FPS Game Performance in Wi-Fi Networks

Yanni Ellen Liu[1,3], Jing Wang[1,3], Michael Kwok[2], Jeff Diamond[3], Michel Toulouse[1,3]
[1]Department of Computer Science, University of Manitoba, Winnipeg, Canada
Email: {yliu,jingwang,toulouse}@cs.umanitoba.ca
[2]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
Email: kfkwok@uwaterloo.ca
[3]TRLabs, Winnipeg, Canada
Email: diamond@win.trlabs.ca

ABSTRACT

Multi-player online games (MOGs) have become increasingly popular on today's Internet. Meanwhile, the IEEE 802.11 (Wi-Fi) wireless networks have been widely used. We study how well an underlying 802.11g network supports a first-person-shooter (FPS) game, often considered the most demanding MOG in terms of network performance. We measure the latency and loss ratio performance experienced by the game traffic; these network-layer metrics were shown to have large impact on the gaming quality experienced at the application layer. The effect on performance of the following factors were examined: the number of game clients, the distance between game clients and the wireless access point, the enabling of data encryption, and the inclusion of FTP and video streaming background traffic. Our experimental results show that FTP and video streaming traffic significantly affect the game traffic performance, whereas the distance and the use of data encryption have rather minimal impact. We also observe that when the amount of background traffic is moderate to high, the performance degrades as the number of game clients increases. Based on our observations, we suggest QoS strategies that may be used to better support games in a Wi-Fi environment.

Keywords

Multi-player games, IEEE 802.11g, wireless, Wi-Fi, QoS-support, experimentation, performance, measurement

1. INTRODUCTION

Multi-player on-line games (MOGs) are computer games in which multiple game players simultaneously participate in the same game session over a computer network. Such games are increasingly popular on today's Internet due to the availability of high-speed networks and affordable high performance personal computers. One popular system architecture for MOGs is client-server where each game client is connected to a game server via a computer network. State update messages are transmitted between the game clients and the game server. Different types of games may have different quality-of-service (QoS) requirements on the underlying network. Such requirements may be in terms of the delay and loss ratio performance in delivering the game traffic. Among types of MOGs, first-person-shooter (FPS) games [1, 2, 3, 4, 5] often have the most stringent requirement on network performance because of the highly interactive nature of such games.

Most game clients today are connected to their respective game servers via wired networks. Over the past few years, the IEEE 802.11 (Wi-Fi) [11] wireless networks have gained wide deployment. Wi-Fi network access points are commonly seen in coffee shops, office buildings, university campus, airports, and many residential homes. The capacity of Wi-Fi networks has also kept increasing. A Wi-Fi network interface has become a standard built-in on many of today's laptop computers. In view of these advances, it is anticipated that participating in a MOG from a Wi-Fi environment may become more and more common.

To better support MOGs, the capability of underlying networks needs to be evaluated with respect to the QoS requirements of these games. Such an evaluation for wired networks has been carried out in the literature (see [7, 19] for example). In comparison, less attention has been paid to the investigation of MOGs in wireless networks. In a Wi-Fi wireless network, many factors may affect the game performance. These include the number of wireless game clients, non-game traffic sharing the same Wi-Fi network with the game traffic, the wireless protocol, and physical environment parameters such as the distance and clearance of sight between wireless clients and the access point, humidity, and interference with other wireless devices. In what follows, we will refer to the non-game traffic sharing the same Wi-Fi network with the game traffic as the *background traffic*.

In this paper, we study the performance of FPS games in an IEEE 802.11g wireless network. We use emulated game traffic as well as other types of background traffic, and measure the latency and loss ratio performance perceived by the game traffic at the network layer, which are referred to as *game traffic performance* in this paper. These two metrics were shown to have large impact on the game performance perceived at the application layer [7, 19]. Differ from many existing studies in which simulation modeling was used, we take an experimental approach and evaluate the performance by measurement. The effect on performance of the following factors were examined: the number of game clients, the distance between game clients and the wireless access point, the enabling of data encryption, and the inclusion of FTP and video streaming background traffic.

Our experimental results show that FTP and video streaming traffic significantly affect the game traffic performance, whereas the distance and the use of data encryption have rather minimal impact. The large impact of background video streaming traffic is as expected because, same as the game traffic, it is delivered using the UDP protocol, which does not possess congestion control; when video load is high, congestion occurs, performance deteriorates. Somewhat unexpected, though often considered elastic, the TCP based FTP traffic also degrades game performance greatly. Thus both TCP and UDP traffic may need to be regulated in order to adequately support games. We provide a detailed analysis explaining these observations. In addition to the above results, we also observe that when the amount of background traffic is moderate or high, the performance degrades as the number of game clients increases. We provide insight into why the number of game clients also affects the game traffic performance. Based on our observations, we suggest QoS strategies that may be used to better support games in a Wi-Fi environment.

