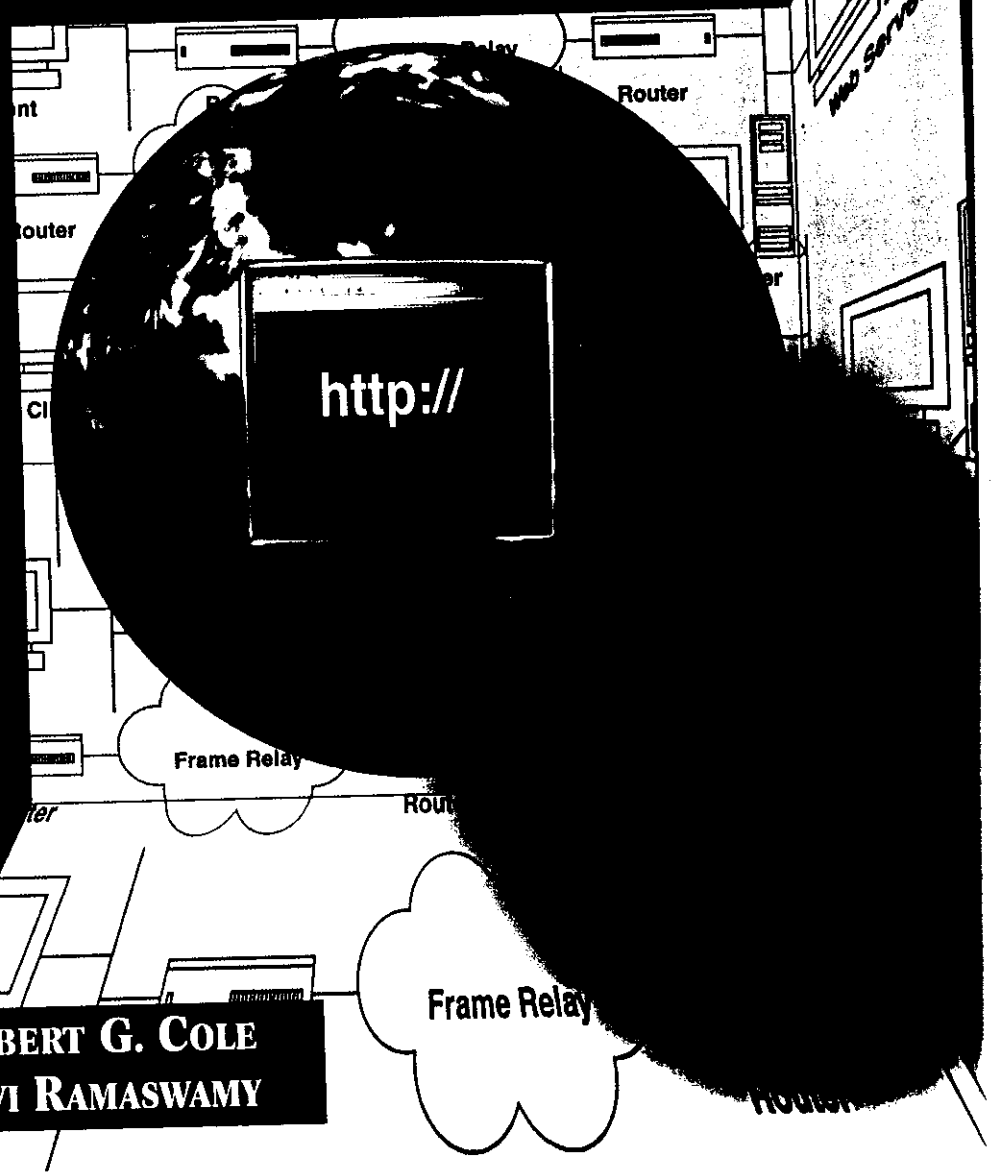
Network Performance Engineering



ROBERT G. COLE
RAVI RAMASWAMY

WIDE-AREA DATA NETWORK PERFORMANCE ENGINEERING

COLE • RAMASWAMY

ISBN 0-89006-569-1
9 780890 065693 90000>

ch House Publishers BOSTON • LONDON

Contents

Preface

This book has its origins in our data network performance engineering work at the erstwhile AT&T Bell Laboratories in the late 1980's and continuing into the better part of this decade in AT&T Solutions. Two distinct, but related, aspects of that work can be singled out as the major contributors to this book: the AT&T Datakit fast packet switch, and the rapid deployment of public frame relay services. Few in the networking industry have heard of Datakit, but it was one of AT&T's early attempts at providing a vehicle to use packet technologies to integrate separate wide-area networks. A handful of AT&T's large customers built private frame relay-like networks based on the Datakit family of switches and end devices. Subsequently, public frame relay services and multi-protocol routers gained popularity in the early 1990s. Wide-area network integration has since matured tremendously. The next phase of the integration, that is, combining voice, data, video, and multi-media applications over IP, is currently under way.

