ACCURACY AND PRACTICALITY OF TECHNIQUES FOR MEASURING WEB EXPERIENCE OF DIAL-UP USERS

Eric Siegel, Keynote Systems
Jing Zhi, Keynote Systems

A set of experiments on widely different web sites examined the technical accuracy and practicalities of different approaches to using an emulated dial-up link instead of true 56kbps (V.90) dial-up to measure end-user web site experience. The experiments showed that attempts to emulate V.90 dial-up connections by using simple calculations, bandwidth-restricted routers, or sophisticated network simulation packages are extremely inaccurate and unusable for comparing different web pages. The primary reason for these inaccuracies appears to be the modem's hardware compression (V.42bis) feature.

EXECUTIVE SUMMARY [The paper was first published by Computer Measurement Group in the Proceedings of the 27th Annual Conference, December 2-7, 2001; that version did not contain an Executive Summary.]

Measurement Agents are devices that generate requests for web pages by using a standard browser and a script. A set of experiments was performed to compare the measurements made by four different types of Agents:

- High-bandwidth (over an unrestricted 45 megabit/second connection),
- Low-bandwidth emulation (using simple router commands to restrict bandwidth),
- Sophisticated emulation (using a network emulation package to model dial-up network behavior), and
- True dial-up through a local telephone exchange into the local Point of Presence (POP) of a consumer ISP.

Five widely different web sites were measured, to see if the differences in page design would influence any discrepancies between the different measurement methods.

The study showed that:

- Attempts to emulate V.90 dial-up (56 kbps) connections by using bandwidth-restricted routers or sophisticated network simulation packages are extremely inaccurate and unusable for measuring web pages and transactions.
- These findings imply that using a simple "multiplier" to calculate dial-up behavior from high-bandwidth behavior will also be extremely inaccurate.
- The deviation is primarily caused by the modem's hardware compression feature and cannot be easily predicted, because it is dependent of the particular design of the page: the proportion of uncompressible images to compressible text (typically HTML); the use of secure pages, etc.

Because of these findings, *emulations or "multiplier" calculations should not be used to compare the end-user's experience of different web pages or different versions of a web page.* The emulated measurements performed in this study had deviations as high as 45% from the true dial-up measurements; different sets of pages could easily have even greater deviations from the correct measurements.

The following table summarizes the results:

	Sophisticated Agent Discrepancy	Low-Bandwidth Agent Discrepancy
Secure Site	– 30% (too fast)	– 45%
Site A	– 18	– 27
Site C	– 2	– 15
Site D	+ 3	– 8
Site B	+ 18% (too slow)	– 2

Deviations from true dial-up measurements by the emulations (Geometric means)

INTRODUCTION

We performed a study of the technical accuracy and practicalities of different approaches to using an emulated dial-up link instead of true V.90 56kbps (kilobit/second) dial-up. Emulation is much less expensive than true dial-up, but it's also less accurate. The goal of this evaluation was to estimate that loss of accuracy and uncover the basic reasons for any inaccuracies.

We took our measurements by using four different types of measurement Agents, which are devices that generate requests for web pages by using a standard browser and a script:

- High-bandwidth (over an unrestricted 45 megabit/second (mbps) connection), used for comparison only;
- Low-bandwidth emulation (using simple router commands to restrict bandwidth);
- Sophisticated emulation (using a network emulation package to model dial-up network behavior); and
- True dial-up through a local telephone exchange into a local Point of Presence (POP).

We measured five widely-varying website pages over those four types of Agents, to see the influence of different website designs on the accuracy of the emulations. Those pages included:

- Simple page with few images
- Complex page with 29 images, JavaScript, and Cascading Style Sheets (CSS)
- Page using a Content Distribution Network (CDN) and high proportion of HTML
- Secure Page

Our results are presented in the following sections:

- Experimental Design
- Technical Background
- Findings and Summary

EXPERIMENTAL DESIGN

Four different measurement Agents were configured in our lab in San Mateo, California. All of them ran simultaneously, and the directly-attached Agents used the same set of gateway routers to enter the Internet and therefore encountered similar Internet congestion conditions. The dial-up Agent dialed into a local ISP's POP in San Mateo. All Agents used a Microsoft Internet Explorer 5.0 measurement core. JavaScript, cascading style sheets (CSS), secure pages, and any other features handled by Explorer 5.0 were completely measured.

