Evaluating Data Networks for VoIP

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 challenge in deploying VoIP successfully is the quality of calls, which is heavily affected by existing data network traffic. In most scenarios, the data network requires tuning to achieve acceptable quality for voice traffic. When changes are made to a network, network administrators need a way to ensure that they will continue to get the same or better quality. This paper describes practical steps for assessing whether a data network is ready for VoIP. Our software, the *Chariot VoIP Assessor*, is designed so that personnel with little training and no VoIP equipment can quickly make useful VoIP-readiness assessments. Another software product, *Chariot* (with its *VoIP Test Module*) is particularly helpful when doing VoIP deployment, assisting in the data network tuning that will probably be required.

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. Is the data network you use ready for this new type of traffic? That is, will IP phone users be satisfied with the quality of their telephone conversations?

Our focus during the past several years has been building software to measure network performance. We have worked with 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 we propose are easy to follow and can be done before investing in any voice equipment or deploying anything new in a data network.

1. Create a test that simulates the data flows of VoIP traffic. One or more conversations can be simulated in a test.
2. Perform a VoIP readiness assessment by running the test periodically on the live network and collecting appropriate network performance measurements.
3. Look at the resulting "score," which estimates the quality of voice conversations on the network. Examine the detailed measurement data if the scores indicate poor quality.
4. Troubleshoot problem areas and perform QoS tuning where appropriate [8].
5. Re-run the assessment to ensure that changes made to the network are sufficient to support VoIP well.
6. Perform more intensive tests in the lab to determine the network's capacity. This information will be important as you deploy and expand the VoIP network.

Chariot VoIP Assessor from NetIQ makes determining a network's readiness for VoIP prior to the purchase and deployment of VoIP equipment easy. VoIP Assessor [6] emulates VoIP traffic on a data network, collects key call quality measurements and analyzes the results – enabling you to make an informed decision about how to proceed with VoIP deployment.

Once you have run the initial assessment, you are likely to find areas within a network that must be either upgraded or tuned. Chariot's VoIP Test Module lets you test VoIP-enabled network equipment and to troubleshoot and tune network performance in preparation for VoIP. Chariot can also be used to do capacity testing in the test lab to better understand the limits of a network to support multiple VoIP conversations.

The equipment and settings involved in making voice work well in a data network are different from those that make traditional business applications work well. The VoIP-readiness assessment lets you determine the status of a real network without any voice hardware. You can discover whether a data 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.

Understanding Voice Quality

Call quality testing has traditionally been subjective: picking up a telephone and listening to the quality of the voice. The leading subjective measurement of voice quality is the MOS (mean opinion score) as described in the ITU (International Telecommunications Union) recommendation P.800 [1].

In using MOS with human listeners, lots of people listen to audio and give their opinion of the call quality. This certainly works well, but you can guess it's expensive to have a bunch of people standing around each time you make a tuning adjustment. The good news is that the human behavioral patterns have been heavily researched and captured. The ITU P.800 recommendation describes how humans react – what score they would give – as they hear audio with different aspects of delay or datagram loss. This mapping between network characteristics and a quality score makes MOS valuable for doing network assessments and tuning.

Considerable progress has been made in establishing objective measurements of call quality. Various standards have been developed:

- PSQM (ITU P.861) / PSQM+
  Perceptual Speech Quality Measure
- MNB (ITU P.861)
  Measuring Normalized Blocks
- PESQ (ITU P.862)
  Perceptual Evaluation of Speech Quality
- PAMS (British Telecom)
  Perceptual Analysis Measurement System
- The E-model (ITU G.107)

MOS is the widely accepted criterion for call quality, and the vendors that implement these scoring algorithms all map their scores to MOS.

PSQM, PSQM+, MNB, and PESQ are part of a succession of algorithm modifications starting in ITU recommendation P.861. British Telecom developed PAMS, which is similar to PSQM. The PSQM and PAMS measurements send a reference signal through the telephony network and then compare the reference signal with the signal that's received on the other end of the network, by means of digital signal processing algorithms. Several traditional voice measurement tools have implemented PSQM and PAMS measurements.

These measurements are good in test labs for analyzing the clarity of individual devices. For example, it makes sense to use PSQM to describe the quality of a telephone handset.

However, these approaches are not really well suited to assessing call quality on a data network, since they don't know about data networking. They're based in the older telephony world.