In our investigation, we assume that a game server is located on the same local area network (LAN) as the Wi-Fi access point to which game clients are connected; the scenarios in which a Wi-Fi network acts as an access network to a wide area network (WAN) and the game server is remotely located from the game clients are not considered. It should be noted however, that our results would provide useful insight into the QoS support to games in a wide-area wireless/wired environment, when combined with results from the WAN performance studies. For example, assuming an end-to-end latency requirement for interactivity is known, given the average delay performance on the wireless segment of an end-to-end game traffic path, one may estimate what range of average latency is required in the wired segment of the same path.

The rest of the paper is organized as follows. In Section 2, we provide background information of this study and review the related literature. In Section 3, we describe the test bed that we set up to evaluate the performance of FPS games in an 802.11g network. In Section 4, we present and analyze our experimental results. In Section 5, we suggest some QoS strategies to support MOGs in a Wi-Fi environment. Finally, we conclude our study in Section 6.

2. BACKGROUND AND RELATED WORK

First-person-shooter (FPS) is "a genre of computer and video games that is characterized by an on-screen view that simulates the in-game character's point of view, and is centered around the act of aiming and shooting handheld weapons, usually with limited ammunition" [2]. In a FPS game, when a player makes a move, e.g., firing a bullet at an enemy player, the game client on the player's machine includes this move in the next state update message sent to the game server. After processing this message, the server distributes the update to those players who are affected by the move in its next state update messages to clients. The shorter the latency in delivering the state update from the client to the server and then to the affected clients, the more realistic the game play. Many works have been carried out to study the traffic characteristics and the performance of FPS games. In general, traffic generated by FPS games has been identified to have small packet sizes, regular inter-arrival intervals, and relatively low bit rate [13, 18].

For the performance of FPS games, researchers have studied it at both the network layer and application layer. In particular, the relationship between the network-layer performance and the user-perceived game performance has been examined, and requirements on the network-layer performance for good gaming experience at the application-layer have been established. It was recommended in [7] that players should avoid servers with ping times over 150 ms or packet loss ratios over 3%. Through real user studies, the same authors also found that shooting is greatly affected by latency. With even modest latency (75-100 ms), accuracy and number of kills can be reduced by up to 50%. In addition, they found that users rarely notice performance degradation with packet loss under 5% during a typical network game [7]; however, the effect of higher than 5% loss ratios was commented on. Note that high loss ratios often occur in a wireless environment. Some of our experimental results showed that the loss ratio could be significantly higher than 5%; we thus include the loss ratio result in our study. The importance of latency is also found in another work in which an average round-trip-time (RTT) of 100 ms between a game client and a game server was suggested [19].

The IEEE 802.11 standard suite includes multiple modulation techniques, all of which use the CSMA/CA media access control (MAC) protocol. Although a new 802.11n standard is being developed which is said to be much faster, the most widely used ones today are 802.11b and 802.11g standards, which have a maximum raw data rate of 11 Mbps and 54 Mbps, respectively. Due to protocol overhead, the maximum throughput that an application can achieve is typically much lower than the above figures [6, 8, 20]. Both 802.11b and 802.11g support the base station (or infrastructure) mode as well as the ad hoc mode. The former assumes the presence of wireless access points when forming a Wi-Fi network, and mobile nodes communicate via these access points; the latter assumes the formation of a Wi-Fi network without any access point, and mobile nodes communicate with each other directly. On a Wi-Fi network, one can set a RTS/CTS threshold value in bytes; when a frame to be sent has a size larger than this threshold, the frame sender first reserves the channel using the RTS/CTS frames before sending the frame. This is to reduce collision thus to improve channel efficiency. In practice, however, most deployed Wi-Fi networks choose a large threshold value, essentially disabling the channel reservation function [12]. In our study, the base station mode 802.11g with RTS/CTS disabled is considered because this setup is commonly seen in most deployed Wi-Fi networks. For security reasons, 802.11 also has an optional encryption standard at the MAC layer called Wired Equivalent Privacy (WEP); each frame can be encrypted before transmission.