High Bandwidth Agent

The High Bandwidth Agent was a standard Keynote transaction Agent located in San Mateo; it used our normal corporate 45 mbps ("T3") access to the Internet in San Mateo.

Low Bandwidth Agent

The Low Bandwidth Agent used a dedicated Cisco router to feed into our normal corporate 45 mbps access to the Internet. This dedicated router was configured to provide exactly 56 kbps into the Internet ("uplink") and out of the Internet ("downlink"). This is a simple technique that's available on most routers, although it is surprisingly easy to misconfigure.

Sophisticated Agent

The Sophisticated Agent used Shunra Software's "Cloud" network emulation package (see www.shunra.com) that was inserted between the

Agent software and our normal corporate 45 mbps access to the Internet. This package was configured to insert a constant 50 ms. delay in each direction and to restrict the bandwidth in the two directions. (The 50 ms. delay was chosen as a result of literature searches and some experimentation.) The normal dial-up PPP (point to point protocol) headers were used, instead of Ethernet headers. No errors or packet losses were inserted into the link.

The uplink and downlink bandwidth were set separately and were varied during the experiment to see the effects of that variation.

The Sophisticated Agent system allowed the packet size to be varied, and we experimented with two sizes: the 1500-byte packets that are always used on Ethernet and are sometimes used on dial-up (usually for Windows/NT and Windows/95), and the 576-byte packets that are sometimes used on dial-up (usually for Windows/98 and Windows/ME).

Compression facilities were not available on the Agent. Modem hardware compression, described later, can have a considerable effect on throughput rates, but it's not practical to provide that compression in software. The compression algorithm, although well-known and standardized as ITU-T Standard V.42bis, is apparently only available in modem hardware. Most of the experimentation with various uplink and downlink speeds was an attempt to compensate for this lack of V.42bis compression.

Dial-Up Agent

The Dial-Up Agent contains a Microsoft Explorer IE 5.0 measurement core and a standard, home-style Diamond Multimedia SupraMax V.90 modem inside the PC, using the PC's PCI bus. The top-rated SupraMax is a "winmodem" typical of the modems bought by home users; unlike professional modems, it uses the PC's CPU (An Intel Celeron 533 MHz CPU in the Agent) for some of the modem's operations. It was attached to a standard analog dial-up line in our San Mateo lab, and it made dial calls into a San Mateo POP.

Measured Web Sites

Five different, public web sites were selected for measurement, based on our desire to have a broad sample of different types. These are shown in Table 1.

Measurements were taken every 15 minutes from April 6 through April 16, 2001, with interruptions for reconfiguration and for re-initialization of the experimental setup. All the page information was obtained on April 9, 2001.

	Secure Site	Site A	Site B	Site C	Site D
Server Location	Phoenix, AZ	Washington D.C	New York, NY	Washington D.C	Washington D.C
Base Page Size	9,358	6,432	41,243	18,948	15,739
Content Size	8,430	48,481	41,343	64,275	70,351
Total Size	17,788	54,913	82,586	83, 223	86,090
Number of Objects	3 Images	9 Images	43 Images	17 Images 5 Javascripts 1 css	29 Images 2 Javascripts 1 css
Page Feature	Secure Page	Javascript / css	CDN (Akamai)	Javascript / css	Javascript / css

Table 1: Measured web pages

TECHNICAL BACKGROUND

A dial-up link using a pair of V.90 ("56 kbps") dial-up modems is much more complex than a direct connection over a dedicated link or over Ethernet. Some of these differences can be precisely emulated, while others can only be approximated. They include:

- PPP headers and protocol
- Asynchronous byte length
- Serialization delay
- Modem electronics delay
- Latency on a local dial-up link
- Maximum Transmission Unit (MTU) and Maximum Segment Size (MSS)
- V.42 error correction and the LAP-M protocol
- V.42bis compression
- V.90 asymmetric transmission speeds
- Use of dial-up Access POP instead of direct-link ISP connection

We'll discuss these in detail, followed by a summary of the relative importance of the factors and the methods we used to experiment with them. The next section, Analysis, presents and discusses the results of our experimentation.