In the days of Datakit and early frame relay services, we spent a great deal of time and effort trying to understand and quantify the performance impact of integrating multiple protocols and applications on a packet network. In those days, the important protocols were IBM's SNA/SDLC and various bisynchronous and asynchronous protocols. Corporate networks also carried IP, Novell IPX, AppleTalk and others, initially over local area networks, but soon over wide-area networks. We soon realized that much of the battle in modeling application performance was in developing a detailed understanding of the packet processing within the protocol stacks and the packet flows across LANs and WANs. A thorough understanding of issues such as protocol overhead,

6

Using Pings for Performance Analysis

6.1 Introduction

Network managers and technicians usually resort to pings to measure latency, and FTP to measure throughput on a network connection, in order to draw conclusions about network performance. The network managers may be troubleshooting a specific performance problem or performing a network validation test.

To be able to draw the right conclusions about network performance from pings, one should compare the measured performance against expected performance based on calculations and some reasonable assumptions. In this chapter, we discuss how to calculate expected ping delays and how these calculations can be used to estimate WAN latency. We will also demonstrate how large pings can be used for rudimentary measurements of bursting available for PVCs in a frame relay network. We will discuss leased lines and frame relay WANs.

The calculation of expected throughput for a network connection using FTP is discussed in Chapter 7.

This chapter is organized as follows. In Section 6.2 we describe the ping program and discuss important aspects such as protocol overhead. In Section 6.3 we discuss ping delay calculation for leased line and frame relay connections. In Section 6.4, we will demonstrate how small pings can be used to highlight network latency issues. Section 6.5 discusses general issues and caveats in using pings for estimating network latency. In Section 6.6 we show how large pings can be used to demonstrate whether or not PVCs can burst above their CIR for a frame relay connection, and some overall issues in the use of large pings for calculating throughput.

6.2 Pings

We start with the ping program. (For an excellent and detailed discussion of ping and FTP and, in fact, of TCP/IP networking in general, see Stevens [1].) The ping program (named in analogy to a sonar ping) sends a number of Internet Control Message Protocol (ICMP) *echo_request* packets to a given destination. The destination is required by the ICMP protocol to reply with an ICMP *echo_reply* message. Pings can be sent between any two hosts and are often used to test network reachability.

In addition to reachability, the ping program returns the round-trip time, which, along with information such as ping size, line speeds, and distance, can be used to measure performance of the connection. The round-trip time is measured by the ping program at the source that clocks the time between when *echo_request* is initiated to the time when the *echo_reply* is received.

Figure 6.1 shows the output from the execution of the ping program. The first line shows the command line options when executing the program. The following lines show the program output. The program, in this case, continues until the packet count, indicated with the "–c 10" command line option,[1] is

```
>ping 10.10.10.10 -c 10 -s 56 -i 2
PING enskog (10.10.10.10): 56 data bytes
64 bytes from 10.10.10.10: icmp_seq=0 ttl=32 time=0.9 ms
64 bytes from 10.10.10.10: icmp_seq=1 ttl=32 time=0.8 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=32 time=0.7 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=32 time=0.8 ms
64 bytes from 10.10.10.10: icmp_seq=4 ttl=32 time=0.8 ms
64 bytes from 10.10.10.10: icmp_seq=5 ttl=32 time=0.8 ms
64 bytes from 10.10.10.10: icmp_seq=6 ttl=32 time=0.8 ms
64 bytes from 10.10.10.10: icmp_seq=7 ttl=32 time=0.7 ms
64 bytes from 10.10.10.10: icmp_seq=8 ttl=32 time=0.8 ms
64 bytes from 10.10.10.10: icmp_seq=9 ttl=32 time=0.8 ms
--- enskog ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.7/0.7/0.9 ms
>
```

Figure 6.1 Typical output from the ping program.

1. The example ping program discussed in this section is specific to the LINUX operating system. Other operating systems may have different command line options.

met. This is shown on the last line. The output lines list the destination address, whether the destination returns the ping, and, if it does return the ping, the measured value of the round-trip delay. The program provides the minimum/average/maximum round-trip delays of the successful pings at the conclusion of the program output.

The ping program is standard in all TCP/IP host implementations. One can use many options with the program. As an example, the ping program distributed with LINUX has command line options that allow the user to specify a number of variables when executing the program. These typically include:

- The number of *echo_request* messages sent (this is specified through the –c option in the command line);
- The interval between sending the ping messages (this is specified through the –i option in the command line and generally defaults to once a second);
- The size of the ICMP *echo_request* data field (this is specified through the –s option in the command line and generally defaults to 56 bytes of ICMP data); and
- Whether to allow for packet fragmentation or not (the default is usually not to allow for packet fragmentation).

These capabilities make the ping program a simple and extremely useful diagnostic tool.

One final piece of information regarding the ping program is required before we can put it to use in building a delay model of a network reference connection: message format and protocol overheads. This discussion is necessary in order to calculate insertion delays of the *echo_request* and *echo_reply* messages.