In the literature, the support of Wi-Fi networks to MOGs

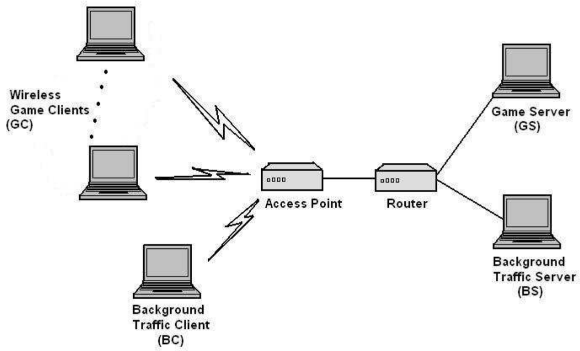

Figure 1: Experimental test bed

has been studied. In [15], the performance of two FPS games, namely *Half-life* [3] and *Quake 3* [4], was measured on an 802.11b network. The infrastructure mode with RTS/ CTS enabled was studied by way of experimentation. They found that 20 *Half-life* or 10 *Quake 3* game players take more than 3.5 Mbps of bandwidth, even though the actual required bandwidth is less than 1 Mbps. Differ from that work, in this paper, we focus on 802.11g without RTS/CTS and various factors' impact on games. In [14], the game traffic performance of FPS games on an 802.11g network was studied; a limited number of two factors were considered. It was found that constant-bit-rate UDP background traffic has a significant impact on game performance. In contrast to that work, we examine more factors and include more realistic background traffic, including both TCP and UDP traffic, namely the FTP and video streaming traffic. In [17, 16], a wireless home entertainment scenario in which multiple types of application share a Wi-Fi network was investigated. To adequately support online games, schemes at the MAC layer and transport layer were devised. These were achieved by modifying protocol parameters at these layers. As a result, the game performance was much improved. The performance metric used was delay jitter while in our study delay is targeted. In addition, simulation was used while our study is based on experimentation.

3. TEST BED DESCRIPTION

In this section, we describe the test bed, traffic model, and performance metrics, that we used for our evaluation.

3.1 Test Bed Description

Our test bed environment, depicted in Fig. 1, consists of 11 machines. These include:

(i) A game server (GS)—has a Pentium III 1.7 GHz CPU, and 512 MB RAM.

(ii) Eight game clients (GCs)—each has a Pentium III 733 MHz CPU, and 256 MB RAM.

(iii) A background traffic server (BS)—has a Pentium III 1.7 GHz CPU, and 1 GB RAM.

(iv) A background traffic client (BC)—has a Pentium III 733 MHz CPU, and 256 MB RAM.

All 11 machines are installed with the Linux operating system (Fedora 3, Kernel 2.6.9-1.667). Since we used emulated game traffic in our evaluation (see Section 3.2), graphics rendering power on the game clients is not considered in this study.

In our test bed, the GCs and the BC are on a wireless network, while the GS and the BS are on a wired network. Particularly, the GC machines are equipped with Linksys WMP54G Wireless-G PCI adapters, and the BC machine with a D-Link AirPlus G High-Speed 2.4GHz DWL-G510 wireless PCI adapter. Both GC and BC machines are associated with a wireless access point (CISCO AIRONET 1200 series, model#: AIR-AP1231G-A_K9). Note that an advantage of using networking equipments from different vendors is that our test bed might closely reflect a real-life scenario. On the wired network, the GS and the BS are connected with the wireless access point via a U.S. Robotics 8054 router. The entire test bed was set up in an office environment in a one-floor building, where there are offices and cubicles.

The wireless access point (AP) was configured using its default settings except that access control by MAC addresses on the AP was turned on and broadcasting SSID was disabled. It is worth noting that besides our Wi-Fi network, there are two other Wi-Fi networks that are in operation in the same building. We observe that their signal strength is comparatively weak (rated 3 to 5 out of 10 as seen from our test bed), when compared to that of our Wi-Fi network (rated 10 out of 10). We consider such an environment adequate for our purpose because in a practical wireless gaming environment, co-existence of multiple Wi-Fi networks may be likely, and some (low) degree of interference may be present.

3.2 Traffic Model

In our experiments, there are two classes of traffic on the test bed: game traffic and background traffic.

