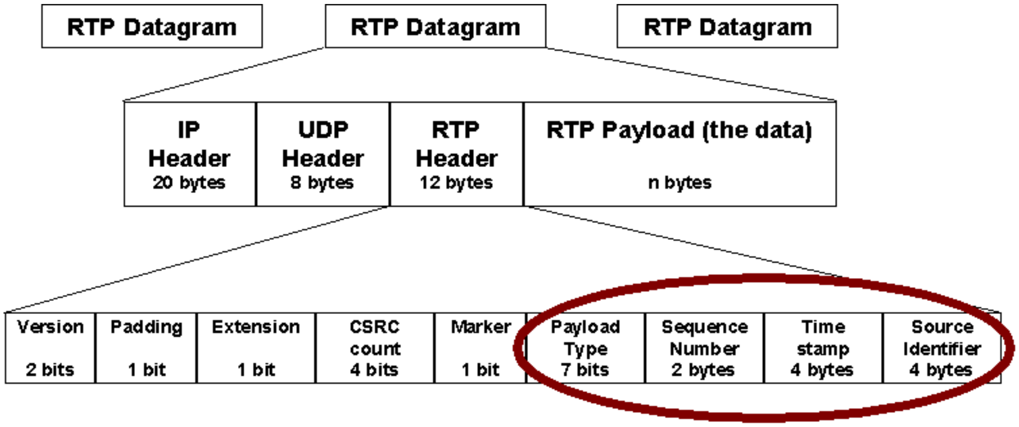


Figure 1. The header used for RTP follows the UDP header in each datagram. The important four fields in the RTP header are described below.

RTP is a connection-less protocol. All the fields related to RTP sit inside the user datagram protocol (UDP). RTP is not commonly part of the TCP/IP protocol stack, so applications are coded to add and recognize an additional 12-byte header in each UDP datagram. The sender fills in each header, which contains four important fields:

RTP Payload Type

Contains the codec indication, describing the type of data (such as, voice, audio, or video) and how is it encoded. A table of the codecs used most commonly in VoIP is shown below, along with their datagrams sizes and bandwidth consumption.

RTP Payload Type (codec)	Coding Type	Data Rate in kbps	Data Bytes per 30ms packet	Total Data-gram Size (bytes)	No Silence Supres-sion, 2 flows (kbps)
G.711	PCM	64.0	240	298	158.93
G.729	CS-ACELP	8.0	30	88	46.93
G.723.1a	ACELP	6.3	24	82	43.73
G.723.1m	MP-MLQ	5.3	20	78	41.60

Figure 2. Common voice codecs and their bandwidth requirements.

Sequence Number

Helps a receiver reassemble the data and to detect lost, out-of-order, and duplicate datagrams.

Timestamp

Used to reconstruct the timing of the original audio or video. Also, helps a receiver determine consistency or the variation of arrival times, known as jitter.

It's the timestamp that brings real value to RTP. An RTP sender streams data at a consistent rate, so the timestamps should be equally spaced. On the receiving side, an RTP application can mark when each datagram actually arrives and compare this to the arriving timestamp. If the datagrams arrive at precisely the same rate they were sent, the receiver sees no variation. However, there could be lots of variation in the arrival rate of the datagrams, depending on network conditions, and this jitter can be easily calculated by the receiver.

Source ID

Helps a receiver distinguish multiple, simultaneous streams, using a unique sender-generated value.

Test with the codec you plan to use in the deployed VoIP system. For general testing (or when you don't know what codec is being used), we've found the G.711 codec at 64 kbps to be the most effective in testing network readiness. Although its larger packets may be more likely to encounter bit errors, the G.711 is less sensitive to lost packets than the non-linear codecs, and the larger frame size is more efficient in its bandwidth usage (that is, the data payload is large compared to the header overhead).

The headers can be a lot of overhead, depending on the size of the data payload. For example, a typical G.729 payload is 30 bytes. With RTP, the total header overhead consists of RTP (12 bytes) + UDP (8 bytes) + IP (20 bytes) = 40 bytes – so more than 50% of the datagram is the header.

For the G.711 codec at 64 kbps, the bandwidth requirements aren't heavy compared to most LANs (although they're not the trivial rates of 8 kbps or less). To achieve lower bandwidths such as 8 kbps, a codec must use more complex compression schemes. This can result in multiple samples of audio compressed into a single frame. The loss of a single frame can encompass a surprisingly long period of audio. Also, some codecs offer packet loss concealment, which tries to minimize the impact of a lost packet. We did not employ packet loss concealment in our testing, giving us a more straightforward evaluation. .