ICMP messages are carried directly inside IP packets. The ICMP echo message consists of an 8-byte header and is followed by the ICMP data. Within the ping program, the size of the ICMP data field is specified on the command line. Figure 6.2 shows an example of an ICMP echo message, which is fragmented. Link layer technologies, for example, Ethernet, token ring, and frame relay, have maximum frame sizes associated with them. The IP protocol addresses this fact through the implementation of IP datagram fragmentation. If a router or host wants to transmit a datagram onto a subnet having a maximum frame size smaller than the datagram, then the router or host may fragment the datagram into smaller multiple datagrams.

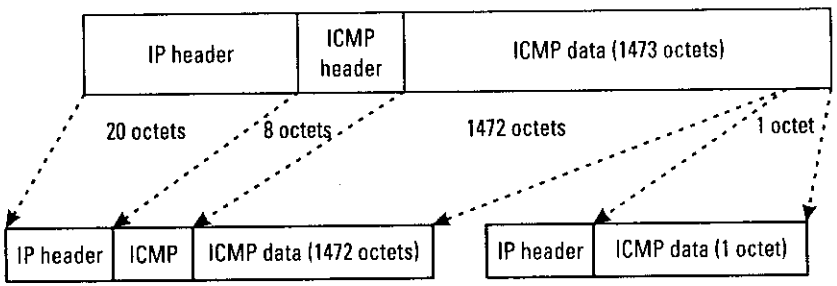


Figure 6.2 Message format for a fragmented ICMP echo message.

6.3 Calculating Ping Delays

We first consider a leased line connection and then a frame relay connection.

6.3.1 Leased Line Connection

Consider reference connection RC#1 shown in Figure 6.3. It shows a leased line connection between location A and B at 56 Kbps.

We will make the following assumptions for the ping calculation:

- The distance between the routers is roughly 1000 (airline) miles.
- Host A pings host B with a ping size of 100 bytes (i.e., ICMP echo packets are 100 bytes).
- The WAN connection is lightly loaded.
- LAN, router, and host delays are negligible.

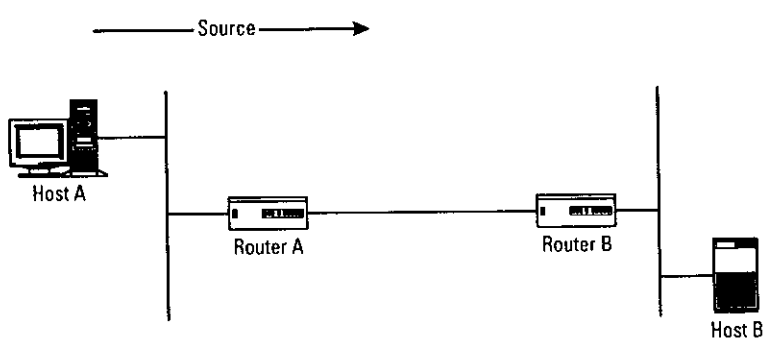


Figure 6.3 Reference connection RC#1 for calculating ping delay.

We now have enough information to calculate the expected ping delay. The following computations apply:

- Frame size = 100 + 20 (IP) + 8 (ICMP) + 8 (PPP) = 136 bytes.
- Insertion delay on the 56 Kbps link = 136 × 8/56,000 sec = 19 msec.
- One-way propagation delay (10 msec for every 1000 airline miles) = 10 msec.

Hence expected ping delay for this connection is 2 × (19 + 10) = 58 msec. The factor 2 accounts for the fact that the ICMP packets are echoed.

The total path traveled by the ping message, along with the associated delay estimates, is shown in Figure 6.4.

