OPScore, or Online Playability Score:
A Metric for Playability of Online Games with Network Impairments

Prepared by Ubicom Inc.

March 2005

We describe a technique for estimating the OPScore, or Online Playability Score, of Quake 3 in a broadband home networking environment. We show that Ubicom's StreamEngine technology improves the OPScore for online Quake 3 by more than 2X in the presence of heavy background traffic.

Introduction

Usage patterns of home networks are changing due to the increasing availability of broadband internet connections. Broadband connections make it possible to play online games, make Voice over IP (VoIP) calls, share files, and hold video conferences. These applications generate significant traffic in the uplink direction, from the home to the internet. However, the bandwidth of the uplink is much less than that of the downlink in many markets. An additional challenge is that most of these new applications are time-sensitive in nature. A user's satisfaction can be severely degraded if a voice call or online game is disrupted by traffic from other family members, or by congestion on the narrow bandwidth uplink.

This degradation in experience motivates us to look for a way to estimate end-user satisfaction

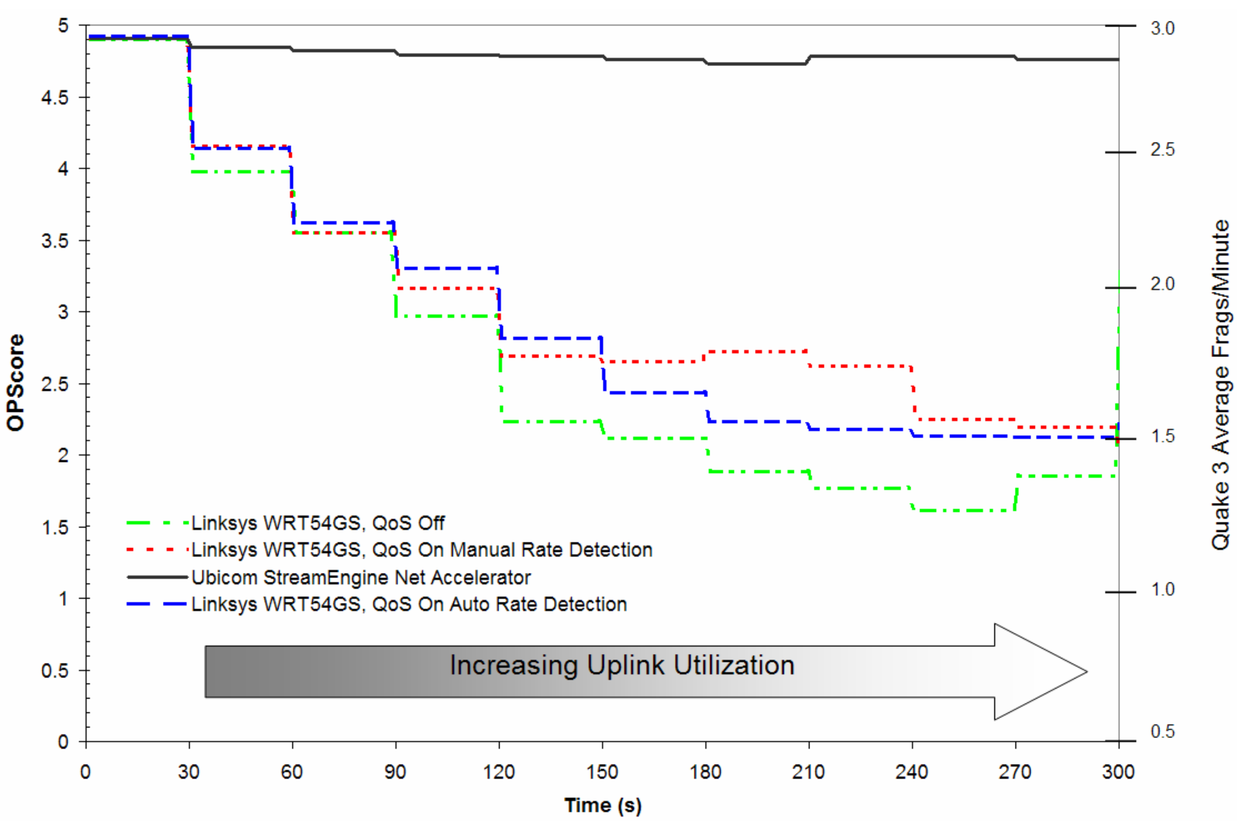

Figure 1: Estimated frags per minute and OPScore for multi-player Quake 3 with increasing background traffic on 1.5 Mbps/384 kbps ADSL modem

in this multiple-application, multiple-user environment. It is easy to measure throughput to gauge the performance of file sharing. There are also tools to estimate the Mean Opinion Score, or MOS, for VoIP calls. However, there is no easy, repeatable method for estimating the satisfaction of online game players.