We don't represent silence suppression in our readiness, since we want to evaluate relatively demanding conversations, not ones with lots of silence. In general, use of silence suppression can reduce the bandwidth consumption of VoIP data streams.

We use random data in our data payloads, to minimize the effects of data compression done by devices in the data network.

Some IP phones let you configure the "delay between packets" or "speech packet length," that is, the rate at which the sender delivers packets into the network. For example, at 64 kbps, a "20 ms speech packet" implies that the sending side creates a 160-byte frame every 20 ms. For a given data rate, if you increase the delay, the frames get larger (since they're sent less frequently). So, a delay of 30 ms at a data rate of 64 kbps would mean sending 240-byte frames.

Lastly, be sure to match any other known configuration parameters to get the best possible assessment. For example, Alcatel equipment uses a narrow range of port numbers for its RTP streams – assign ports from this range to the traffic being tested. VoIP gateways by Cisco Systems set the DiffServ bits in each IP frame of VoIP traffic they generate to the bit value "101000," indicating that the datagrams should be treated with "expedited" priority. DiffServ is one of several Quality of Service (QoS) tuning techniques for TCP/IP; giving RTP streams a higher setting than all zeros (best effort) may improve how they're handled as they pass through routers and other network devices.

Running the Test

Having created this assessment, pick a representative pair of locations in the network between which to run it. The locations for these endpoints of the RTP traffic should be in the same places where you would put your call gateways, that is, from the place where the voice conversations are digitized to the place where they are converted back to audio. This might also be the location of IP phones, which are connected directly to a LAN.

We've seen that the best way to do a thorough assessment is to repeat the tests periodically over the course of one or more days. Start with the simple case: run the test for a minute in duration, once every fifteen minutes, for 24 hours. We designed our test to take a measurement every six seconds, so we get ten samples every minute. Some quick math says we run for four total minutes every hour, collecting 176 samples in a 24-hour day. We've seen some assessments that collected many more samples, however, and even ran for as long as a week continuously.

You want to see the network's behavior during its peaks and valleys, when it's heavily loaded and when nobody's there. You might want to consider running for multiple days, if there are some days where network traffic may be significantly heavier. For example, there may be much more financial data exchanged on the network at the end of the month; you'd certainly like to know that the network could also support telephone calls on those days.

Also, you may have areas in the network topology where the traffic characteristics vary from other areas. For example, are large CAD diagrams exchanged between some departments? Is streaming video or video-conferencing already prevalent in some parts of the network? If you can identify these places in your topology, consider adding some additional representative endpoints to your assessment, but don't drown in data – try to avoid more than 8-10 endpoints in your initial assessment.

Analyzing the Data

Three network measurements influence the perceived quality of voice conversations: one-way delay, jitter, and packet loss.

1. **One-way delay.** The time it takes to get across the network is the primary indicator of the "walkie-talkie" effect. Humans are used to having conversation where they both talk at the same time. If the one-way delay between the speakers' ears is more than 200 ms, people find it disconcerting and rate the voice quality as poor.

   When one-way delay is difficult to measure, round-trip time divided by two can be used as a reasonable approximation. However, this hides assumptions about the symmetry of the paths between the two endpoints.

2. **Jitter.** A jitter value captures the amount of variability in the arrival times of the packets at the receiver. The sending side sends packets at a regular periodic rate, say every 20 ms. Ideally, the receiving side would receive the frames at the same rate, in which case there's no jitter. However, all kinds of things can happen in data networks, and some packets arrive quickly while others arrive more slowly. The slowest frames essentially become part of the delay – if they arrive too slowly, the overall delay appears to increase.

   One method of damping the variability of arrival rates is to put a "jitter buffer" between the network layer and the VoIP application. A jitter buffer holds frames at the receiving side. It can compensate for variability of arrival rates and also deal with frames which arrive out of order. It thus hands the frames to the processing application in order, at a more consistent rate. However, since the jitter buffer needs to hold the frames for some time to do this damping, it further increases the delay. And, compounding the problems somewhat, packets can be lost when a jitter buffer is overrun.

3. **Packet loss.** Packets which are lost generally can't be recovered, so they appear as momentary gaps in the conversation. Some tiny gaps are okay, but a consistently high rate of lost packets or bursts where lost of packets are lost are disturbing to human listeners.