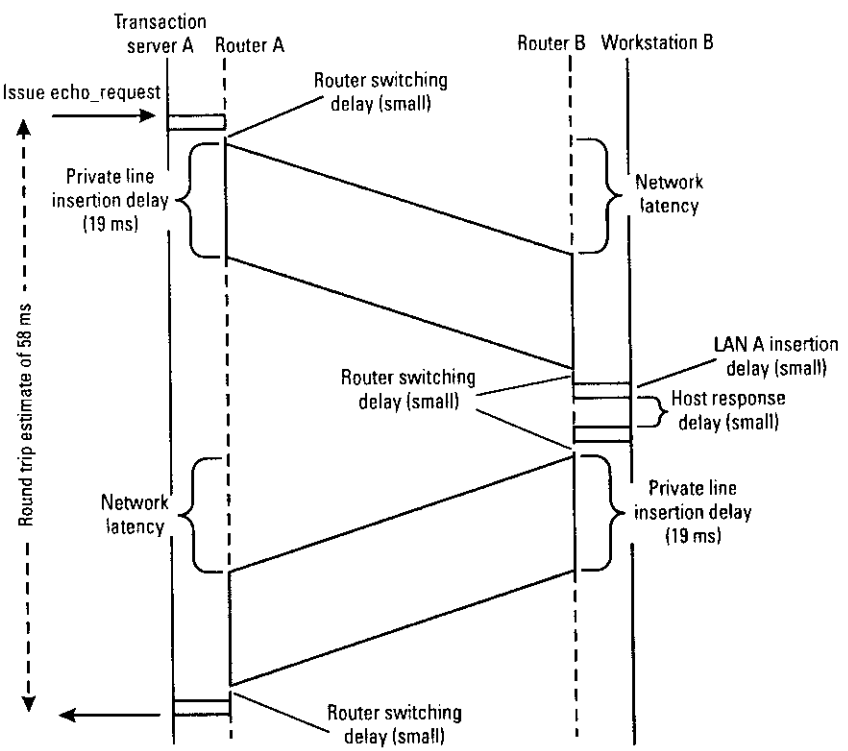


Figure 6.4 Calculation of the minimum ping delay for RC#1.

6.3.2 Frame Relay Connection

Consider a frame relay connection RC#2, as shown in Figure 6.5. Calculating ping delays for the frame relay case is a little more involved. One has to take into account issues such as pipelining, store-and-forward delays at the network edges, switch delays, and burst levels in the PVC.

For the sake of illustration, we will make the following assumptions:

- The frame relay connection is lightly loaded.
- Host A pings host B with a ping size of 100 bytes (excluding headers).
- The distance between the routers is roughly 1000 (airline) miles.
- The PVC is carried via four hops (five switches) and each switch adds 1 msec of latency.
- The carrier provides full bursting to the port speed without frame drops in both directions.
- LAN, router, and host delays are negligible.

The following computations apply for frame relay:

- Frame size = same as for leased line = 136 bytes.
- Insertion delay on 512 Kbps port = 136 × 8/512,000 sec = 2 msec.
- One-way network latency = 10 msec (1000 miles) + 5 msec (switch latency) = 15 msec.
- Insertion delay on 128 Kbps port = 136 × 8/128,000 sec = 8.5 msec.
- Expected ping delay = (2 + 15 + 8.5) × 2 = 51 msec.

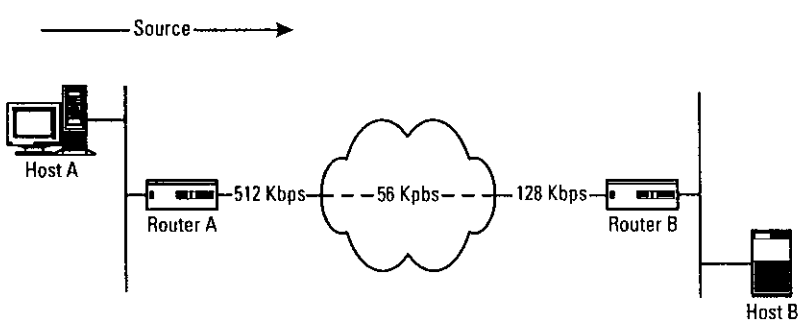


Figure 6.5 Reference connection RC#2 for calculating ping delay.

The calculation for the minimum ping for RC#2 is shown in Figure 6.6.

6.3.3 Observations

Note that propagation delay is a major contributor to the overall ping delay. For larger ping sizes, the relative contribution of the propagation delay will decrease.

The service provider (leased line or frame relay) should be able to provide the network latency (which includes propagation delay) number—in our case, 10 msec for leased line and 15 msec for frame relay. It is important to understand how the carrier measures latency—switch to switch or point of presence