Game traffic is sent between individual GCs and the GS using UDP. We used a traffic emulator from [13] to generate the game traffic of a FPS game, namely *Half-life* [3]. Specifically, from the GS to a GC, one packet is sent every 60 ms; the packet size follows a lognormal distribution with an average of 203 bytes and a standard deviation of 0.31 bytes. When there is more than one GC in a game session, at every timeout (i.e., 60 ms), the GS sends one packet to each GC in a row, following the order in which the GCs join the session (or initially connect to the GS). From a GC to the GS, on the other hand, one packet is sent every 41.5 ms, whose size follows a normal distribution with parameters (71.57, 6.84) in bytes.

There are also two types of background traffic in our experiments: File Transfer Protocol (FTP) traffic and video streaming traffic. Both are considered popular on today's computer networks. The former is a type of TCP traffic that is elastic in nature and reacts to congestion conditions in the network; the latter is a type of UDP traffic that is more QoS demanding and does not perform any congestion control during transmissions. We used an FTP program provided by Fedora, called `ncftpget`, to download a large file from the BS to the BC. In our experiments, the FTP session is started before a game session starts, and is terminated after the game session is completed. So, the FTP session is provisioned to be long-lived.

The video streaming traffic is generated by an emulator that we developed. It sends video frames of various sizes according to video trace data described in [10]. Multiple video streams are included in our experiments. For each video stream, one video frame is sent every 40 ms. Based on the average bit rate of each trace, the number of video streams is varied to produce various levels of background video traffic. A total of seven video trace files are used in our study, and their bit rates range from 850 Kbps to 1.2 Mbps. In each experiment, all video streams are started in the first 40 ms, within which the actual start times are random. Similar to the FTP traffic, video streaming traffic is sent from the BS to the BC. This is under the assumption that users in a Wi-Fi environment may be more likely to act as content consumers rather than content producers. Our preliminary experimental results also indicated that the game traffic performance is much more significantly affected by the background traffic sent from the BS to the BC than that sent in the opposite direction. Again, in our experiments, all video sessions are started before the game traffic starts and are terminated after the game session ends.

3.3 Performance Metrics

In our evaluation, the performance metrics of interest are: (i) the average round-trip-time, RTT, from a GC to the GS, (ii) the packet loss ratio, LR_{s2c}, for game packets sent from the GS to the GCs, and (iii) the packet loss ratio, LR_{c2s}, for game packets sent from all the GCs to the GS. As described in Section 2, both latency and loss ratio performance at the network layer may have major impact on the game performance at the application layer.

RTT is collected using the "ping" utility that is provided by the operating system. To reduce the negative impact that is brought in by additional ping traffic, the time interval between consecutive ICMP Echo Request packets of the ping utility is set to 1 s. This is much longer than the 41.5 ms time interval between consecutive game packets that are sent from a GC to the GS. This interval is not too large either; so it can accurately capture the RTT encountered by the game traffic. When there is more than one GC in an experiment, the GC that last joins the game session is used to collect the RTT results. Based on preliminary experiments, the difference in RTT performance experienced by the various GCs was found to be minimal.

To calculate LR_{s2c}, two variables, the total number of game packets that are sent by the GS to all the GCs in an experiment, Ns, and the total number of game packets received by all the GCs from the GS in an experiment, Nc, are maintained. For the latter, each GC records the number of packets received; Nc is obtained from the sum of all these values. Then, LR_{s2c} is calculated by $(Ns - Nc)/Ns * 100\%$. Similarly, to calculate LR_{c2s}, the total number of packets that are sent by all the GCs, Nc', and the total number of packets that are received by the GS, Ns', are obtained from the game traffic generator program. LR_{c2s} is then given by $(Nc' - Ns')/Nc' * 100\%$. However, our preliminary results showed that LR_{c2s} never exceeded 1%; thus, we will not show the results for LR_{c2s} in the rest of this paper.

4. EXPERIMENTS AND RESULTS

In this section, we describe the experiments that we performed and report on the results that we obtained. We first make use of 2^k factorial experimental designs to identify important factors that affect the performance. We then investigate the impact of these important factors using an increased number of levels per factor in the experiments.

4.1 Effect of Distance on Performance

There are a total of five factors in our experiments:

(i) NC: the number of GCs in a game session,

(ii) Video: bandwidth usage of background video traffic,

(iii) FTP: with or without the background FTP traffic,

(iv) WEP: enabling or disabling of data encryption,

(v) Distance: the distance from the GCs to the AP.