Having a low overall average (say 1%) but having that occur in large bursts of high-percentage loss isn't good.

The network services team for one of the users of our product suggests the following constraints for the minimum data network quality:

- One-way delay: between endpoints, delay should be less than 50ms.
- Jitter: between endpoints, jitter should be less than 20ms. This value has some latitude depending on the type of service the jitter buffer has in relationship to other router buffers.
- Packet Loss: the maximum loss of packets (or frames) should be 0.2% or less.

Calculating a Score

While we capture granular measurements for one-way delay, jitter, and lost packets, it can be a lot to analyze for someone not extensively trained. Our goal is to make the evaluation simple, so we developed a single numerical score to estimate the quality of the voice conversation. Like all scores, it's strongest at the extremes, which results in a simple set of rules for those doing an assessment:

- If the score is clearly high, the network passes the assessment.
- If the score is clearly low, the network fails the assessment.
- If the score is in the middle, the network's probably not in great shape, and more examination of the underlying data is called for.

We calculate a voice quality score based on the ITU G.107 recommendation. G.107 consists of the E-model, a computational model for use in transmission planning. The E-model provides a way to compute a scalar quality rating value, R, which varies directly with the overall conversational quality. The E-model takes a large number of parameters, all of which have recommended default values, which we used.

The Mean Opinion Score (MOS) in ITU P.800 [2] is a subjective measurement of call quality as perceived by the receiver. A MOS can range from 1 to 5, using the following rating scale:

MOS	Quality Rating
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

An estimate of the MOS can be calculated from the R factor, the quality rating of the E-model.

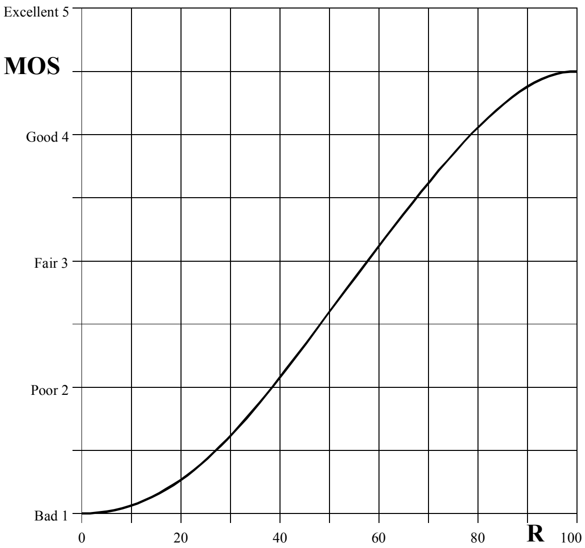
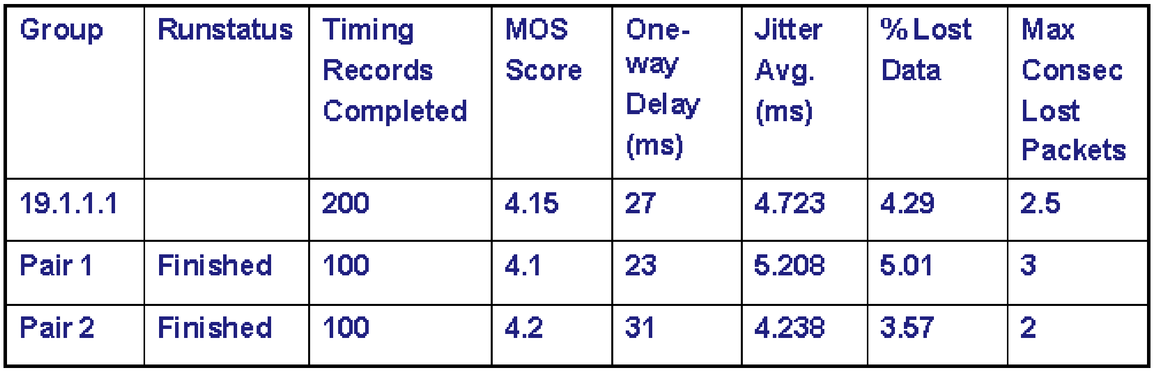


Figure 3. R factor values from the E-model are shown on the X-axis, with MOS values on the Y-axis. The S-curve shows the mapping between R factor values and an estimated MOS.

The only control we offered users is to specify the codec, which has an implicit delay function. Any burstiness, jitter, or delay measured by the test is used in the score calculation. We extended the E-model to factor in percentage of packet loss, packet loss burstiness (calculated from maximum consecutive packet loss), and codec.