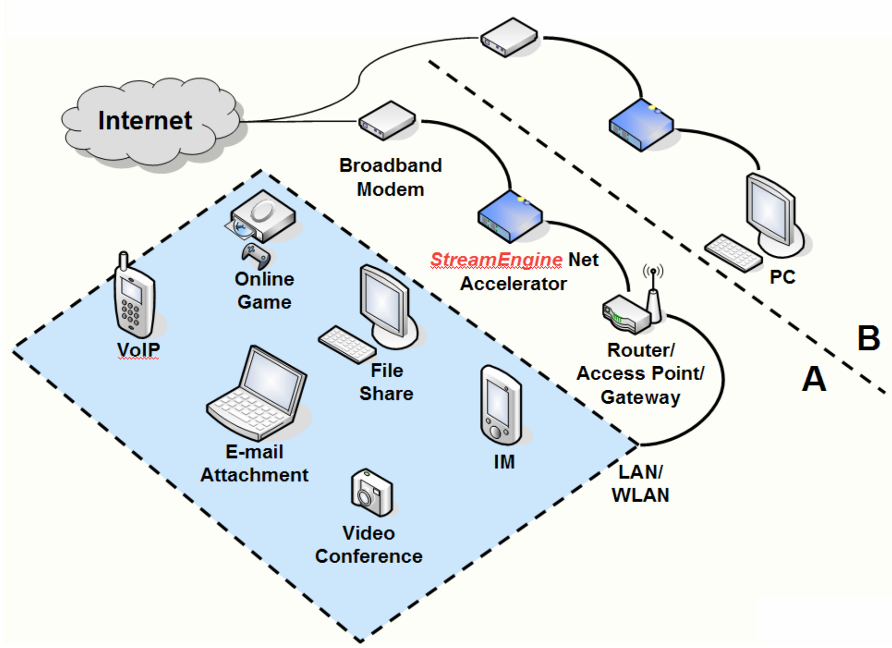

Figure 2: The StreamEngine Net Accelerator sits between the broadband modem and router. It optimizes networks for interactive applications used by A) multiple users, and B) individual users.

This paper introduces the concept of the Online Playability Score, or OPScore, measurement. The OPScore shows the relationship between network impairments, such as latency, jitter, and packet loss on user satisfaction in online games. The methodology also specifies a test methodology using a network traffic model. This model generates traffic that is representative of the multiple applications used simultaneously in homes with broadband networks. The test creates data streams for two online games, two voice calls, and background FTP transfers. The number of FTP transfers increase every 30 seconds during the 5 minute test duration to show the effects of increasing uplink utilization.

Figure 1 shows estimates of OPScore in four different network configurations. All tests use a WRT54GS broadband router. The OPScore was measured both with the router's built-in QoS feature turned off, and with QoS turned on, first with manual, and then automatic uplink data rate detection. We repeated the same test with the router's QoS feature turned off, but with a Ubicom StreamEngine Net Accelerator placed between the router and broadband modem. We see that StreamEngine can more than double the playability of online games in the presence of heavy background traffic.

Broadband Home Networking Environment

The deployment of broadband connections in residential environments is increasing the number and variety of applications for networks in the home. In households that use dial-up modems for internet connectivity, network applications are mostly restricted to web browsing, e-mail, and instant messaging. However, broadband internet connections through xDSL and cable modems have made applications such as online gaming, Voice over IP (VoIP), video conferencing, application sharing, and audio/video streaming popular.

Figure 2 illustrates the multiuser, multiapplication usage environment that forms the model for the "connected home". Web browsing and file downloads generate downstream (server to client) traffic primarily. For this reason, many xDSL and cable modem services were provisioned to provide greater bandwidth in the downstream than the upstream directions. It is typical in North America to find 1.5 Mbps download speeds paired with 128 kbps uplinks for ADSL residential service.

Figure 3: Comparison between different views of online first-person shooting game maintained by client (left) and game server (right) caused by network latency [Halo for PC shown]

Emerging applications place a number of new requirements on the network: First, file and application sharing, VoIP, and video conferencing require at least as much upstream (client to server or peer to peer) bandwidth as downstream bandwidth. Second, since these applications are time-sensitive, they must minimize latency, jitter, and packet loss in order to deliver a satisfying user experience.

It is extremely challenging to meet these requirements. Each application may have a very different tolerance to latency, jitter, and packet loss, and may require different allocations of upstream bandwidth. Multiple users may access the network at the same time. This means that the same network must meet requirements from multiple applications simultaneously.

Ultimately, the satisfaction a user feels in using a particular application is subjective and qualitative. Several benchmarks have been developed that attempt to relate measurable quantities to performance or user satisfaction in a particular environment.

For example, the telephony industry has produced the MOS, or Mean Opinion Score metric. MOS describes average user satisfaction for voice calls on a one through five scale, with a score of five being most satisfied, and a score of one being least satisfied. The International Telecommunications Union issued Recommendation G.107, the so called "E-Model".[12]

The E-Model describes a method of using quantitative measurements of latency, jitter, and packet loss, among other things, to forecast MOS in a particular implementation. A metric such as MOS is useful because it describes user satisfaction with a single number. It encapsulates the complex dependence of users' experience on multiple parameters.

Rather than trying to estimate user satisfaction, some benchmarks provide a standardized, quantitative means of measuring system performance in a multiapplication environment. As examples, PCMark04 by FutureMark, and SYSmark 2004 by BAPCo measure the performance of personal computers running multiple office applications simultaneously. These benchmarks are important for this discussion since they are examples of a way to measure performance in a multiapplication environment. They measure not only the performance of a system, but describe the context in which it is used as well.

The following is a list of some popular applications and their associated benchmarks used to describe performance or user satisfaction:

- PC Games: Frames per Second

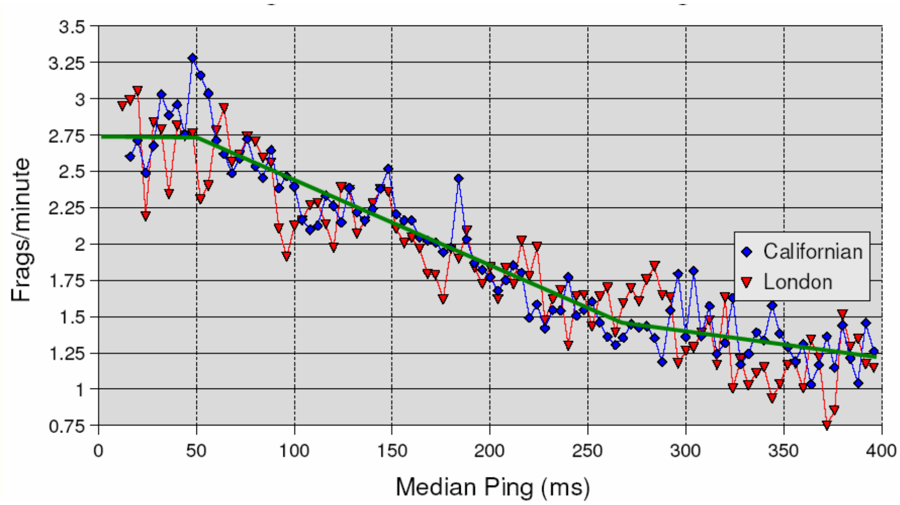

Figure 4: Average frags (frags) per minute as a function of median ping times for 2 public Quake 3 servers[10]