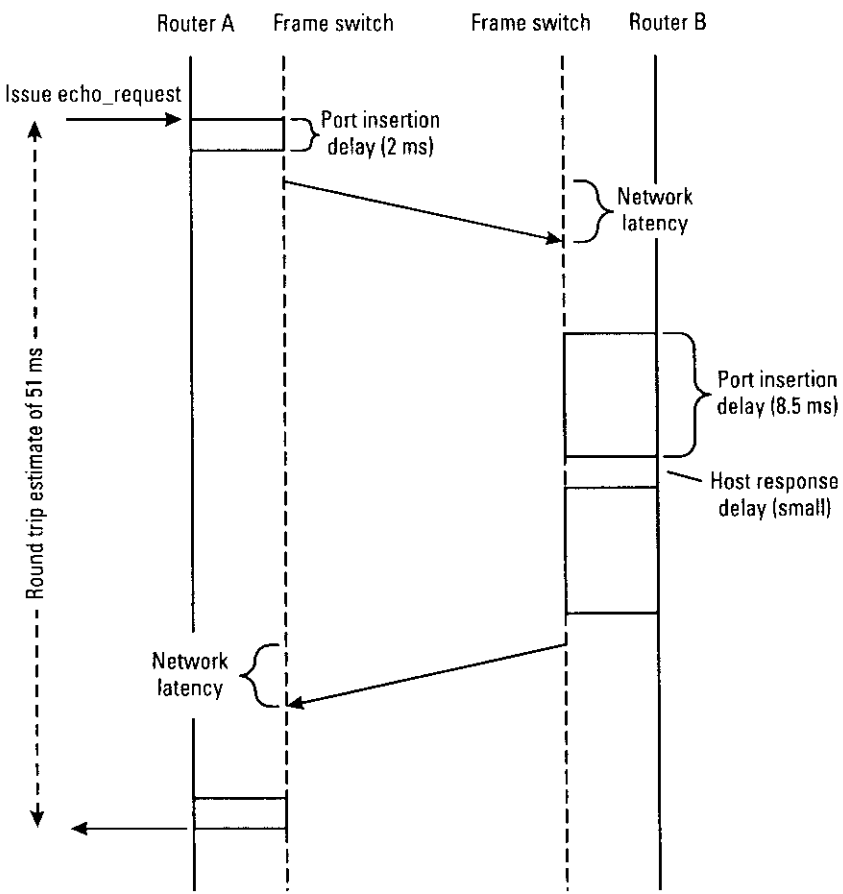


Figure 6.6 Calculation of the minimum ping delay for RC#2.

(POP) to POP. Most carriers advertise POP-to-POP delay objectives for frame relay, but cannot track these delays because not all POPs have frame relay switches. In other words, frame relay connections may have to be back-hauled to the nearest available POP.

Comparing the ping delay for the leased line and frame relay connections, we find that the leased line connection delay is actually longer. This is because we assumed full bursting to port speed. This offsets the extra delays over frame relay.

If, for some reason, the carrier limits the bursting to CIR in both directions, then the ping delay calculation needs to be changed slightly, as follows (assuming that frames are pipelined into the network using ATM cells):

- Insertion delay for a 136-byte frame in the A to B direction = $136 \times 8/56{,}000 + 0.015 + 136 \times 8/128{,}000 = 42.5$ msec;
- Insertion delay for a 136-byte frame in the B to A direction = $136 \times 8/56{,}000 + 0.015 + 136 \times 8/512{,}000 = 36$ msec; and
- Total ping delay = 80 msec (approximately).

If the underlying network is more complex (e.g., a hierarchical backbone of routers), then one must account for the fact that the forward and return paths may vary depending on what routes are chosen. In particular, the return path may not be the same as the forward path.

We have assumed that host and router delays are negligible, but they may not always be. One can use pings to estimate the additional latency due to these components. We show a simple method using the connection in Figure 6.3.

Suppose we send a ping from router A to host B. The ping delay components are (1) router processing delay at router A, (2) network delay, (3) router processing delay at router B, (4) LAN delay between router B and host B, and (5) host B processing delay.

To isolate the network delay component, one can first get independent measurements of router processing delay at router A and router B by sending pings between the local interfaces of the routers. These numbers can be subtracted from the ping delay between the two routers A and B. The remainder should be a reasonable estimate of the network delay. Next, if we were to subtract the router to router ping number from the router A to host B ping number, this gives an estimate of the host and LAN delays at the location B. Alternatively, one can ping router B from host B and subtract the processing delay on router B to get another estimate of the same metric.

One should perform these measurements a number of times to get an idea of the average and variability. Also note that the router processing times obtained thus are not representative of the overall processing delays in the router. For instance, an SNA packet requiring TCP/IP encapsulation may require more CPU cycles than pings. In addition, pings may be treated as lower priority packets for processing.

For the leased line case mentioned earlier, a simple way to account for background load would be to double the insertion delay. This implicitly assumes a 50% load on the WAN link in both directions and an M/M/1 queuing model [see (3.5) in Chapter 3]. If more accurate utilization numbers are available, then the following formula can be used to estimate the delay in any one direction:

$$\text{Insertion + Queuing delay} = \text{Insertion delay on the 56-Kbps link} / (1 - U)$$

where U is the utilization of the link in the direction in question.

For the reference connection in question, the assumption of 50% load in both directions will mean an increase in the ping delay estimate from 58 to 96 msec.

For the frame relay case, the calculation of ping delay under load is more complex and will not be discussed here.

6.4 Using Pings to Verify Network Latency

One useful application of the ping program is to ping between various location pairs within a data network and to compare the measured results to the expected delays. If the expected and measured delays differ significantly (say, more than 20%), then two possibilities arise—either the network is not performing as designed, or the assumptions underlying the calculations (for example, the link is assumed to have low load) are inaccurate. In both cases, investigations into the discrepancy should take place.

We first need to show how network latency can be estimated from the results of a ping.

In Section 6.3, we showed how ping delays can be calculated using some assumptions about distances and switch delays for an unloaded leased line or frame relay connection. Conversely, one can estimate the network latency given information about the ping delay, the ping size, and line speeds for the connection. Again, one would have to make some assumptions regarding LAN and router delays, and background load.

We will illustrate with two actual examples. These examples are taken from a real network with headquarters in Pittsburgh, Pennsylvania. Please also see Case Study 1 in Chapter 11.