To facilitate our investigation of the distance factor, we consider the following two cases. In Case I, we have one game client (NC=1) and we vary the distance between two levels. At distance level 1, the GC is placed in the "experiment room" where the AP is located, and the distance to the AP is around 10 feet. (See Fig. 2 for the layout of our experiment environment.) At distance level 2, the GC is placed in a room (at the top left corner in Fig. 2) that is around 75 feet away from the AP. Case I is to see the impact of distance when there is only one game client in the Wi-Fi network. In Case II, we increase the number of game clients to 8 (i.e., NC=8). Similar to Case I, two distance levels are under consideration. At distance level 1, all 8 GCs are located in the experiment room. These GCs form a circle with a diameter of around 10 feet and the AP is positioned in the center of the circle. There are no physical obstacles between the AP and the GCs. At distance level 2, four GCs are remained in the experiment room. The other four GCs are spread out in the office environment (see Fig. 2); one is 25 feet away from the AP, two are 50 feet away, and one is 75 feet away. We attempt to use this placement of GCs to model a Wi-Fi gaming environment where players may sit at different locations within the Wi-Fi network. As shown in Fig. 2, there are solid walls and cubicle dividers between these four GCs and the AP. Finally, in both cases, the GS, BS, and BC are located in the experiment room.

As to factors (ii) to (iv), we also select two levels for each of them. The 2^4 factorial designs for Cases I and II are illustrated in Table 1. Particularly, for Video, we choose 0 and 22 Mbps, where 22 Mbps represents a heavy load condition on an 802.11g network [6, 8, 20]. Also, when WEP is enabled, a 128-bit shared key encryption is used.

Each experiment is performed for a duration of 8 minutes. This length is considered representative in a typical FPS

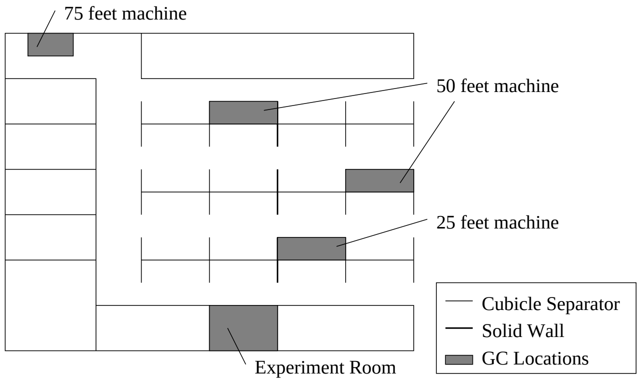

Figure 2: Test bed environment layout

Table 1: 2^4 Factorial Experimental Designs for Cases I and II

Factor	**Case I (NC=1)**	**Case II (NC=8)**
Video (Mbps)	[0, 22]	[0, 22]
FTP	[with, without]	[with, without]
WEP	[enabled, disabled]	[enabled, disabled]
Distance level	[1, 2]	[1, 2]

game session [9]. Each experiment is also repeated for six times. The 95% confidence intervals are computed. Since our results show that the width of the confidence intervals are extremely small when compared to the sample mean, we only report on the sample mean of our performance metrics.

Based on the results obtained from our experiments, we calculate the percentages of variation in the results that are explained by each factor and their interactions. This is summarized in Tables 2 and 3 for Cases I and II, respectively. Note that only those factors and factor interactions whose percentages of variation are larger than 1% are shown in these tables. We find that in both cases, FTP, Video, and their interaction account for most of the variation in the results of RTT and loss ratio. In contrast, distance, WEP, or any factor interaction involving these two factors explains lower than 1% of variation. We conclude that for the scenarios of interest where distance is varied, the distance and WEP factors do not have a significant impact on the game traffic performance.

Table 2: Percentage of Variation Explained in Case I (%)

Metric	FTP	Video	FTP+Video
RTT	86.21	5.16	7.5
Loss ratio	65.55	30.95	2.96

Table 3: Percentage of Variation Explained in Case II (%)

Metric	FTP	Video	FTP+Video
RTT	82.87	2.23	11.25
Loss ratio	27.75	45.00	24.53

4.2 Significance of Other Factors on Performance

Based on the above findings, in all subsequent experiments, we fix the distance to level 1, meaning that we position all GCs in the experiment room. To learn the significance of the other four factors on performance, we conduct another 2^4 factorial design for these factors. For NC, two levels are experimented: 1 and 8. The upper bound of 8 is selected mainly due to our resource availability The levels for Video, FTP, and WEP are the same as shown in Table 1.