•Voice over IP: Mean Opinion Score

•Office Applications: PCMark04, SYSmark 2004

•File Sharing: Throughput

There is one application for the connected home for which a benchmark is not available: online games. Any effective benchmark for online games should meet the following requirements:

•Show a clear relationship between quantitative metrics and qualitative user satisfaction.

•Accurately represent effects of multiple users and multiple network applications on online games in a real world environment.

•Be reproducible.

OPScore: A New Benchmark for Online Gaming

This paper introduces a new benchmark, OPScore, or the Online Playability Score, to describe the effects of network impairments on the playability of online games. The benchmark encompasses a measurement methodology, a concise model of both game and background traffic, and algorithms for projecting the performance of players of online games given measurements of real world network impairments. These elements combine to allow testing of QoS features of routers and other devices in a realistic environment.

Effects of Network Impairments on Online Game Play

Latency

Latency, also known as delay, is one of the impairments that most affects online game play. It is the average amount of time necessary to transmit information about a player's actions from the Application Layer of the client to the Application Layer of the server. Since it takes a finite amount of time to transmit data from the client to the server, the view of the game as maintained by the server can never be perfectly synchronized with that of the client. This causes the effect commonly referred to as "game lag".

We can see the effects of latency, for example, in an online first-person shooting game. Figure 3 shows a view of the game from the frame of reference of one of the game clients. In this view, the player is presented with the most up-to-date coordinates of the opponent from the server. The player pulls the trigger, and launches a bullet towards the target. This action is encoded in a packet that is transmitted from client to server. The latency of transmission creates a lag between the frames of reference of the client and server.

The player sees the bullet hit the target within the client's frame of reference. However, in the server's frame, illustrated in Figure 3, the target has already moved out of the path of the bullet. The player's client displays a hit, while the server shows that the target was missed. The server's view of the game is authoritative, and so the player has not scored in this example. It is easy to see how the lag created by packet latency can affect online game play, and players' enjoyment of the game.

A recent paper[1] shows the influence of latency on online game play for Quake 3. The study measured frags (game kills) per minute and ping times for thousands of Quake 3 clients using two publicly available servers across 164 days of continuous game time. If we assume that ping times are uncorrelated with players' skill levels, then player skill averages out across all measurements.

Ping times are related to but not equivalent to latency. The ping command measures round trip latency. To study impairments caused by the network on online games, we must look at the one-way delay. In the case where the capabilities of the network are the same in the uplink and downlink directions, we can assume that one-way delay is roughly half of the stated ping times.

Figure 4 shows a significant effect of latency on game play in online shooting games. Players experienced, on average, roughly a 50% reduction in frags per minute for 300 ms ping times compared to ping times of 50 ms or less. Non-wireless internet connections can typically experience up to 400 ms of latency. Under adverse conditions, especially with background file transfers, latency can quickly jump to over 1000 ms.

Latency is measured from the time data is generated by the client application to when it is received by the server application. For this reason, any part of the system may contribute to overall latency. In the absence of other traffic on the uplink, the latency is dominated by transmission across the internet itself. However, when other network-based applications are transferring data on the same uplink, the largest contribution to latency is from the xDSL or cable modem and router. When packets are generated by the WAN interface of the client's router, they are buffered by the modem before being transferred at the relatively low data rate of the uplink. This buffering process is responsible for most latency when the uplink is being used by multiple applications.

Jitter

At a basic level, jitter is defined as the variance of the latency. Jitter may come from several sources. Packets in a stream do not necessarily traverse the same nodes as they travel across the internet. Even the number of hops taken from client to server may vary greatly. A larger numbers of hops is associated with longer latency, while a smaller number of hops tend to create shorter latency. Packets are queued in routers and switches along the path to the server. Depending on traffic patterns the queuing delay for each packet may vary and result in jitter.

Just as congestion on the uplink may be a significant source of latency, this congestion may also cause jitter. Again, it is typical for the bandwidth of the WAN interface of the client router to greatly exceed that of the uplink of the modem. Packets queue in the modem buffer before reaching the internet. Most online game clients generate low data rate streams. Any background traffic will take up space within the modem buffer between each of the game-generated packets. Many types of applications such as FTP transfers, video streams, and application sharing are of a bursty nature. These bursts of traffic will place random amounts of data between the game packets waiting in the modem buffer. Experimentally we see that even small amounts of background traffic can drastically increase the jitter in the game data stream.

With very little background traffic on the uplink, jitter will rarely make up more than 20% of overall latency. However, when other applications share the uplink with the online game traffic, jitter can easily exceed 50% of the average latency.

What are the effects of packet jitter on online game play? Some applications, such as video streaming, can mitigate the effects of jitter using a so called "jitter buffer". A jitter buffer places incoming packets in a queue that trades jitter for a fixed delay in propagating through the buffer. Unfortunately, the highly time-sensitive nature of most games makes the use of jitter buffers impractical.

We might speculate that players can compensate for small amounts of latency by "shooting ahead" of the target. However, jitter is effectively impossible to compensate for by the player due to its inherent randomness. Another study[4] shows that online shooting games are even more sensitive to jitter than to latency alone. It is difficult to interpret this data, however. Since jitter is the variance of latency, and both must be positively valued, it is impossible to have jitter in the absence of latency. For this reason, it is challenging to completely decouple the effects of latency and jitter on game play, since they almost always occur together.