Example 1: Pings Reveal That the Network Connection Is Healthy

The first example is a 512-Kbps leased line circuit between Pittsburgh, Pennsylvania, and Anaheim, California. The observed ping delay between the WAN routers at these locations for a 100-byte ping is 56/59/60 (min/avg/max). The connection is lightly loaded (peak utilization under 20%).

Let us use the minimum ping delay number, 56 msec, to calculate the best possible network latency. The average and maximum delays can be used to estimate representative numbers as well as a measure of delay variability.

The estimated network latency in this case is:

= Observed ping delay − 2 × Insertion delay
= 56 − 2 × (136 × 8/512,000) = 52 msec (approximately), or about 26-msec one-way delay.

Is this consistent with the fact that the connection is a leased line between Pittsburgh and Anaheim? The distance between Pittsburgh and Anaheim is about 2300 miles. Hence the propagation delay is expected to be about 23 msec one way. This is fairly close to the measured number and hence the conclusion is that there are no unusual latency issues in this connection, other than delays that could occur due to overutilized physical connections.

Example 2: Global Frame Relay Connection

The second example is an international frame relay connection between Pittsburgh and Australia. The frame relay ports at the two locations are 256 Kbps with a 128-Kbps PVC between them. As before, the connection is lightly loaded. For a 100-byte ping between the two WAN routers, the ping delay is measured as 384/390/404 msec (min/avg/max).

As in Example 1, the round-trip network latency between Pittsburgh and Australia can be estimated as

$$0.384 - (136 \times 8/256{,}000 + 136 \times 8/256{,}000) \times 2 \text{ sec} = 367 \text{ msec}$$

Is this reasonable? Rationalizing the latency using distances is very hard for global frame relay connections. For instance, even if we use a conservative distance estimate of 10,000 miles from Pittsburgh to Australia, the round-trip latency estimate is only 200 msec. The reason for the discrepancy is that global frame relay connections between the United States and other countries usually traverse three "clouds"—a U.S. domestic frame relay network, a global backbone, and a third frame relay network in that country. These clouds are connected via NNIs that typically clock at T1 or higher speeds. Hence there are several store-and-forward delays along with switch delays in the clouds.

Rather than attempting to validate the 367 msec, the best approach would be to repeat the ping tests multiple times and arrive at the network latency (which, for Australia, could well be in the vicinity of 350 msec).

6.5 General Comments Regarding the Use of Pings to Estimate Network Latency

While it is easy to use pings to estimate network latency, one should keep some key issues in mind.

Should We Use the Minimum, Average, or Maximum Ping Delay?

As we have seen, the min/avg/max delays are reported when a ping is successfully executed. If the ping delay exceeds a time-out value, then it is discarded from the min/avg/max delay reporting. The question is which numbers should be used for estimating network latency?

The minimum round-trip delay is well defined. In fact, it is the sum of all the fixed delays in the connection for the packet (as shown, for example, in the timing diagram in Figure 6.4). It is the best delay possible on the connection during the period of observation. Thus to obtain an estimate of network latency (i.e., delay on an unloaded circuit), one can use the minimum ping delay. In addition, if a large number of ping measurements result in a significant number of the round-trip measurements close to the minimum delay, then one can be fairly certain about the accuracy of the measurement of the minimum delay.

The other fixed delays include the switching delays in the routers, which tend to be relatively small in general, and the host response delay. One way to estimate (or back out) some of these delays is to run pings on intermediate points along the reference connection, for example, ping the routers along the path.

Finally, the maximum ping delay measurement, while giving some indication of the variation in the round-trip delays, is not very meaningful and will likely vary significantly with each separate run of ping measurements.

Effect of Background Load

The inference of network latency from ping delays are most accurate when the underlying connection is lightly loaded. By network latency, we mean switch delays (for frame relay) and propagation delay. Hence, as such, using ping to estimate network latency will assist in uncovering unusual problems in the

connection (such as excessive backhaul for the carrier's POP to the nearest frame relay switch). Drawing inferences about network latency using ping delays under load is very difficult and subject to error.

Using Ping Delays to Represent Application Delays

This is quite obvious, but it must be stated: Ping delays cannot be automatically used to infer how applications will perform over the connection. One would have to understand application transactions, message sizes, frequency, and so on, to quantify response times. However, ping delays (as mentioned before) gives an estimate of network latency, which can then be used to characterize application performance.

Pings May Be Treated as Low-Priority Packets

Some hosts, including routers, may treat ICMP echoes as lower priority packets, thereby potentially misrepresenting the delays. Allocating lower priority on a slow-speed WAN interface on a router can impact the delays more than if ICMP echoes are treated as lower priority packets in the processor. Some routers allow for explicitly treating ICMP echoes as high-priority packets.

Using Pings to Estimate Delays for Global Connections

Pings are especially useful when trying to estimate latency for global frame relay connections. Please see discussion in Chapter 5, Section 5.3.