- The underlying models are not based on data network issues, so they can't map back to the network issues of delay, jitter, and datagram loss. Their output doesn't direct the network staff how to tune the data network.
- They don't factor in the end-to-end delay between the telephone speaker and listener. Excessive delay adversely affects MOS.
- They show quality in one direction at a time, rather than the two-way flow used in a real telephone conversation.
- They don't scale, letting you see the effect of multiple simultaneous calls between a pair of locations.
- They require invasive hardware probes, which you need to purchase and deploy before beginning VoIP measurements.

ITU recommendation G.107 [2,3] introduced the E-model. The output of an E-model calculation is a single scalar, called an "R value," derived from delays and equipment impairment factors. Once an R value is obtained, it can be mapped to an estimated MOS.

NetIQ's *Chariot* [5] and *Chariot VoIP Assessor* [6] use a modified form of the E-model when evaluating voice quality. They calculate an R value and convert that to an estimated MOS. Their testing works by generating real-time transport protocol (RTP) streams that mimic VoIP traffic. The RTP traffic flows between two endpoints in a data network. Each time a test or assessment is run, measurements are collected for the one-way delay time, the number of datagrams lost, the number of consecutive datagrams lost, and the amount of variability in the arrival time of the datagrams (known as jitter). These measurements capture what's important for voice quality: how the two people at the two telephones perceive the quality of their conversation.

How It Works

The Chariot family of products cause precisely-defined network traffic to be generated between pairs of computers and observe the performance of the traffic. Traffic patterns are highly tailorable, letting you recreate the traffic generated by real user applications. Performance tests, capturing everything about the traffic patterns and the computers involved, can be saved and reliably repeated. For example, you can see the effect of changing the hardware or software along network paths; you can see the effect of adding new users; or you can track the level of service available in a network.

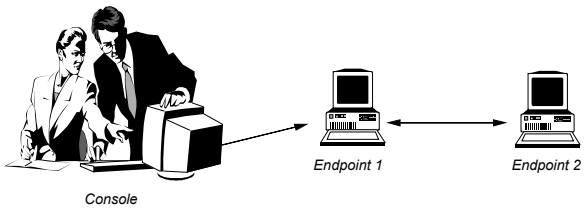

Figure 1: A simple performance test is set up at a Chariot console and run between a pair of computers, labeled Endpoint 1 and Endpoint 2.

The set of programs is operated from the *console*, where you create and run *tests*. Creating a test consists of deciding which computers to use and the type of data traffic to run between them. The computers executing the tests are referred to as *endpoints*. An *endpoint pair* comprises the network addresses of the two endpoint computers, the network protocol to use between them, and the type of application to emulate.

For each endpoint pair, select an *application script* corresponding to the application you are simulating. The endpoint programs use an application script to generate the same data flows that an application would, without installing the application. A set of pre-built application scripts provides standard performance benchmarks and emulates common end-user applications.

Today, endpoints run on twenty operating systems, supporting six network protocols. Support for TCP, UDP, and RTP is available on all these systems; VoIP testing is supported on Windows, Linux, Solaris. Chariot supports tests with multiple concurrent connections between any endpoints.

The VoIP Test Module for Chariot and Chariot VoIP Assessor include high-precision measurements for one-way delay between endpoints. The products make it simple to evaluate voice quality. Improved tables and graphs show the one-way delay and the distribution of lost datagrams, and can estimate a mean opinion score for the simulated voice connection between pairs of endpoints.

Emulating a Call

Implementing a real telephone call on a data network involves the call setup – that is, 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 then the telephone conversation itself. There are several higher-layer protocols that accomplish the call setup and takedown, such as H.323, MGCP, SIP, and Megaco. They principally use TCP, a connection-oriented network protocol, to execute 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 – letting both participants speak at the same time. Each of these two data flows uses the Real-time Transport Protocol (RTP)[4].

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 or out-of-order datagrams.

The two RTP streams, that is, the conversation itself, are the important elements in determining call quality of the voice conversations. Let's look at the composition of the RTP datagrams, which transport the voice datagrams.