PPP Headers and Protocol

A serial link normally uses the Point to Point Protocol (PPP); an Ethernet link uses Ethernet framing. An Ethernet frame uses approximately 50 more bytes per frame than PPP, as PPP can reduce the size of the TCP/IP headers through header compression. However, the difference in length between Ethernet and PPP is small when compared to the length of the typical 1500-byte frame. Our emulations did not, therefore, take these differences into consideration.

Asynchronous Byte Length

Dial-up modems are asynchronous; that is, their serial link between the modem and the computer uses ten-bit bytes instead of the normal 8-bit bytes, and there isn't an accompanying clock signal on a separate physical line. However, most home computers and professional dial-in access units no longer use an external modem with the accompanying serial link. Instead, they normally use an internal card that's connected through the parallel bus within the PC or access unit. Therefore, the issue of ten-bit bytes on the serial link does not appear.

The byte length on the telephone line for older asynchronous modems is ten bits per byte, but all modern modems use V.42 error correction by default. V.42 error correction, discussed later, results in eight-bit bytes on the telephone line. We can therefore assume that the byte length on the telephone line for the typical home user is eight bits. Any issues of ten-bit bytes can therefore be ignored.

Serialization Delay

The modem must feed the data bits onto a serial line, and the speed of that output to the line or input from the line is determined by the line speed and is called serialization delay. For example, it takes 0.1 second to feed 600 bytes (4,800 bits) onto a 48,000 bps line.

Modems that are external to a PC or access unit may incur an additional set of serialization delays between them and the connected device, but we're assuming that modems are connected to an internal parallel bus, as discussed above. We therefore do not need to consider those additional delays.

Latency through Modem Electronics, the Local Dial-Up Link, and the PC Driver

There are considerable delays while passing through modem electronics, and those delays do not exist for Ethernet or for high-speed digital communications links. The winmodem used by us (and by the majority of home PCs) uses the PC's CPU (a 533 MHz Intel Celeron in the measurement agent) for some computations. A typical pair of modern modems inserts upwards of 40 ms. one-way latency; some modems insert 100 ms. or more. [GOOD99]

There is also latency through the wiring and electronics of the local telephone exchange, and that latency does not appear for an Ethernet connection. The ITU-T G.114 Recommendation estimates this latency at 12 ms. one-way for an analog exchange and 3 ms one-way for a digital exchange. (Delays for long-distance connections are considerably higher; for that reason, use of a long-distance connection between a measurement Agent and its Internet POP is usually not recommended.)

The total of these delays is particularly important for short files, as each file transfer begins with TCP's Slow Start Algorithm. The Slow Start Algorithm is extremely sensitive to round-trip latency, because it uses that round-trip latency to govern the speed at which it gradually increases bandwidth use. TCP also uses round-trip latency to detect missing packets and to regulate the speed of its recovery from missing packets, so it's important over noisy transmission lines. (See [RFC 2581].)

The sophisticated Agent included a 50 ms. latency in each direction, as discussed in the Findings section; the low-bandwidth Agent could not.

Maximum Transmission Unit (MTU) and Maximum Segment Size (MSS)

Each packet sent on a transmission line is restricted to a maximum length set by the media or the protocol; this is the Maximum Transmission Unit (MTU). It is normally 1500 bytes, but some common home computers use 576 bytes.

- The original TCP specification gave 576 bytes as the default MTU; systems were not required to accept anything larger.
- Windows/98 and Windows/ME normally use an MTU of 576 bytes on dial-up lines that are slower than 128 kbps.
- Windows/NT and Windows/95 use an MTU of 1500 bytes on all dial-up lines.
- The dial-up links used by many of the access servers that are accessed by home computers use an MTU of 1500 bytes.
- Ethernet's MTU is always 1500 bytes.