The coupling between latency and jitter makes interpretation difficult, but the study illustrates the drastic effects that jitter can have on game play. Frags per minute reduce from 1.8 to 0.8 in the presence of only 70 ms of jitter.

Packet Loss

Packet loss has received little attention as an impairment to online games. With clients that have little or no background traffic, packet loss is a rare occurrence. In this case, lost packets make up no more than 0.5% of overall traffic.

The trouble arises when background traffic is increased. When the buffers on either the modem or client router become filled, any additional packets destined for the server will be discarded.

Again we must note the differences between what is seen by the client and by the game server. Online shooting games have mechanisms for correcting for small amounts of packet loss. As a player moves within the game map, the server is constantly forecasting the most likely trajectory for the player's future movement. If a packet is dropped on its way to the server, the server will compensate by replacing the updated player coordinates from the client with the projected coordinates. During intense game play, the coordinate forecast and the coordinates contained in the dropped packets will differ. When the server successfully receives the next packet from the client, the player will appear to snap back into the correct position creating jerky and erratic movement.

If the packets are experiencing a large amount of jitter, the player will see an effect similar to that of packet loss. Packets that are reordered in the stream or come at a time other than the regular interval may be ignored by the server, even though they were received correctly. In this way, jitter can induce an effect very similar to that of packet loss during game play.

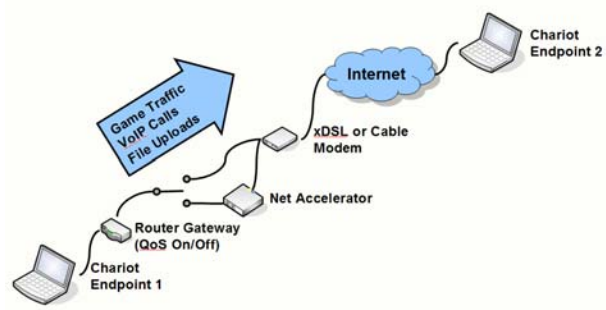

Figure 5: Hardware configuration for OPScore measurement

One of the greatest impacts on game play comes from lost packets. Consider the case where the player sights the target, and pulls the trigger. The action of firing the weapon is encoded in a packet, and sent to the server. If that packet is lost on its way to the server, the player will see the shot, but the server is never notified. In the server's view, the player never fired, and so no kill is scored.

It is clear to see how packet loss has a direct impact on game play. It could be argued that, if this happens during a game, the player could recover and immediately shoot again. However, it may be nearly a second before the player realizes that the shot did not actually reach the target. This is a long time in a first person shooting game. Most likely, the opponent has already moved, and the chance to score has been lost due to the loss of packets.

Uplink Utilization

Most online games consume very little upstream bandwidth. Even a relatively narrow uplink has more than enough capacity to handle a game stream. For this reason, we do not consider low upstream bandwidth an impairment in itself. However, its influence on the impairments of latency, jitter, and packet loss is so great that it deserves its own discussion.

Quake 3 G-Model

$R = (W_L \times L + W_J \times J)\ (1+E)$

$W_L \equiv 1$

$W_J = (\Delta K / \Delta J) / (\Delta K / \Delta L)$

OPScore = Lookup(R)

L: Latency, or one-way delay, in ms

J: Jitter as defined in RFC 1889, in ms

K: Average frags per minute

W_L: Latency weighting factor, equal to 1 by definition

W_J: Jitter weighting factor

E: Packet loss, as a percentage of bytes lost

R: Impairment factor, in ms

Lookup( ): Piece-wise linear lookup table to change R into OPScore

The narrowness of the uplink amplifies the effect of impairments on the network. A small upstream bandwidth will cause the modem buffer to fill more quickly, increasing latency. If the modem is uploading data very slowly, then there will be more time for packets to gather between each of the game packets within the buffer. This increases jitter. These same conditions will increase the chance that packets are dropped by a node because the buffer is filled.

For these reasons, the important parameter to consider is not the uplink bandwidth itself, but the uplink utilization. Uplink utilization is the average upload speed as a percentage of the maximum transfer speed. A network with a narrow uplink will experience game lag earlier than one with a wide bandwidth. Of course, these conditions will also make loss of packets a more common occurrence.

Measurement Methodology

Most studies of the effects of network impairments use ping times as a measure of latency. There are several problems with this approach. Ping times show round-trip delay, when the critical parameter for online game playability is one-way delay from the client to the server. We cannot even assume that one-way delay is roughly half of the ping time. The asymmetric data rates and bandwidth utilizations can create very different conditions on the uplink versus the downlink. Interpreting ping times is even less meaningful when QoS techniques are used. The QoS mechanisms may handle packets generated by the ping command differently than those from the game client.

Measurement of one-way delay requires a special approach such as that used by Ixia Chariot or the NetMeter package from the Advanced Broadband Communications Center at the Polytechnic University of Catalonia. These software packages use special applications running on both the client and server machines. For Chariot, these applications are called Performance Endpoints. During a negotiation phase, the two applications exchange time-stamped packets that are used to synchronize high-accuracy clocks within each computer. Once the clocks are synchronized, time-stamped packets can be sent from the one computer to the other. The time stamp is then compared to the output of the high-accuracy clock to determine the one-way delay. This capability motivates us in not using the original applications to generate traffic, but a tool such as Chariot. This tool gives us a way to measure latency and other parameters.

Figure 5 shows the hardware configuration of a test to characterize the effect of different QoS implementations on online game play. The computer labeled Chariot Endpoint 1 is acting as the client in this scenario, generating traffic that is representative of the various applications used in the networked home. Chariot Endpoint 1 is connected to the LAN or WLAN port of a router.