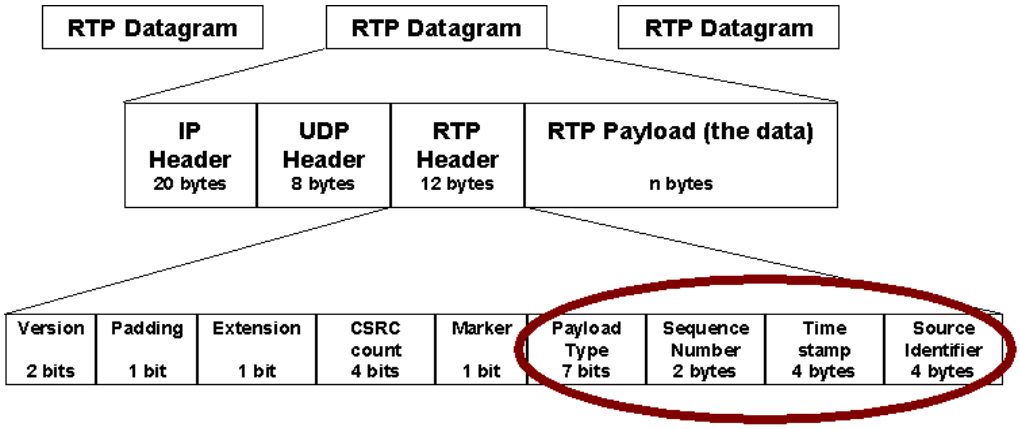

Figure 2. The header used for RTP follows the UDP header in each datagram. The four important fields in the RTP header are described below.

All the fields related to RTP sit inside the user datagram protocol (UDP). So, like UDP, RTP is a connection-less protocol. 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

Indicates which codec to use. The codec describes the type of data (such as voice, audio, or video) and how is it encoded. A table of codecs used commonly in VoIP is shown below, along with their datagram sizes and overall bandwidth consumption.

RTP Payload Type (codec)	Coding Type	Data Rate in kbps	Data Bytes per 30ms packet	Total Datagram Size (bytes)	Combined bandwidth for 2 flows (kbps)
G.711	PCM	64.0	240	298	158.93
G.726-32	PCM	32.0	120	178	94.93
G.729	CS-ACELP	8.0	30	88	46.93
G.723.1m	MP-MLQ	6.3	24	82	43.73
G.723.1a	ACELP	5.3	20	78	41.60

Figure 3. Common voice codecs and their bandwidth requirements. Total datagram size includes a 40-byte IP/UDP/RTP header and an 18-byte Ethernet header.

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 puts a timestamp in each datagram it sends. An RTP receiver sees when each datagram actually arrives and compares this to the timestamp. If the time between datagrams arrivals is the same as when they were sent, there's no variation. However, there could be lots of variation in the arrival times of datagrams depending on network conditions, and the receiver can easily calculate this jitter.

Source ID

Lets a receiver distinguish multiple, simultaneous streams.

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 datagrams may be more likely to encounter bit errors, the G.711 is less sensitive to lost datagrams than the non-linear codecs, and

the larger frame size uses bandwidth more efficiently (that is, the data payload is large compared to the header overhead).

The headers can add 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. This means that more than 50% of the datagram is the header. By the way, a router function called RTP header compression (cRTP) can reduce the header to a tenth of this size, but may introduce more latency and distortion.

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). The lower-bandwidth codecs, such as G.729, must use more complex compression schemes. This can result in multiple samples of audio compressed into a single frame. However, the loss of one 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.

Use of silence suppression in voice conversations can reduce the bandwidth consumption of VoIP data streams by 30% to 50%. You can set the amount of silence suppression for each pair of endpoints being tested. For the most demanding evaluation of a network's readiness, don't test with silence suppression

Endpoints use random data in their data payloads to minimize the effects of compression done by devices in the data network. Randomly-generated data can't be compressed.

Some IP phones let you configure the "delay between packets" or "speech packet length," that is, the rate at which the sender delivers datagrams into a network. For example, at 64 kbps, a "20 millisecond speech packet" implies that the sending side creates a 160-byte datagram payload every 20ms.

There is a simple equation that relates the codec speed, the delay between voice packets, and the datagram payload size:

```
Payload size (in bytes) =

 Codec speed (in bits/sec) x packet delay (ms)
 ---------------------------------------------
             8 (bits/byte) x 1000 (ms/sec)
```

In this case, the size of the datagram payload, 160 bytes, was determined as:

```
160 bytes = (64000 x 20)/8000
```

For a given data rate, increasing the delay causes the datagrams to get larger, since they're sent less frequently. A delay of 30ms at a data rate of 64 kbps would mean sending 240-byte datagrams.

Creating a Test

These VoIP setup values come together in the Chariot "Add VoIP Endpoint Pair" dialog. A toolbar icon let's you quickly bring up the dialog:

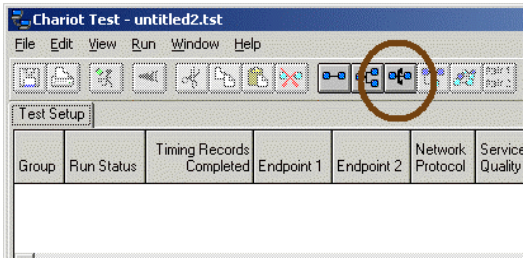

Figure 4: The "Add VoIP Pair" icon on the Chariot Console toolbar.