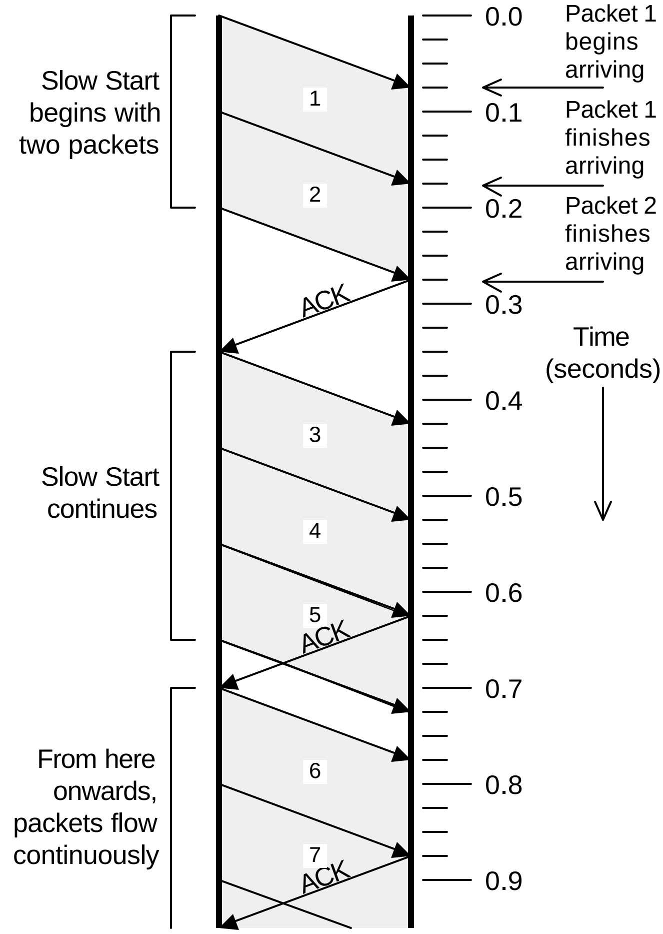
An MTU of 1500 bytes restricts the size of each TCP segment (the Maximum Segment Size, or MSS) to 1460 bytes, after subtracting the normal TCP and IP headers. An MTU of 576 bytes results in a TCP MSS of 536 bytes.

Note that the MTU is not symmetric; it restricts only traffic that's being *transmitted*. However, some TCP stack software examines the MTU associated with the hardware that it is using, and the TCP stack then places an MSS option into the outgoing TCP connection request. That option asks the remote TCP station to use the specified MSS, which has the effect of restricting the size of TCP segments that are *received*.

The practical result of these situations is:

- Directly-connected links (T-1, etc.) almost always use an MTU of 1500 bytes in both directions.
- Dial-up links using Windows/NT or Windows/95 almost always use an MTU of 1500 bytes in both directions.
- Dial-up links using Windows/98 or Windows/ME almost always use an MTU of 576 bytes in both directions, because they also request an MSS of 536 in their TCP connect.

Preliminary analysis shows that the difference between the two MTU sizes should be small for a V.90 dial-up line. The reason is shown in Figure 1, Dial-Up Line with 600-byte Packet, and Figure 2, Dial-Up Line with 1500-byte Packet. Both of these figures assume 75 ms. one-way latency (50 ms. one-way latency through a pair of modems and the local communications line, plus 25 ms. one-way latency through the Internet to the server), and both assume 48 kbps transmission rate from the server and 30 kbps to the server (these are the typical speeds we saw in our lab, as discussed later). The figures also assume full-duplex, without another, concurrent, traffic flow that could interfere with the timing of the ACK packets. The 600-byte packet is representative of an MTU of 576 bytes with a few bytes of framing overhead; the 1500-byte packet is representative of an MTU of 1500 bytes with a few bytes of overhead. (All these numbers are approximations, but they make the calculations simpler without concealing the impact of the major differences.)



Assumptions:
- 75 ms one-way latency
- 600 byte packet
- 48 kb/s downlink
- 30 kb/s uplink