6.6 Calculating Delays for Large Pings

Default ping sizes on most systems are small (less than 100 bytes). However, pings can be quite large, sometimes as large as 65 Kbytes. Small and large pings may be generated in response to performance problems, while the network manager is attempting to validate delays in the underlying WAN. Large pings can also be used quite effectively over frame relay connections to verify whether or not the carrier is allowing PVCs to burst, as we will demonstrate in this section.

If the ping size is less than the serial line MTU (maximum transfer unit) on the routers, then the ping delay calculation proceeds exactly as shown in the last section. For instance, if the ping size is 1000 bytes, then one would use a frame size of 1036 bytes and perform the calculations, since the serial line MTU on most routers defaults to 1500 bytes. If the ping size exceeds 1500 bytes, then the router will have to fragment the IP packet. The calculation of ping delays then becomes a little more involved, especially for frame relay, as shown in the following examples.

6.6.1 Example 1: Leased Lines

Assume the reference connection RC#1, a leased line connection with 56 Kbps of WAN bandwidth between locations A and B. Suppose an 11,832-byte ping is sent from A to B. Figure 6.7 shows the timing diagram for this ping packet. Due to the fact that the ping message is now larger than the MTU size on the Ethernet LAN segment, that is, 1500 bytes, the message must be fragmented into eight IP packets. The maximum data field of an Ethernet frame is 1500 bytes. From this we must subtract 20 bytes for IP header information within each Ethernet frame. This yields 1480 bytes per frame for higher level data. Finally, given that the ping message is fragmented, only the first fragment carries the ICMP header of 8 bytes. Therefore, our ping message should fragment into eight separate frames, the first with 1472 bytes of ping data and the last seven with 1480 bytes of data (or a total of 11,832 bytes).

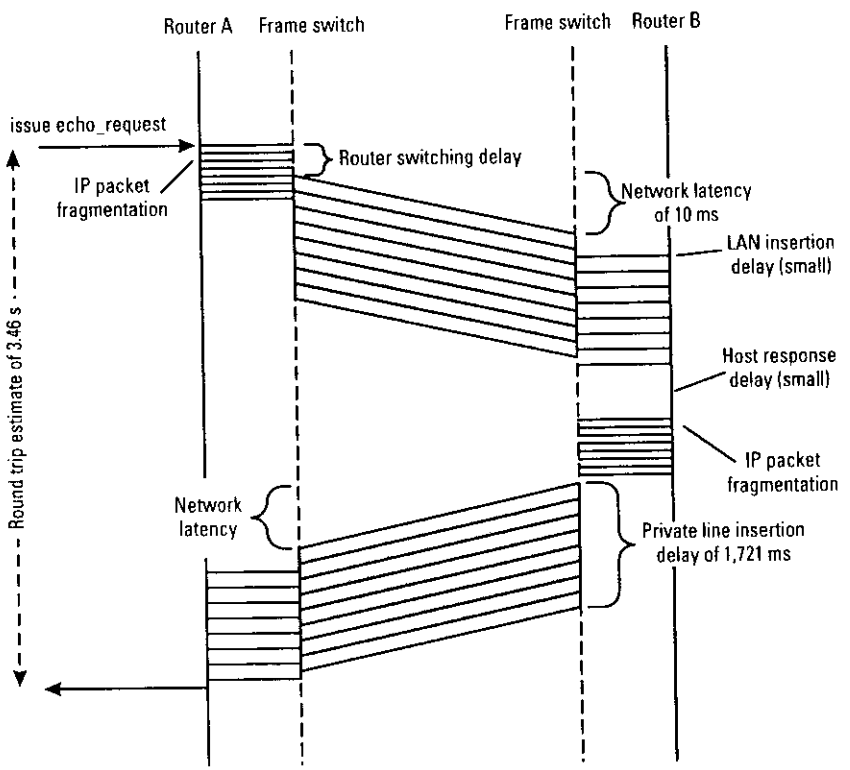


Figure 6.7 Timing diagram of a ping with IP packet fragmentation.

For large ping messages, the transmitting workstation performs the fragmentation necessary for the message to be transmitted over the local Ethernet segment. The eight separate IP packets pipeline their way across the reference connection. The destination host must accumulate all eight packets in order to reconstruct the entire ping message prior to issuing the ICMP *echo_reply* message. Because this reply is equal in size to the ICMP *echo_request* message, it must also be fragmented by the destination host. Again, these eight IP packets pipeline their way back to the original sender, are accumulated, and the message is fully reconstructed prior to stopping the ping round-trip timer.