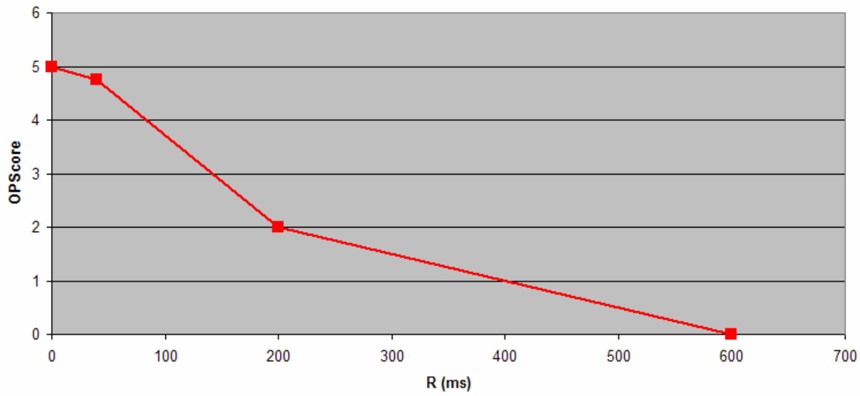

Figure 6: Lookup table used to convert impairment factor R to OPScore for Quake 3

The router may or may not have QoS features turned on, depending on the test. These tests were run both with and without a Ubicom Net Accelerator placed between the router and broadband modem. The WAN port of the router or Net Accelerator is connected to the LAN port of the broadband modem. For these tests, we used an ADSL modem with 1.5 Mbps downlink, and 384 kbps uplink specified by the service provider. The broadband modem connects to the public internet through the service provider. A second PC, Chariot Endpoint 2, is connected to the public internet via a high speed connection. Chariot Endpoint 2 acts as the server in this case, receiving packets from Endpoint 1, and relaying test results back to Chariot Endpoint 1 for analysis. Results are gathered by a separate application running on Endpoint 1 called the Chariot Console.

The uplink traffic is generated by a network traffic model implemented as a Chariot test. The test generates traffic representing two online games, two VoIP calls, and a variable amount of FTP upload as background traffic. The level of background traffic increases every 30 seconds throughout the five minute test duration. This model is explained in greater detail below. At the conclusion of the test, time records containing throughput, latency, jitter, and packet loss for each stream are exported as a comma delimited text file for analysis. The time records undergo post-processing to convert the time records for the online game traffic into OPScore measurements. The post-processing algorithm is called the G-Model, and was written as an automated Microsoft Excel Workbook.

Equipment and Software

•(2) Asus M5N Notebook Computers: 1.5 GHz Intel Pentium M Processor, 2 MB cache, 768 MB RAM, Windows XP SP 2

•Linksys WRT54GS Wireless G Broadband Router with SpeedBooster

•Ubicom StreamEngine Net Accelerator Version 0.3.3.

•Efficient Networks Speedstream 5100 ADSL Modem

•ADSL Connection: 1.5 Mbps Downstream, 384 kbps Upstream

•Ixia Chariot Console and (2) Performance Endpoints version 6.0

•Microsoft Office Excel 2003

G-Model for Estimating Playability of Online Games

The purpose of the G-Model is to transform measurements of latency, jitter, and packet loss into estimates of end-user satisfaction for specific online games. The initial implementation of the G-Model targets Quake 3, and uses frags per minute as a proxy for user satisfaction. It is expected that the structure of the G-Model may change from one type of online game to the next. For example, the G-Model for first person shooting games may be different from that of an online sports game. The G-Model for games in the same class are expected to differ only in the model coefficients. For instance, Quake 3 and Unreal Tournament have the same G-Model structures, but different constants to transform

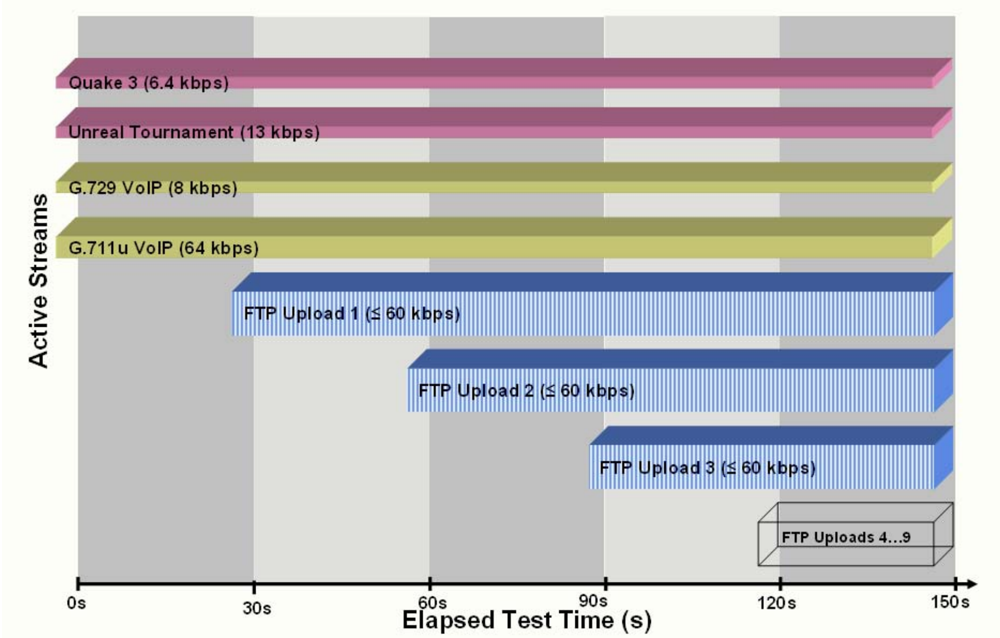

Figure 7: Network traffic model used for OPScore characterization. 5 minute total test duration.