Figure 1: Dial-Up Line with 600-byte Packet

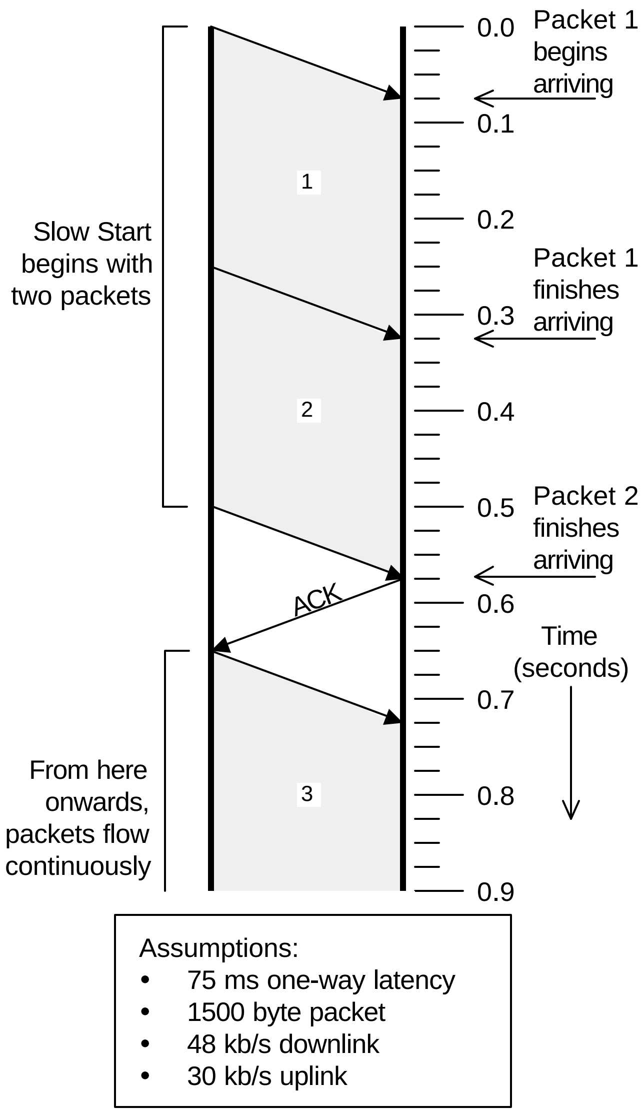


Figure 2: Dial-Up Line with 1500-byte Packet

Figure 1 shows that it takes 0.075 seconds for the first bit to make its way from sender to receiver, and it then takes 0.1 seconds for the entire set of 600 bytes to finish arriving. In contrast, Figure 2 shows that it takes 0.25 seconds for the entire set of 1500 bytes to finish arriving.

As can be seen, the only real difference between the two figures is the small gap between packets 5 and 6 in Figure 1. That gap is present because of the operation of TCP's Slow Start algorithm, and it would be larger if the latency were larger. In both figures there's no difference in performance after the link is filled and packets flow continuously. Differences in packet size should therefore produce an effect only for very short files, and that effect should be very small.

There is a possibility that, in actual use, the concurrency of parallel TCP flows and the asymmetric link speeds would interfere with ACK spacing and make the frame size more important than this preliminary analysis shows. Errors and retransmissions would also have an effect. We made a few measurements with both packet sizes in our lab, but the results were inconclusive because the test was too limited (only 36 data points). We did not see any large effects in those tests, however, and we used 1500-byte packets in our major experiments.

V.42 Error Correction and the LAP-M Protocol

Modern modems use V.42 error correction to ensure that bits sent between the modems don't contain any errors. This protocol, which is invisible to the modem's users, is similar to the old HDLC protocol. It uses LAP-M framing, which adds six bytes per frame and can carry up to 128 bytes in each frame. Up to 15 frames (1,920 bytes) can be transmitted before an acknowledgement is required. On a 48,000 bps link with 50 ms. one-way latency, there's almost no chance that transmission will be forced to pause while the protocol waits for acknowledgements. (This is another reason to avoid using long-distance dial-up lines when measuring web performance.) The impact of V.42 therefore does not need to be considered on low-noise lines. V.42 uses eight-bit bytes, avoiding the ten-bit bytes used by older asynchronous transmission methods.

V.42bis Compression

Modern modems use V.42bis compression along with V.42 error correction. This compression, invisible to the modem's users, tries to compress data before placing it on the telephone line. It monitors the success of the compression, and, if it discovers that a compressed data string is longer than the uncompressed string, it will discontinue compression for that string. As virtually all images and streaming media are already compressed before they reach the modem for transmission, V.42bis compression is usually not active for them. However, V.42bis is useful in transmitting plain text, such as unencrypted, uncompressed HTML. The compression of such data can result in an effective increase in transmission speed for plain text, *but not for compressed or encrypted data*, of anywhere from two to four times the raw telephone line speed.

Unfortunately, it's completely impractical to create an exact emulation of a modem's hardware compression feature. There is no commercial software that we could find that incorporates the algorithm.

V.42bis is a key difference between any emulation and an actual modem connection. Although it is possible to disable V.42bis by using a modem control string, it's safe to assume that home users never do this. Therefore, *the need for an effective way to emulate the operation of V.42bis is a major factor in building any accurate emulation*.