In this dialog, you set the fields discussed above needed to create a VoIP test. In the following figure, you can see that the G.711 codec has been selected for testing between endpoints named Houston and Portland. Choosing the G.711 codec causes Chariot internally to use its G.711 application script. This streams RTP datagrams from Endpoint 1 to Endpoint 2 at 64 kbps. This test uses the default speech packet length of 20ms. Using the G.711 codec, which runs at 64000 bits/sec, the payload size is 160 bytes. A timing record measurement is taken every 3 seconds.

The Service quality field lets you specify the TCP/IP Quality of Service to be used with this endpoint pair. In the following figure, we leave the "Service quality" field blank, meaning that we didn't specify an explicit QoS for this stream of traffic.

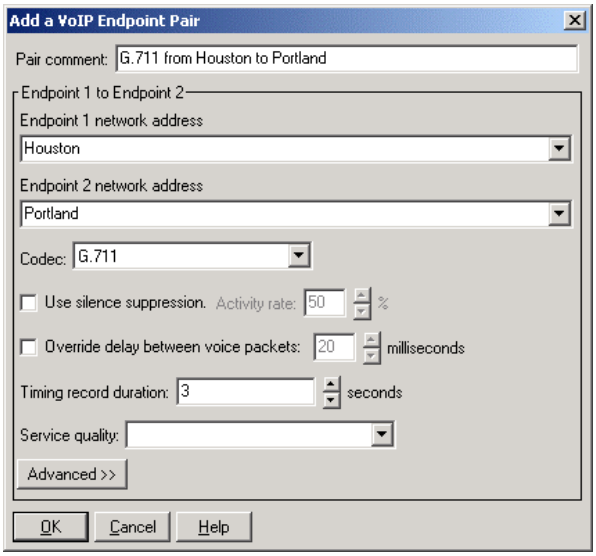

Figure 5: The "Add a VoIP Endpoint Pair" dialog lets you set the parameters needed for testing voice quality between one pair of endpoints.

Many VoIP gateways 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 flow" 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. QoS parameters can be set and saved in Chariot using a QoS template.

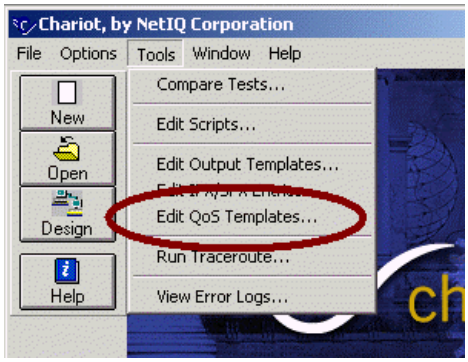

Figure 6: Create a new QoS template at the Chariot main window.

To emulate the VoIP Gateway traffic, we added a new DiffServ template named "Expedited Flow DSCP." Figure 7 shows the setup dialog for a DiffServ template.

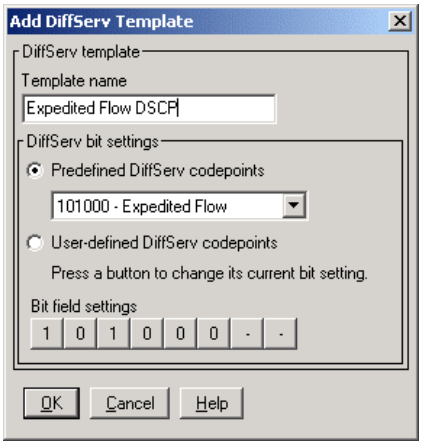

Figure 7: The DiffServ setting for the codepoint '101000' is saved in a named QoS template.

Also, 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. In the following example, we pressed the **Advanced** button on the **Add a VoIP Endpoint Pair** dialog to specify source and destination port numbers. We also used the DiffServ QoS template we created above.