A lookup table, shown in Figure 6, is used to show how the OPScore in Quake 3 changes with respect to an overall impairment factor R. Ideally, OPScore should map fairly linearly to frags per minute for a particular first person shooting game. The lookup table was constructed based on investigations by university researchers into the dependence of latency, jitter, and packet loss into frags per minute.

The Quake 3 G-Model has the following form: Both the latency and jitter values are scaled by weighting factors, W_L and W_J, respectively. These weighting factors are used to describe the relative level of impairment that latency or jitter causes in a particular online game. The latency weighting factor was set to 1. The jitter weighting factor shows how changes in jitter influence the average frags per minute in Quake 3 relative to changes in latency. If jitter has a large effect on game play, it will have a large weighting factor. If jitter has relatively little affect on a particular game, the jitter weighting factor would be small.

Once the latency and jitter are weighted, their sum is scaled by the packet loss. The result is the impairment factor R. Since packet loss immediately translates into reduced frags per minute in first person shooting games, we chose to directly scale R by this amount. The impairment factor describes the overall level of impairment on the online game, given some combination of latency, jitter, and packet loss.

frags per minute in first person shooting games on network impairments such as latency and jitter.[1, 3, 4, 5, 10] By correctly choosing coefficients for the G-Model, typical values of latency, jitter, and packet loss should yield an OPScore estimate that directly relates to frags per minute. Frags per minute is not an appropriate figure of merit for all online games. For this reason, we chose to use OPScore as a metric, rather than show frags per minute directly.

Network Traffic Model

The G-Model is a tool for estimating the playability of an online game given a certain combination of latency, jitter, and packet loss. We have already seen that the context in which an online game is used greatly affects these impairments. The context of an online game includes not only the physical configuration of the network, but the various other applications that are used while the online game is being played. We can generate packets that have the same characteristics as the actual applications using a network traffic model. The network traffic model allows us to generate traffic in a repeatable and measurable way.

The network traffic model used in this study attempts to replicate the usage environment of the

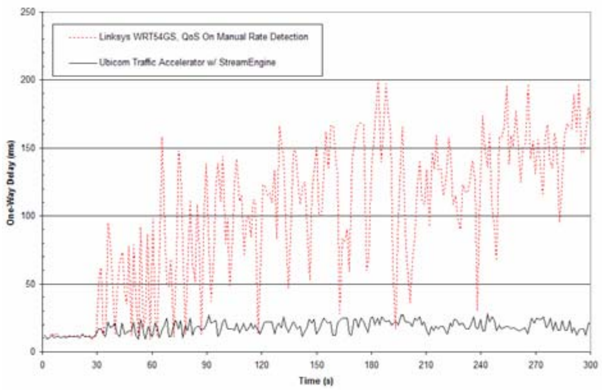

Figure 8: Latency (one-way delay) of synthetic Quake 3 packets on 384 kbps uplink with increasing background traffic.

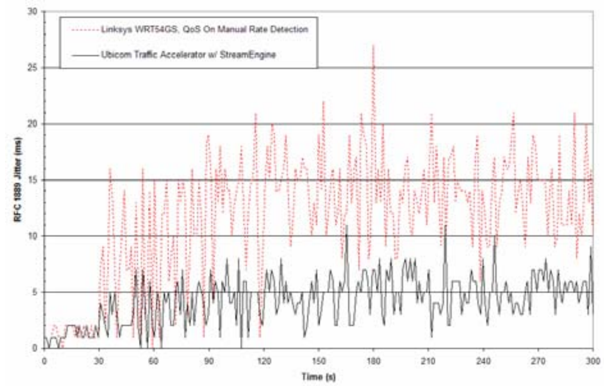

Figure 10: RFC 1889 jitter of synthetic Quake 3 packets on 384 kbps uplink with increasing background traffic

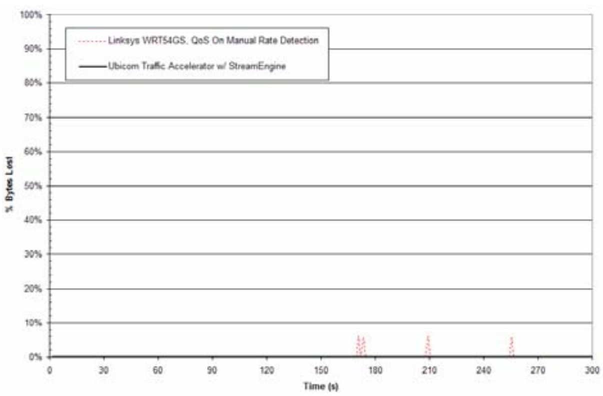

Figure 9: Percent bytes lost for high-priority synthetic Quake 3 packets on 384 kbps uplink with background traffic

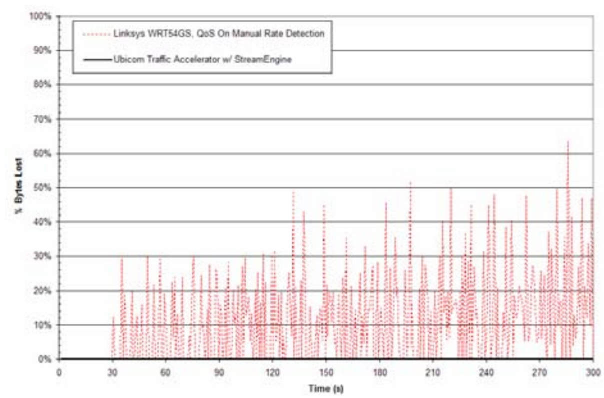

Figure 11: Percent bytes lost for low-priority synthetic Unreal Tournament packets on 384 kbps uplink with background traffic

connected home. The connected home is a multiapplication, multiuser environment. We must faithfully represent the important characteristics of any applications that might be used in this environment so that we can see their affects on the online game.