After obtaining the experimental results, we calculate the percentages of variation in the performance metrics that are explained by the factors and their interactions. These percentages are shown in Table 4. As before, only the terms larger than 1% are shown in the table. It can be observed that for RTT, only FTP, Video, and their interaction account for larger than 1% of variation. For loss ratio, NC also shows non-negligible effect. Again, the factor of WEP does not have significant impact on performance. In the following sections, we will focus on FTP, Video and NC and provide a detailed analysis of their impact on performance. In this analysis, we keep our distance at level 1 and enable WEP encryption.

Table 4: Percentage of Variation Explained (%)

Metric	FTP	Video	FTP+Video	NC	NC+FTP
RTT	86.51	2.23	10.30		
Loss ratio	33.51	24.73	4.62	24.89	9.70

4.3 Effect of FTP and Video Traffic

We first examine the effect of FTP traffic. In Fig. 3, we plot the results for RTT against the amount of video streaming traffic when with and without FTP traffic for the case of NC=1. The corresponding results for loss ratio are shown in Fig. 4. Note that in these results, the highest video load level is 26 Mbps. This is higher than the 22 Mbps used in the 2^4 designs above. The higher levels are selected to illustrate what would happen when the wireless channel is approaching saturation. On the other hand, in the 2^4 designs, it was expected that the video load level is kept at 22 Mbps or lower in order to avoid starvation of other traffic on a shared 802.11g network. The conclusions from 2^4 designs above would remain valid had 26 Mbps been used instead.

We observe that when there is only one GC, the inclusion of FTP traffic greatly lowers the performance of game traffic. With FTP traffic, when the amount of video traffic varies, RTT ranges from 15 to 25 ms; without FTP traffic, RTT is below 10 ms for most video load levels. As to the loss ratio, the largest difference between with and without FTP traffic approaches 3%.

The performance degradation when there is FTP traffic can be explained as follows. FTP traffic is delivered using the TCP protocol at the transport layer. TCP congestion control uses network loss events to infer the congestion level in

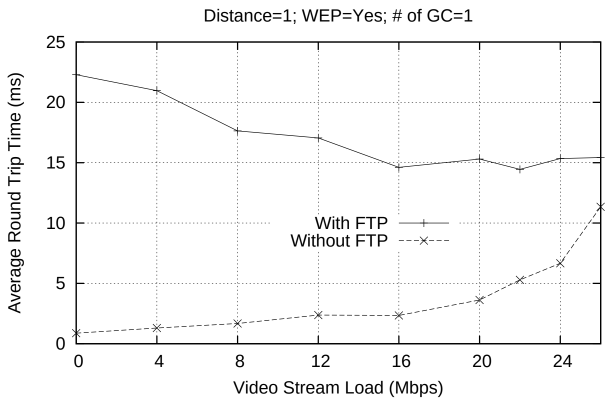

Figure 3: RTT vs. amount of video traffic (NC=1)

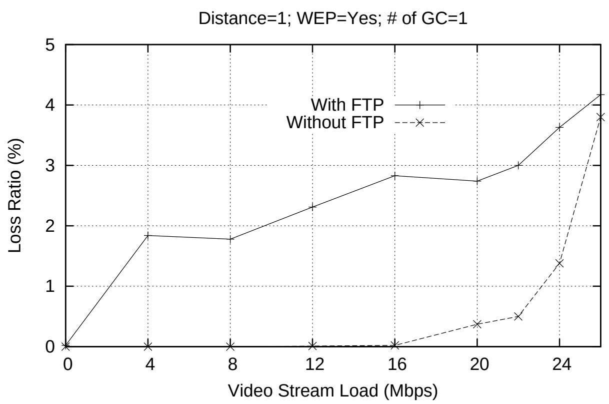

Figure 4: Loss ratio vs. amount of video traffic (NC=1)

the network and attempts to explore and use the maximum available bandwidth in the traffic path. When a queue (at a router or an AP) inside the network is overflowed, loss events occur, and congestion control is triggered. This implies that TCP may tend to keep the queue length within the buffer size limit. At the same time, because TCP endeavors to make use of the available bandwidth as much as possible, it tends to keep the bottleneck queue relatively full. In our test bed network, the major queue in the network is the one at the AP along the server to client direction. When the video load is low to moderate and there is no FTP traffic, the queue is relatively short, yielding good RTT and loss ratio performance. With FTP traffic, however, the queue is relatively full with the FTP traffic. With first-come-first-served (FCFS) scheduling at this queue, the game traffic will experience a much longer queue, when compared to the case of no FTP traffic, resulting in worse performance. When the video traffic load is high enough (e.g., above 24 Mbps), even without FTP traffic, the wireless channel is approaching saturation and the queue starts to overflow. In this case, it is observed that the performance is equally bad regardless if there is FTP traffic or not. These observations indicate that QoS strategies may be needed in order to well support game traffic when there is FTP background traffic in the network.