In our testing, we have tried to tune the emulated dial-up line to compensate for the missing V.42bis compression function. We've done that by increasing the line speed in both directions, as is described below and in the Findings section.

Clearly the uplink speed should be increased by a greater percentage than the downlink speed, as browsers and measurement Agents usually transmit unencrypted plain text. The Agents usually transmit simple HTTP GET and POST commands. They never transmit images, and transmission of encrypted (e.g., HTTPS) data is relatively infrequent compared to the total number of bytes that an Agent transmits. Our measurements have indicated that our modems are doing approximately a 3:1 compression on the uplink for the sites in this series of tests.

The downlink speed is much more of a problem. The compression depends completely on the data being transmitted. Sites with compressed or encrypted HTML would therefore have virtually *no* additional compression in the modem, whereas sites with long, uncompressed HTML files and relatively few graphics would have up to 3:1 or higher degrees of compression. Our measurements have indicated that our modems are doing approximately a 1.5:1 compression on the downlink *for the sites in this series of tests*. (Remember, this is the combined total; individual sites may vary widely, depending on their content.)

The new V.44 modem compression standard, discussed below, will increase this problem because it will increase the compression ratio for uncompressed, unencrypted data.

V.90 Asymmetric Transmission Speeds

The V.90 home modems are asymmetric; they can transmit up to 53 kbps towards the home modem and up to 33.6 kbps away from the home modem. (The "home modem" is on the analog end of the connection; the other end must be a digital connection to the access provider. Otherwise, the modem will transmit 33.6 kbps maximum in both directions.) In practical use, our company has found that most of our downlink V.90 connections are approximately 48 kbps. The literature indicates that uplink connections tend to be at approximately 30 kbps if the downlink connection is at a high speed.

The new V.92 modem standard increases the uplink speed up to 48 kbps, but that won't greatly affect dial-up measurements, as browsers and Agents usually send very little data on the uplink. *Unfortunately, the new modems also include the new V.44 compression standard, which may increase compression ratios and thereby exacerbate the problem of emulating V.42bis compression of uncompressed, unencrypted data*.

Dial-Up Access POP vs. Direct Link Connection

The daily pattern of congestion and resulting performance changes on a dial-up access network may be different that that on high-bandwidth connections into the Internet. This is probably most marked for specialized networks, such as AOL, that do not always connect directly into the Internet fabric with the smallest possible number of hops and latency. Rigorous evaluation of that possible difference was beyond the scope of this paper; however, we have seen some evidence of that difference in the past.

Summary of Differences

As we've now seen, the critical differences between Ethernet connections and dial-up connections fall into the following areas:

- Serialization delay
- Latency through modem electronics, the local dial-up link, and the PC driver
- V.42bis compression
- V.90 asymmetric transmission speeds
- Dial-up access POP vs. direct link connection

Of these, serialization delay can be emulated through the use of a bandwidth limiter with infinite buffers. Standard routers, bandwidth-restriction devices, or specialized transmission-line emulation software or hardware packages can accomplish that function. Bandwidth restriction is a common application for routers at ISPs, as ISPs need to restrict customer bandwidth for fractional T-1 links to the amount that the customer is paying for. The fact that it's a common application doesn't mean that it's simple to configure, however; we have seen surprising cases of misconfigured bandwidth restriction devices used by measurement Agents.

Modem delay and latency on a local dial-up link primarily affect the Slow Start phase of each file transfer. After Slow Start is over, data will flow at a steady rate with the dial-up link filled to capacity, as shown in Figures 1 and 2. However, delays longer than the estimated 75 ms. one-way (not at all unusual on the Internet) or any errors (which may force another slow start) will greatly increase the effect of these delays. These cannot be emulated by a router or by a bandwidth-restriction device; they must be emulated by a specialized transmission-line emulation software or hardware package. For our sophisticated emulation, we used a transmission-line emulation software package (Shunra Software's "Cloud") that was set to 50 ms. latency in each direction.

V.42bis compression is a major problem for uncompressed text files. We ignored the problem with the low bandwidth emulation and attempted to compensate, very roughly, for the absence of compression by boosting bandwidth in the sophisticated emulation. This problem should not occur for images or for encrypted files, as they're uncompressible; but it will greatly affect standard Web pages.