As illustrated in Figure 7, the network traffic model contains three different types of traffic. These correspond to online games, VoIP, and file transfers. A stream in this discussion refers to traffic representing packets from a single application. The network traffic model contains thirteen discrete streams: two online games, two VoIP calls, and nine FTP file transfers, with all traffic transferred in the uplink direction. The game and VoIP streams are active throughout the test. The number of active FTP file transfers changes during the test to show the effects of increasing background traffic. No FTP streams are active during the first 30 seconds of the test. One additional FTP stream becomes active at 30 second intervals from initialization of the test. The average uplink utilization becomes greater for each 30 second test segment as the number of simultaneous FTP transfers increases. The total test duration is 5 minutes. For a typical DSL uplink speed of 384 kbps, the uplink utilization goes from very low at the start of the test to being completely saturated when the test concludes. The actual uplink utilization will vary due to specific line conditions and the type of QoS facilities active during the test.

The FTP streams are designed to model bursts of traffic from multiple file uploads. Each discrete file upload transfers 25k bytes of data at the maximum possible rate. There is a random time interval between each transfer. In the absence of other traffic, the average throughput of a single FTP stream is approximately 60 kbps on a 384 kbps uplink. Actual throughput of each stream will vary based on line conditions, and the number of other streams transferring data at the same time.

There are two VoIP streams as part of the network traffic model. These are a single G.729 stream at a data rate of 8 kbps and a G.711u stream at 64 kbps. These two streams are active throughout the test as defined in this model.

The packets corresponding to online games are generated by Chariot as a Game Source Model. Each Game Source Model creates traffic that has the properties necessary to estimate the effects of the network and other traffic on an online game. In this case, we control the port number, packet size, data rate, headers, and packet payloads to closely resemble those of actual game client applications. The test includes Game Source Models for Quake 3 and Unreal Tournament. Both of these streams run continuously throughout the test.

Taken together, these streams represent the typical environment of the connected home. The test can be used to see the case of both low uplink utilization at the start of the test, and high uplink utilization near the test's conclusion. Since several time-dependent applications are included as part of the network traffic model, any one of these streams may be analyzed to determine their level of impairment.

Effects of QoS on Quake 3 Online

Figure 1 shows the average OPScore for various hardware configurations and QoS settings. The amount of background traffic increases as a step function every 30 seconds, and so we show the average of the instantaneous OPScore values in each of these segments.

We repeated the test for 4 different network configurations:

Test Case 1: Using Linksys WRT54GS router only, with no QoS features enabled on the router.

Test Case 2: Using Linksys WRT54GS router only, with the built-in QoS feature turned on. The TCP port associated with the Quake 3 Game Source Model was assigned a High priority. All other ports were assigned a low priority. The Uplink Rate Detection parameter on the router was set to "Auto".

Test Case 3: Same as Test Case 2, however, the Uplink Rate Detection was set to "Manual" with an Uplink Data Rate of "300 kbps". This value was found to be optimal after extensive experimentation.

Test Case 4: Using Linksys WRT54GS, with a Ubicom StreamEngine Net Accelerator[11] placed between the router and ADSL modem. All QoS features on the router were turned off. The Net Accelerator does not require configuration for this test.

At the beginning of the test, there are no active FTP transfers occurring in the background. During this segment, all tests yielded an average OPScore value of approximately 4.75. This corresponds to game play that is almost completely without impairment. The only penalty is given for the inherent latency and jitter of the internet itself.

The first background FTP transfer begins in the second 30 second segment of the test. During this segment, we notice an immediate decrease in OPScore for Quake 3 in Test Cases 1 through 3. There is a slight improvement in these tests when QoS is on. However, when a Ubicom Net Accelerator is placed between the router and modem, there is virtually no degradation in game play.

The uplink utilization goes up as the test progresses. The uplink is nearly saturated roughly halfway into the test. However, because of the bursted nature of the FTP transfers, the uplink is not completely saturated until the last minute of the test.

As the test progresses, we notice a steady decline in the OPScore values, even with QoS turned on in the router. There was virtually no degradation in the quality of game play when using the Net Accelerator, even at 100% uplink utilization.

By the conclusion of the test, Test Case 1 through 3 have all suffered roughly a 60% decline in OPScore. This corresponds to a reduction in frags per minute in Quake 3 from 2.75 to about 1.5. This is a very significant reduction in the playability of this online game. The frags per minute with a Net Accelerator in place remained effectively unchanged, regardless of the uplink utilization.

We can see that for Test Case 2, with rate detection in its default setting, there was a marginal improvement in OPScore from Test Case 1 with QoS disabled. It was discovered that performance was improved when the uplink data rate was "backed off" from the maximum uplink data rate. In this case, the maximum uplink rate was about 330 kbps. The OPScore was maximized in Test Case 3 with the uplink data rate manually set to 300 kbps on the router. This optimization probably would not have been possible without the use of advanced tools such as Chariot, and extensive experimentation.

Interpreting Latency, Jitter, and Packet Loss Measurements for Online Games

How do the specific impairments of latency, jitter, and packet loss contribute to the OPScore during these tests? The most degradation to OPScore comes from latency and jitter. We can see that in both test cases shown in Figure 8, the latency is well below 50 ms for the first test segment. However, when the first FTP transfer begins, the latency in Test Case 3 begins to fluctuate widely. This may be due to the bursts in traffic caused by each individual file transfer. As the test progresses, the latency may reach up to 200 ms.