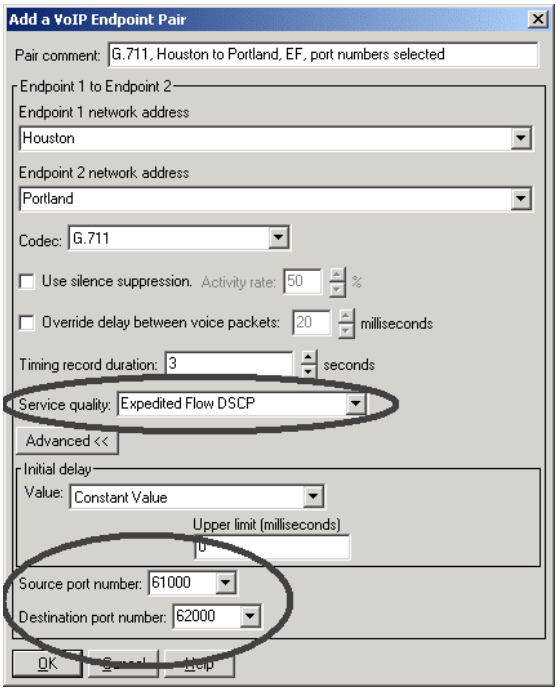

Figure 8: The Advanced version of the "Add a VoIP Endpoint Pair" dialog. Compared to the previous dialog, we specified a QoS template, as well as source and destination port numbers.

A voice conversation consists of two RTP streams. Only one has been set up so far, so make a second copy of this pair and exchange the two endpoint addresses. Do this by replicating the existing VoIP endpoint pair in the Chariot test window. Edit the new pair, flipping the network addresses you used for Endpoint 1 and Endpoint 2. When you've finished, you should see two rows in the Chariot test window as shown below. Be sure to save the file with a unique name.

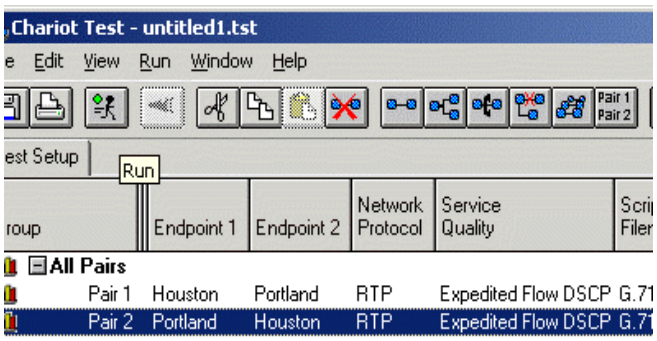

Figure 9: Two VoIP endpoint pairs, ready to be used for an initial voice readiness assessment.

Running a Test

When you're ready to begin testing, pick a representative pair of locations in the network between which to run tests. The locations for these endpoints of the VoIP traffic should be in the same places where you would put the call gateways or IP PBXes, that is, at the place where the voice conversations are digitized and at 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. Chariot VoIP Assessor is designed just for this purpose. VoIP Assessor clearly guides a user through the steps needed to perform a VoIP readiness assessment over a period of days. Once the assessment is complete, VoIP Assessor analyzes the results and generates a polished, customizable report of the network's readiness for VoIP.

There are two sets of numbers you can obtain from the local data network and telephony management teams:

- When is the peak data network usage? How heavily is the network utilized or congested during that time? How long does the peak last?
- When is the peak call time? How many calls occur during that time? How long does the peak last?

You want to see voice quality during the data network's peaks and valleys: when it's heavily loaded and when nobody's there. Consider running the evaluation 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, in some areas of the network topology, 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 yourself in data.

Analyzing the Data

Three network measurements influence the perceived quality of voice conversations: end-to-end delay, jitter, and lost data. The following section details how these results are presented in Chariot.

End-to-End Delay

End-to-End delay, the time it takes to get data across the network, is the primary indicator of the "walkie-talkie" effect. Humans are used to having conversations where they both talk at the same time. Most listeners notice when the delay is more that than about 150 ms; when it exceeds 200ms, they find it disturbing and describe the voice quality as poor.

The end-to-end delay is actually made up of four components:

- **Propagation delay**: the time to travel end-to-end across the network. The propagation delay between Singapore and Boston is much longer than between New York and Boston.
- **Transport delay**: the time to get through the network devices along the path. Networks with many firewalls, many routers, and slow WANs introduce more transport delay than a LAN on one floor of a building.
- **Packetization delay**: the time for the codec to digitize the analog signal and build frames – and undo it at the other end. The G.729 codec has a higher packetization delay than the G.711 codec, because it takes longer to do its compression.
- **Jitter buffer delay**: the fixed delay introduced by the receiver to hold one or more datagrams, to damp variations in arrival times.

The combined value of propagation delay and transport delay is what is termed as "one-way delay" in Chariot. Packetization delay and the jitter buffer delay are constants for any given endpoint pair. They're not shown in the delay graphs, but are incorporated in the overall MOS values.