Here's an example of the output from our software, *Chariot* [7]:



Group	Runstatus	Timing Records Completed	MOS Score	One-way Delay (ms)	Jitter Avg. (ms)	% Lost Data	Max Consec Lost Packets
19.1.1.1		200	4.15	27	4.723	4.29	2.5
Pair 1	Finished	100	4.1	23	5.208	5.01	3
Pair 2	Finished	100	4.2	31	4.238	3.57	2

Figure 4. Chariot output, showing two RTP sessions (pair 1 and pair 2) – representing a two-way voice conversation. The MOS scores over 4.0 indicate that this data network is probably ready for a VoIP deployment.

Follow-on Steps

The first question is to determine if the network's suitable for one VoIP conversation. If the score indicates that it isn't, it's time to upgrade and tune your data network. Do all the network equipment upgrades and tuning necessary to carry the VoIP traffic well – but without actually introducing any VoIP devices. Assess the network repeatedly, until you're convinced it's ready, and has been stabilized for its existing applications and users.

If the VoIP assessment indicates the network's ready now, you'll want to test its capacity to see how many calls can be supported. Replicate the test setup described above, but don't run for 24 hours. Run the test for a one-minute period, a few times during the day where your 24-hour test showed heavy activity. Test ten conversations at a time for a minute; what happens to the scores? Next try twenty, then thirty concurrent conversations. Plot the results on a graph; you should start to see that as the number of calls increases, the quality decreases. Don't kill your data network during prime time by stress testing its capacity. However, start to form the graphs showing how many conversations can be supported with good quality.

Network traffic can be tuned using many router and gateway tuning parameters. Quality of service techniques assist in tuning by allowing some traffic to be classified to get better handling than traffic with other classifications. For example, you might classify RTP traffic using G.729 codec to get an assured amount of bandwidth from end-to-end in the network. We think network tuning should be a post-deployment exercise. Make sure the network's ready for the new traffic, deploy it and get it running well, then begin doing any optimizations.

Summary

Using data networks to carry telephone conversations is another step along the convergence path. While its bandwidth consumption may be relatively low, it has stringent demands for low latency and the regular arrival of data packets. These constraints are new to many network personnel, who must fit them against a background of the existing data network traffic.

We believe a staged approach to VoIP deployment can be cost efficient. The first stage is to assure the readiness of the data network for the added VoIP data traffic. A straightforward methodology and set of tools can help you quickly judge the suitability of the network. If it's okay, proceed to the next stage of evaluating VoIP equipment and training your deployment team. If the data network's not ready for VoIP, fix it first. Do all the upgrades and tuning necessary in the data network to carry the VoIP traffic well. Assess the network repeatedly, until you're convinced it's ready, and has been stabilized for its existing applications and users. Then, move to the next stage of evaluation and training.

We've shown a methodology and set of tools to help assure successful VoIP deployments. We've focused on understanding the quality of the RTP data flows that encapsulate the voice conversations, since they're the traffic with the new constraints. Finally, we've introduced an objective scoring system using the G.107 E-model, so personnel with simple software, little training, and no additional equipment can quickly make useful assessments.

For Additional Information

1. McCullough, D. J. and J. Q. Walker II. "Interested in VOIP? How to Proceed," *Voice 2000*, a supplement to *Business Communications Review*, April 1999, pages 16-22.
2. ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality."
3. ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning."
4. ITU-T Recommendation G.108, "Application of the E-model: A planning guide."
5. ITU-T Recommendation P.861, "Objective quality measurement of telephone-band (300-3400 Hz) speech codecs"
6. Walker II, J. Q. and J. T. Hicks. "Protocol ensures safer multimedia delivery," *Network World*, volume 16, number 44, November 1, 1999, page 53.
7. *Chariot*, by NetIQ Corporation. See www.netiq.com/Products/Network_Performance/Chariot/ for additional information.

About the Authors

John Q. Walker II is the director of network development at NetIQ Corporation. He was a founder of Ganymede Software Inc., which became part of NetIQ in spring 2000. He can be reached at johnq@netiq.com.

Jeff Hicks is a senior software developer and the leader of the Chariot development team with NetIQ Corporation. He can be reached at jeff.hicks@netiq.com.

Acknowledgments

Gracious thanks to the readers who helped improve this paper: Scott Patterson, Kim Shorb, Carl Sommer, and John Wood.