The absolute values from these experiments indicate that, using the recommended upper bound of 3% on game traffic loss ratio [7], when there is just one game client with FTP, background video should be kept below 22 Mbps; without FTP, it should be kept below 25 Mbps.

We next examine the impact of video traffic on the game traffic performance. In Fig. 3, we observe that without FTP traffic, the RTT performance deteriorates as the video traffic load is increased because the average queue length grows as the load is increased. Contrarily, with FTP traffic, the RTT performance actually becomes slightly better as the video traffic load is increased. This may be because as the video load becomes higher, with TCP congestion control, the FTP traffic backs off more significantly, leading to better RTT performance. For the loss ratio, we observe in Fig. 4 that for both levels of FTP, the higher the video traffic load, the higher the loss ratio. Without FTP traffic, the loss ratio has a sharp increase between the video load levels of 24 and 26 Mbps; with FTP traffic, this degradation of loss ratio is more gradual. This is because of the "damping" effect of TCP traffic, which affects the dynamics of the queue length. Specifically, when there is only UDP traffic, there would be a sharp increase in queue length as the channel approaches saturation. But with FTP traffic, the queue tends to stay relatively full all the time and such a sharp increase in queue length at high video traffic load would not occur.

The RTT and loss ratio performance when NC=8 is plotted in Figs. 5 and 6, respectively. Similar to the case of NC=1, both graphs show large performance degradation with FTP traffic. As to the impact of video streaming traffic on performance, without FTP traffic, the same trend as in the case of NC=1 is observed on the results for both RTT and loss ratio. With FTP traffic, on the other hand, it is important to note that although some fluctuation, the RTT of game traffic stays at around 20 ms regardless of the amount of background video. This may be because the buffer at the AP is almost full nearly all the time. Hence, on average the game traffic always encounters queueing delay that approximates the time it takes to service a full buffer of data. For the loss ratio, it slowly increases when the amount of video traffic is lower than 20 Mbps. This is due to the fact that the TCP congestion control effectively reduces the FTP traffic sending rate so that the loss at the AP buffer is maintained at a low level. However, when the video traffic load is increased above 20 Mbps, the amount of video traffic is so large that even if TCP remains at the slow-start phase (meaning that the FTP traffic is at the minimal sending rate), the channel is already overloaded by the UDP traffic. Therefore, a sharp increase in loss ratio occurs. This observation implies that in order to achieve good game loss ratio performance, the amount of video traffic needs to be controlled.

Our experimental results also indicate that when the number of game clients is 8, with FTP the game traffic loss rate exceeds 3% irrespective of video load level, thus for adequate game support FTP traffic should be kept to a minimum. Without FTP, the amount of video should be kept below 22 Mbps.

We conclude that on our 802.11g network, the inclusion of FTP traffic, which is delivered using the TCP protocol, sig-

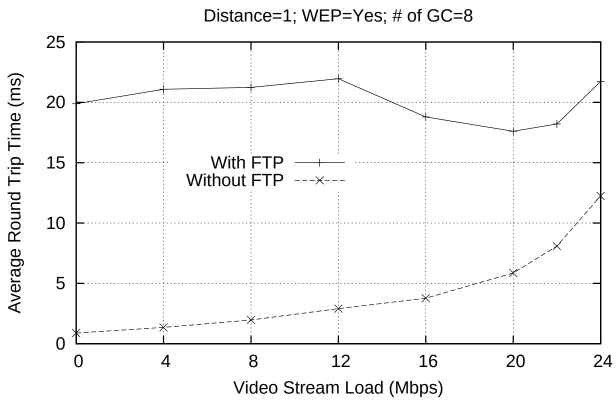

Figure 5: RTT vs. amount of video traffic (NC=8)

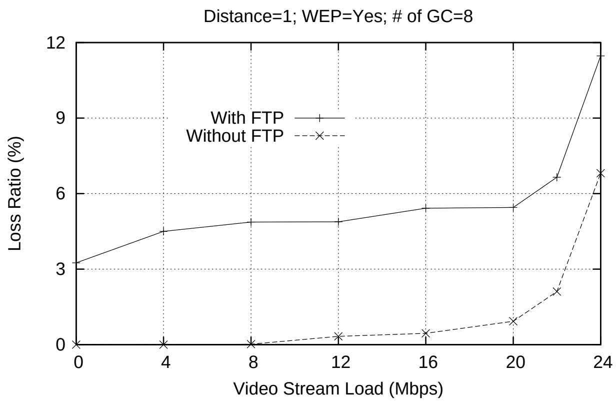

Figure 6: Loss ratio vs. amount of video traffic (NC=8)