Measuring response time (round-trip delay) and dividing the resulting time measurement by two isn't always a good approximation of one-way delay. Response time hides assumptions about the symmetry of the paths between two endpoints. The two RTP streams in a VoIP call can take different paths through an IP network.

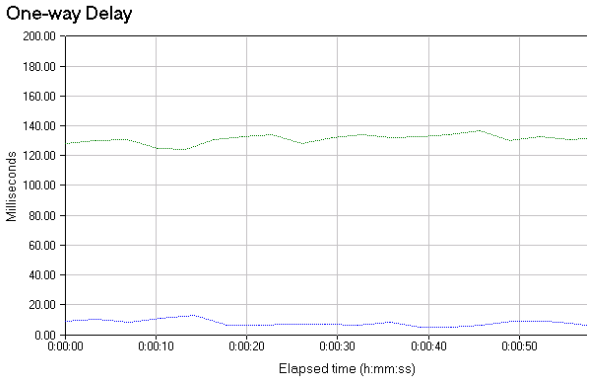

Figure 10: There's quite a difference between the one-way delay values in the two directions of this conversation. At about 130ms, the one-way delay slightly affects the MOS.

Performance endpoints calculate one-way delay explicitly, rather than just taking the round-trip time and dividing it in half. The endpoints start with flows similar to those used by the Network Time Protocol (NTP) [7]. NTP generally has an accuracy of around plus-or-minus 200ms. Our design for giving good MOS feedback called for clock precision of about plus-or-minus 1ms, which led us to design more precise algorithms for software-based clocking.

Endpoints maintain local copies of their clocks, because there can be many simultaneous connections. Also, the internal clocks in every different operating system and computer platform seem to be a little different, and the clocks drift apart over time.

The endpoints maintain virtual (software) clocks for each partner involved in a VoIP test. The virtual clocks consist of the offset between the microsecond clocks maintained by the two endpoints. The microsecond clock is a high-resolution clock that's maintained independently of the operating system's system clock.

The endpoints compare their respective views of the clocks prior to the start of each test and periodically during a test run. They also measure clock synchronization and drift between test runs, to establish a track record for the expected delay.

Our one-way delay algorithms have proven robust, in measurements with thousands of endpoint pairs. We've also verified their effectiveness in testing with stratum 1 GPS timeservers.

Jitter

A jitter value captures the amount of variability in the arrival times of the datagrams at the receiver. The sending side sends datagrams at a regular periodic rate, say every 20ms. Ideally, the receiving side would receive the datagrams at the same rate, in which case there's no jitter. However, all kinds of things can happen in data networks, and some datagrams arrive quickly while others arrive more slowly. If slow datagrams arrive too late, they are discarded to make way for the datagram which follows them.

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 datagrams at the receiving side. It can compensate for variability of arrival rates and also deal with datagrams that arrive out of order. It hands the arriving datagrams to the processing application in order, at a more consistent rate. However, since the jitter buffer needs to hold the datagrams for some time to do this damping, it further increases the delay. Compounding the problems somewhat, datagrams can be lost when a jitter buffer is overrun.

Chariot tests can simulate the effect of a jitter buffer, showing the effect of various size buffers on the estimated mean opinion score.

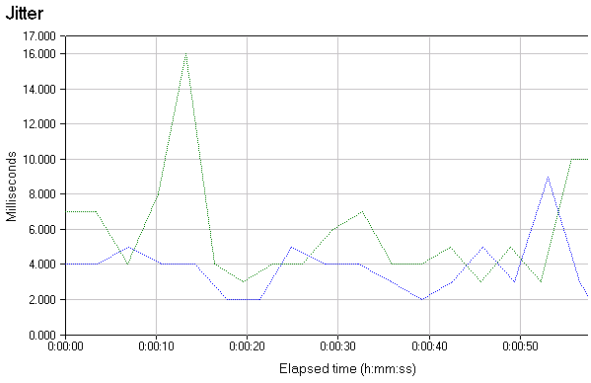

Figure 11: The jitter for the two RTP flows in this conversation.

Lost Data

Datagrams that 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 datagrams or periods where lots of datagrams are lost are disturbing to human listeners. Even with a low overall average (say 1%), if loss occurs in bursts, the quality suffers.