Adding up all of these delays, we get a total round-trip delay of roughly 3.46 sec. Here are the details for the calculation:

Insertion delay for large packet = $8 \times [(1472 + 8 + 20 + 8) + 7 \times (1480 + 20 + 8)] / 56{,}000$ sec = 1.721 sec

Propagation delay = 10 msec

Total delay = 1.731 msec × 2 = 3.46 sec

6.6.2 Example 2: Calculating Large Ping Delays Over Frame Relay

This example is the same as Example 2 in Section 6.4.2 of this chapter. The frame relay connection is between Pittsburgh, Pennsylvania, and Australia with a 256-Kbps port at both ends and a PVC with 128-Kbps CIR connecting them. The connection is lightly loaded. The frame relay connection goes through two NNI links at T1 speeds (three frame relay networks). An 18,024-byte ping was issued from the Pittsburgh router to the router in Australia. The observed delay was 1688/1810/2052 msec. How does it compare with the target (i.e. calculated) delay?

The 18,024-byte payload will be broken up into 13 fragments—the first being 1472 bytes, the next 11 being 1480 bytes, and the last being 272 bytes. Each will be sent as an IP datagram with 20 bytes of header and 8 bytes for frame relay, with the first packet having an extra 8 bytes of header for ICMP echo. Hence the individual frame sizes over the WAN are 12 × 1508 plus 300 bytes for a total of 18,396 bytes.

The next step is to assume a number for network latency. We can use the same numbers from Example 2 in the previous section—367 msec. However, this number slightly underestimates the latency because the 1508-byte frame has to traverse the NNI links in a store-and-forward manner. Thus the increase should be $4 \times 1508 \times 8/1{,}536{,}000$ sec (four hops through a T1 for a 1508-byte frame at T1 speeds), which is equal to 31 msec. Hence the total network latency will be about 400 msec.

To calculate the expected ping delay, let us first assume that the connection is bursting to the full port speed (best case). For this case,

Delay from Pittsburgh to Australia = $8 \times 1508/256{,}000 + 0.2$ (one-way latency) + $8 \times 18{,}396/256{,}000$ sec = 822 msec

Ping delay = $2 \times 0.822 = 1.64$ sec

Notice how close the calculated ping delay is to the minimum observed ping delay—1.64 sec versus 1.69 sec, a 3% error. This is remarkable considering the fact that a number of assumptions have been made regarding the network and the latency.

What conclusions can be drawn from this observation? It is clear that the PVC is bursting to the full port speed in both directions (at least during the time the ping test was done). If it were not, the observed ping delay should be much larger. For instance, if there was no bursting at all on the 128-Kbps PVC, then the calculated ping delay will be

Delay from Pittsburgh to Australia = $8 \times 18{,}396/128{,}000 + 0.2 + 8 \times 300/256{,}000$ (last frame) sec = 1.36 sec

Ping delay = $2 \times 1.36 = 2.72$ sec

Clearly, the maximum observed ping delay is larger because of some intermittent background traffic skewing the average delay number. As stated before, the minimum delay number provides some insight into whether or not the PVC is bursting to port speed.

6.6.3 Some Comments Regarding the Use of Large Pings to Calculate Throughput

It is sometimes useful to perform a large ping test and use the results to make some low-level estimates of connection throughput. In the first example (Example 1) of this section, we were able to transfer 11,832 bytes in one direction in 1.73 sec (one-half of the ping delay). This assumes symmetry in the reference connection. (This is true in our analysis because we have assumed that there are no queuing delays along the path that would generate an asymmetry.)

Hence the calculated throughput is about 55 Kbps out of a maximum of 56 Kbps.

In Example 2 for the global frame relay connection, we found that a ping message with 18,024 bytes of data had a round-trip time of roughly 1.7 sec. Hence the connection transferred 18,024 bytes in one direction in roughly half the round-trip time, or 0.85 sec. Equivalently, the throughput is 18,024 bytes per 0.85 sec or 170 Kbps, out of a maximum of 256 Kbps.

Is this the best possible throughput for this connection? No. Throughput should really be measured by sending a continuous stream of packets in any one direction, as in a TCP bulk data transfer with a large enough window size. A large ping, no matter how large, is window limited (as in Novell burst mode), and therefore inappropriate for throughput calculations.

One other comment regarding determining burst levels in PVCs is in order. In Example 2, we were able to demonstrate that the PVC was indeed capable of bursting at port speed rate. We required that the connection be lightly loaded to facilitate the analysis. However, what is the guarantee that the service provider can sustain burst levels to port speeds when their backbone network is busy? This is an important question for most network managers. The only way to ascertain this is to continue to perform large ping tests at all representative periods of the day, and verify that the calculated ping delay is close to the observed minimum delay.

6.7 Summary

We discussed how pings work, how ping delays can be calculated, and how these pings can be used to estimate network latency and throughput. We showed that small pings can be used to estimate network latency and that large pings can be used to estimate throughput over a WAN connection, especially bursting over the CIR for frame relay. We also discussed some caveats for using pings to estimate network latency and throughput.

Reference

[1] Stevens, W. R., *TCP/IP Illustrated, Volume 1: The Protocols*, Reading, MA: Addison-Wesley, 1994.

Part II
Specific Application/Protocol Suites