This latency fluctuation may have an adverse effect on game play that is not captured by the OPScore measurement. The OPScore is estimated for each time record generated by Chariot for the Quake 3 stream. Latency and jitter are measured only for the period associated with each specific time record. Chariot returns time records for the Quake 3 stream about once every 3 seconds. It will not detect variation in latency that occurs on time scales longer than 3 seconds. The fluctuation in latency at longer time scales does not show up in the jitter measurements for each time record, even though it may impair actual game play. OPScore measurements shown in Figure 1 were averaged across each 30 second test segment for clarity. This hides the impact of instability in the latency. If the jitter describes the variance of the latency, then no adjustment is necessary. However, if the jitter measurement does not capture the variance in latency, the measurement algorithm may need to be revised in the future to account for this.

Figure 10 shows the RFC 1889 jitter for Test Cases 3 and 4. Again, the jitter is quite low for the first test segment, but then increases dramatically when a Net Accelerator is not used. It is important to pay close attention to jitter levels in the online game traffic. Research has shown that jitter has an even greater impact on the playability of online games than latency alone.[4] We can see that the StreamEngine Net Accelerator greatly reduces the jitter during the test with respect to using the router's built-in QoS features.

An interesting effect can be seen when comparing the packet loss for various application streams in the network traffic model. For Test Case 3, using only the WRT54GS router, the packet loss remains low for the Quake 3 stream set to high priority. This is shown in Figure 9. However, in Figure 10 we see the packet loss for the other game stream, unprioritized Unreal Tournament, is extremely high. We saw similar degradation in the MOS of VoIP calls as well. The packet loss is so high in this case that the game would be essentially unplayable. It was necessary to set Quake 3 as the only high priority application for Test Cases 2 through 3. If any other application received high priority, the test results showed no improvement in OPScore from having QoS off. In contrast, the packet loss was essentially zero for all streams when using the StreamEngine Net Accelerator.

These tests show that we can get a slight benefit from using the built-in QoS functions of the

Linksys WRT54GS. However, this comes at the expense of the other time-critical applications in use at the same time. We can see that the StreamEngine Net Accelerator maintains excellent playability of Quake 3, as well as other applications, even with very high background traffic. In addition, no configuration was necessary of the Net Accelerator, compared to the extensive tuning needed to show improvement in OPScore with the router alone.

Conclusion

This paper proposes a technique for estimating the effects of network impairments on online game play, focusing specifically on Quake 3 as a target application. This technique uses measurements of traffic in a realistic home network environment to forecast the playability of online games. The OPScore, or Online Playability Score, is a means of representing the level of impairment on an online game caused by these network impairments. We show that the OPScore for Quake 3 can be severely impacted by traffic from other applications being sent on the same uplink connection. The router QoS features that were tested were shown to be fairly ineffective at improving OPScore under these conditions. Overall, under heavy background traffic, the StreamEngine Net Accelerator brought about over a 100% improvement in OPScore compared to the cases using only a router with standard QoS features.

References

[1]S. Zander, G. Armitage, "Empirically Measuring the QoS Sensitivity of Interactive Online Game Players," Proc. Australian Telecommunications Networks & Applications Conference 2004 (ATNAC 2004), Sydney, Australia, December 2004

[2]T. Lang, G.J. Armitage, P. Branch, H. Choo, "A Synthetic Traffic Model for Half-Life," Australian Telecommunications Networks & Applications Conference 2003, Melbourne (ATNAC 2003), Australia, December 2003

[3]G. Armitage, "Sensitivity of Quake3 Players To Network Latency," Poster session, SIGCOMM Internet Measurement Workshop, San Francisco, November 2001

[4]G. Armitage, L. Stewart, "Limitations of using Real-World Public Servers to Estimate Jitter Tolerance of First Person Shooter Games," ACM SIGCHI ACE2004 Conference, Singapore, June 2004.

[5]T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, "The Effects of Packet Loss and Latency on Player Performance in Unreal Tournament 2003," Major Qualifying Project Report, Worcester Polytechnic Institute, March 3, 2004.

[6]Quake 3, http://www.idsoftware.com/games/quake/quake3-arena/ (as of March 2005)

[7]I. S. MacKenzie, S. Ware, "Lag as a Determinant of Human Performance in Interactive Systems," INTERCHI, pp. 488-493, 1992.

[8]M. Borella, "Source Models of Network Game Traffic," Computer Communications, vol. 23, no. 4, pp. 403-410, February 2000.

[9]T. Lang, P. Branch, G. Armitage, "A Synthetic Traffic Model for Quake 3", ACM SIGCHI ACE2004 conference, Singapore, June 2004.

[10]G. Armitage, "An Experimental Estimation of Latency Senstivity in Multiplayer Quake 3," ICON 2003, Sidney, Australia, September 2003.

[11]"Solving Performance Problems with Interactive Applications in a Broadband Environment using StreamEngine™ Technology", Ubicom Inc., October 2004.

[12]"International Telecommunication, Recommendation G.107: The E-model, a computational model for use in transmission planning," December 1998.

[13]S. Ubik, V. Smotlacha, "Precise Measurement of One-Way Delay and Analysis of Synchronization Issues," Cesnet Technical Report, July 2001.

Ubicom, Inc. **IP3K-DWP-OPSCORE-10**

Technical Inquiries:

Contact: Kevin Gee
Tel 650 210 1503
Fax 650 210 8715
Email kgee@ubicom.com

Sales and General Inquiries:

635 Clyde Avenue
Mountain View, CA 94043
Tel 650 210 1500
Fax 650 210 8715
Email sales@ubicom.com
Web www.ubicom.com

Ubicom™, Inc. develops communications processor and software platforms that address the unique demands of real-time, interactive applications in home and small office environments. The company provides optimized system-level solutions to OEMs for a wide range of products including wireless routers, access points, VoIP gateways, streaming media devices, print servers and other network devices. Ubicom's unique multithreaded processor design, real-time operating system, and application-level solutions combine to ensure a high-quality user experience with unprecedented ease of use. Ubicom is a venture backed, privately held company with corporate headquarters in Mountain View, California.