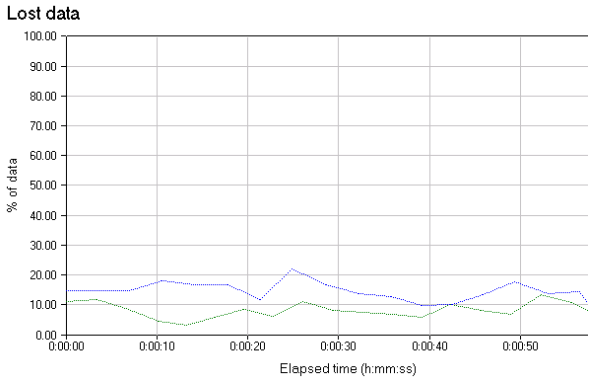

Figure 12: The amount of lost data for the two directions is relatively high. However, the next graph shows that most of the data was lost just one datagram at a time.

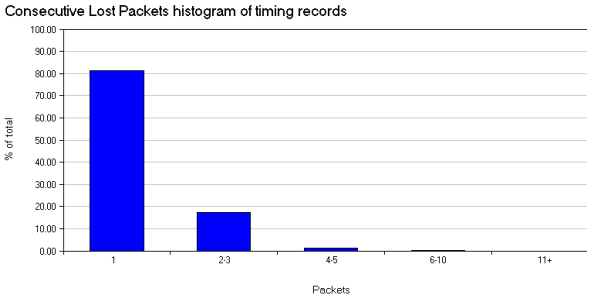

Figure 13: By far, most of the loss of datagrams did not occur in lengthy bursts. However, this wasn't evenly distributed between these two flows.

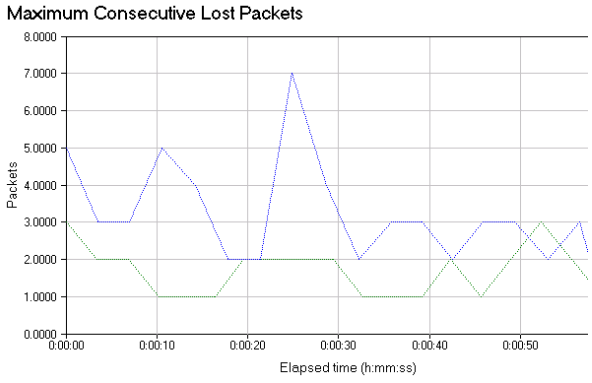

Figure 14: The incoming flow (the higher of these two lines) had more periods where consecutive datagrams were lost.

Calculating a Score

These granular measurements for one-way delay, jitter, and lost data can be a lot to analyze for someone not extensively trained. Our goal is to make the evaluation simple, so a single numerical score is used to estimate the quality of a 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.

The Chariot products calculate their voice quality scores based on the ITU G.107 and P.800 recommendations. G.107 describes the E-model, which computes a scalar quality rating value, R. The E-model takes a large number of parameters. Most of which have recommended default values, which are used in the calculations.

The Mean Opinion Score (MOS) in ITU P.800 [1] is a subjective measurement of call quality as perceived by the receiver. A MOS can range from 5 down to 1, 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 value, the quality rating of the E-model.

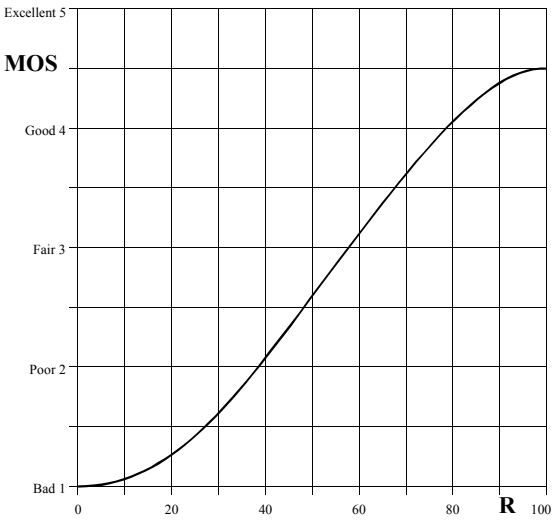

Figure 15. R values from the E-model are shown on the X-axis, with MOS values on the Y-axis. The S-curve shows the mapping between R values and an estimated MOS.

The only control in the E-model offered to users is to specify the codec, which has an implicit delay function. Bursts of consecutive lost datagrams, jitter, and one-way delay measured by the test are used in the calculation of a MOS estimate. The E-model was extended to factor in percentage of packet loss, packet loss burstiness (calculated from maximum consecutive packet loss), the jitter buffer, and the codec.

Figure 16 shows the MOS estimate for the same two RTP flows shown above.