Unfortunately, it's not easy to create a multiplier factor for a Web page, as it depends on the precise mix of compressible and uncompressible data. A simple change to the page makes the multiplication factor completely wrong. This would totally invalidate the primary use of dial-up measurements: comparing different versions of a page. Additionally, using a single multiplication factor for a transaction that moves between different pages, some uncompressible (e.g., a secure page) and others highly compressible (e.g., unsecured, with a large amount of HTML) will clearly give completely incorrect results.

The asymmetry in V.90 modem speeds is being handled, as described above, by using different bandwidths in the sophisticated emulation. It's being ignored in the low bandwidth emulation, although it's possible to configure a router to provide different bandwidths in the two directions.

Finally, no emulation can recreate the possible difference in the daily pattern of congestion and resulting performance changes between a high-bandwidth connection to the Internet and a V90 dial-up link to specialized access networks, such as AOL.

FINDINGS

This section summarizes our findings.

Neither the low-bandwidth (bandwidth-restricted) nor the sophisticated (simulated network) emulation should be used to compare total download times of different web pages or to compare different versions of the same page.

Unfortunately, it appears that the total download times can be greatly affected by modem compression, which neither the low bandwidth Agent nor the sophisticated Agent could emulate. As discussed in the Technical Background section, that compression varies greatly depending on the design of the web page. For example, secure pages (such as for our Secure Page example) are completely uncompressible, as are (untested) streaming media downloads; HTTP is usually compressible (unless it has already been compressed, as is done by some web servers and by products such as BoostWorks); images are uncompressible because they've already been compressed.

The classical way that emulations have tried to deal with the compression problem is by increasing the effective bandwidth. *The difficulty is that this "tuning" is completely dependent on the particular design of each separate web page and the exact pattern of the bytes in the compressible sections.* For example, a secure page, which is uncompressible, would be given a 30 kbps uplink and a 45 kbps downlink, whereas a pure text page might be given a 90 kbps uplink and a 135 kbps downlink! (The compression ratio varies between 2:1 and 4:1 depending on the data pattern.)

We selected 90 kbps uplink and 60 kbps downlink for the sophisticated Agent to use on the set of pages in this study. We selected those numbers by experimentation and by using statistics from our dial-up Agent, which told us the compression ratios being used by the real dial-up Agent's modem. This yielded the best results for most of our pages, but, of course, a completely uncompressible page (our Secure Page example) had very poor results. *A different set of pages would require a different choice of bandwidths, and the spread between the emulated and actual measurements might be very different.*

Figure 3 shows the overall results of our study. (These are geometric means and "geometric deviations," which are much more accurate and reasonable for Internet statistics than arithmetic means and standard deviations. A geometric mean is the nth root of the product of the n measurements; the "geometric deviation" is the equivalent of the standard deviation, except that it's in log space. It can be calculated as: e**{std deviation of (ln of individual values)}, and it is used as a factor in creating error bars in a graph. Note that the resulting error bars are not symmetrical.)

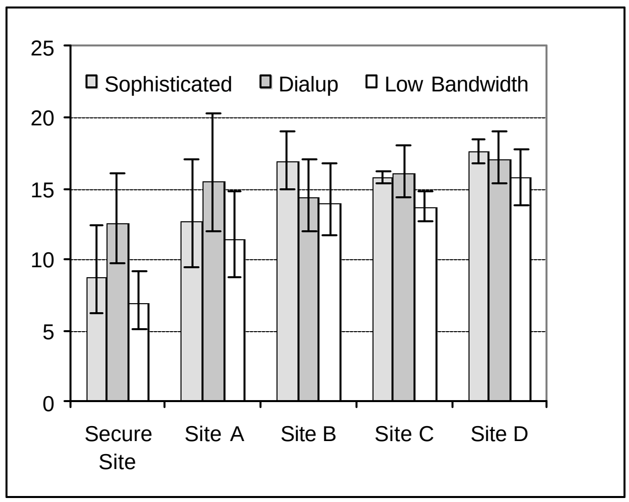


Figure 3: Comparison of Geometric Means and Geometric Deviations of Total Download Times (seconds)

The differences among the web sites are clearly visible:

- The Secure Site, an https: secure page, is poorly emulated by the upload and download speeds because it is uncompressible. The emulated speeds, which try to compensate for modem compression that isn't used in this case, overshoot the mark considerably. The resulting deviations from the correct measurement are greater than 30% (sophisticated) and 45% (low bandwidth).

- Site B uses Akamai and 43 images; it also has a huge HTML file, especially when compared to the total size of the page. (Half of the total download file is compressible HTML "Base Page"; see Table 1.) Therefore, the Site B page undergoes more compression by the true dialup modem than by the emulated "compression" in the sophisticated Agent.
- Note that in some cases the sophisticated emulation is too fast, in others it is too slow. This is consistent with the fact that the emulation depends on an estimate of latency and of the improvement caused by modem compression.
- The low-bandwidth emulation is always too slow, but by widely varying amounts. This is because our particular low-bandwidth setup used symmetric speeds for both uplink and downlink. Tuning those speeds would result in improvements, although it would not be as good as could be obtained by tuning the sophisticated emulation.

The error percentages are summarized in Table 2. Note that the total error variation is slightly greater for the sophisticated Agent than for the low bandwidth Agent. We don't believe this is meaningful, and they are both unacceptably large (approximately 45%). The sophisticated Agent, as we'll see in the next point, has much more accurate "First Packet" times than the low bandwidth Agent.

	Sophisticated Agent Discrepancy	Low-Bandwidth Agent Discrepancy
Secure Site	– 30% (too fast)	– 45%
Site A	– 18	– 27
Site C	– 2	– 15
Site D	+ 3	– 8
Site B	+ 18% (too slow)	– 2

Table 2: Deviations from the true dial-up measurements by the emulations (Geometric means)

The sophisticated Agent does a good job of providing first-packet times for non-secure web sites; the low bandwidth Agent is useless for this task. This first-packet behavior implies that the low bandwidth Agent is less accurate than the sophisticated Agent for web pages that have a large number of small elements for which TCP's slow-start is important.

The first-packet times depend very strongly on round-trip latency, which was emulated by the sophisticated Agent and was not emulated by the low-bandwidth Agent. We used 50 ms. each way, and that appears to have been a reasonable choice. (It may be very slightly high; additional tuning might lower this latency to approximately 45 ms.) As is seen in Figure 5, the first-packet times for all sites except the Secure Site are emulated well by the sophisticated Agent and are very poorly emulated by the low-bandwidth Agent.

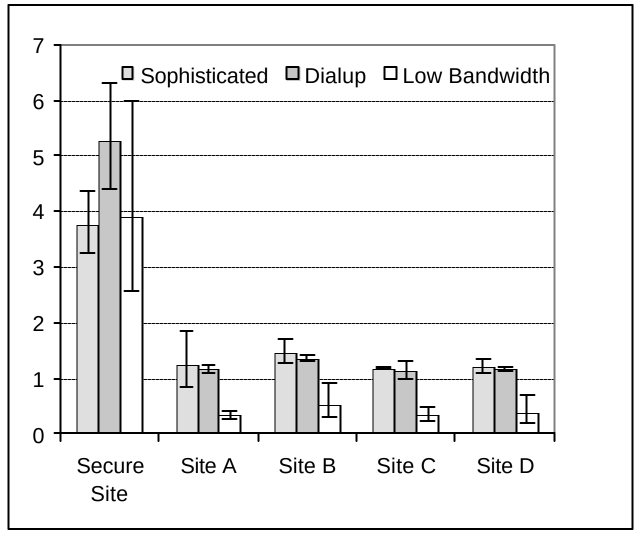


Figure 5: Comparison of Geometric Means and "Geometric Deviations" of "First Packet" Download Times (seconds)

The discrepancy for the Secure Site is easily explained. That page is secure, and the time for establishing the secure link (the SSL Handshake) is included in First Packet time for the transaction and dial-up Agents. That complex set of uploads and downloads means that "First Packet" is no longer a simple measure of the time to deliver the first packet, as is true for the other web pages in this study.

SUMMARY

Neither the sophisticated nor the low bandwidth emulations should be used for web pages. They're quite poor for comparing different pages, and they're also misleading for watching the daily performance cycles of a single web page.

Further research may also reveal that the daily pattern of congestion and resulting performance changes on a V90 dial-up link to specialized access networks, such as AOL, are different than those on a high-bandwidth connection to the Internet.

REFERENCES

[GOOD99] Bill Goodman, "Internet Telephony and Modem Delay," *IEEE Network*, vol. 13 no. 4, May/June 1999]

[RFC 2581] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control," RFC 2581, April 1999