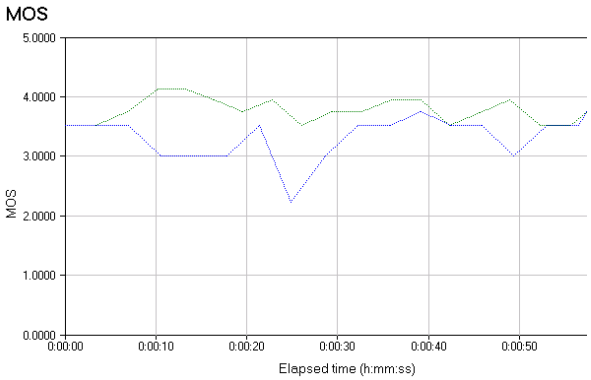

Figure 16: The MOS estimate for the two RTP flows. The bottom line (the incoming flow), has a lower score. It was heavily influenced by seeing larger sets of consecutive lost datagrams.

Figure 17 shows the tabular output from running a simple VoIP test with Chariot. It shows the two VoIP pairs in this conversation, as viewed from Endpoint 1. The outgoing pair has a higher average MOS estimate (3.777) than the incoming pair (3.362). The difference is mostly influenced by the longer bursts of lost datagrams seen by the incoming flow.

Bringing these measurements together, one of the commercial users of the Chariot family of products suggests the following constraints for good voice quality on a data network:

- One-way delay: between endpoints, delay should be less than 100ms.
- Jitter: between endpoints, jitter should be less than 20ms (with some latitude, depending on the use of jitter buffers).
- Lost data: the maximum loss of datagrams should be 0.2% or less.

Pair	MOS Average	MOS Minimum	MOS Maximum	One-Way Delay Average (Ms)	Jitter Average (Ms)	Percent Bytes Lost	Maximum Consecutive Lost Datagrams
Outgoing	3.777	3.51	4.139	131.158	6	8.23%	3
Incoming	3.362	2.223	4.409	7.5	3.667	14.87%	7

Figure 17. Chariot output, showing two RTP sessions (Pair 1 and Pair 2) representing a two-way voice conversation. The MOS estimates around 3.5 indicate that this data network is marginally ready for a VoIP deployment.

Follow-on Steps

In assessing a network's readiness for voice, you need to determine how well the network handles the expected call volume. If the MOS estimate indicates low quality, it's time to upgrade and tune the 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 understand its capacity to see how many calls can be supported. Earlier in the assessment, you asked the local PBX management team for details on the peak number telephone calls and when these occur. You can use this information in a couple of ways.

First, use the numbers to determine raw bandwidth requirements for concurrent VoIP calls. If you want to support 10 concurrent VoIP calls using the G.711 codec with no silence suppression, you'll need 1.5893 Mbps of bandwidth to support these calls on a given network segment (10 x 158.93kbps – the total bandwidth consumption of the two RTP flows). Add this additional bandwidth requirement to the existing bandwidth usage of the network to set the new base requirement.

Second, use the numbers to do further testing. Replicate the test setup created above, but run the test for a one-minute period, a few times during the day where the assessment results showed heavy activity. Test five conversations at a; what happens to the MOS estimates? Next try ten, then twenty concurrent conversations. Plot the results on a graph; you should start to see the point where, as the number of calls increases, the quality decreases. Don't kill the 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 by using the value in the RTP payload type to get an

assured amount of bandwidth from end-to-end in the network. 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 datagrams. 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 effective. 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 the 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 based on 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. ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality."
2. ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning."
3. ITU-T Recommendation G.108, "Application of the E-model: A planning guide."
4. Walker II, J. Q. and J. T. Hicks. "Protocol ensures safer multimedia delivery," *Network World*, volume 16, number 44, November 1, 1999, page 53.
5. *Chariot*, by NetIQ Corporation. See www.netiq.com/Products/chr for additional information.
6. *Chariot VoIP Assessor*, by NetIQ Corporation. See www.netiq.com/Products/va for additional information.
7. Network Time Protocol version 3, RFC 1305, www.ietf.org/rfc/rfc1305.txt
8. "A Handbook for Successful VoIP Deployment: Network Testing, QoS, and More" www.netiq.com/Products/va/whitepapers.asp.

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 joined 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 many readers who helped improve this paper: Paula Acker, Steve Joyce, Scott Patterson, Susan Pearsall, Peter Schwaller, Kim Shorb, Carl Sommer, Jennifer Thomas, and John Wood.

Copyright Information