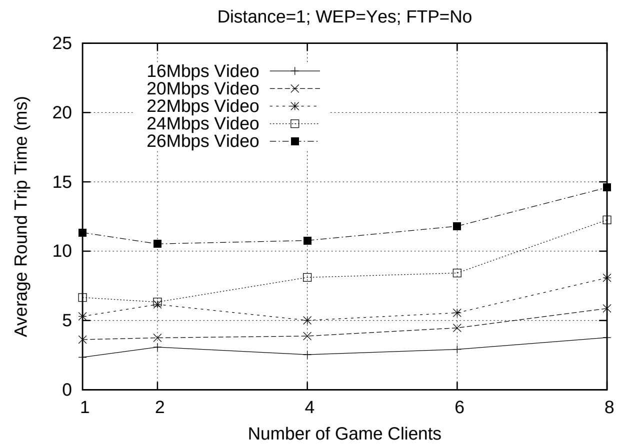

Figure 7: RTT vs. number of GCs (no FTP)

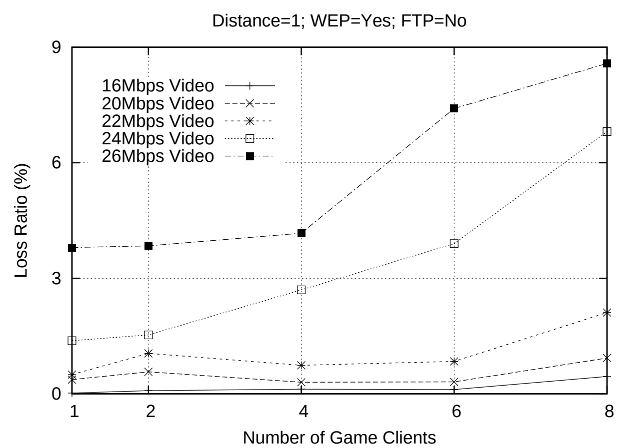

Figure 8: Loss ratio vs. number of GCs (no FTP)

nificantly affects the game traffic performance. Similarly, the amount of video background traffic, which, like the game traffic, is delivered using UDP, also has large impact on the game traffic performance. To provide good gaming experience, proper QoS strategies are needed.

4.4 Effect of NC

To evaluate the effect of NC, combinations of video and FTP traffic are considered. For each combination, NC is varied from 1 to 8. Consider first the scenario where there is no FTP traffic. The results for RTT and loss ratio are shown in Figs. 7 and 8, respectively. We observe that NC has large impact on performance, especially the loss ratio, at heavy video traffic load. This phenomenon may be explained as follows. In a game session, at regular time intervals, the server sends state update messages back-to-back to all GCs. These updates form a burst—a train of packets. The higher the number of GCs, the longer the burst. When the video traffic load is heavy, the queue at the AP is almost full, a longer burst would result in higher loss of game packets.

Consider next the presence of FTP traffic. The RTT and loss ratio performance is shown in Figs. 9 and 10, respectively. It can be observed that when there is background FTP traffic, the RTT of game traffic is kept at a high level regardless of the number of GCs. This is because when with FTP traffic, the queue is kept at an “almost full” level by TCP congestion control. With FCFS channel scheduling, queueing delay encountered by the game traffic is about the same, which corresponds to the time it takes to empty (or serve) an almost-full queue. In contrast, the number of GCs does have a large impact on the loss ratio performance of game traffic. The more GCs, the higher the loss ratio. This can be explained by the relationship between the loss ratio and the state update burst size, as described previously for the case of no FTP traffic. Particularly, with FTP traffic, the buffer at AP is kept relatively full. When NC is increased, a longer burst of game packets arrives at the queue, resulting in a higher chance of buffer overflow.

We conclude that the number of GCs largely affects the loss ratio performance experienced by game traffic. When there is no FTP traffic, this holds only when the video traffic load is high. With FTP traffic, this holds for all video load levels. For the RTT performance, when there is no FTP traffic, the number of GCs has some impact also when video traffic load is high. However, the RTT results always stay at the range of 15 to 25 ms when the FTP traffic is present.

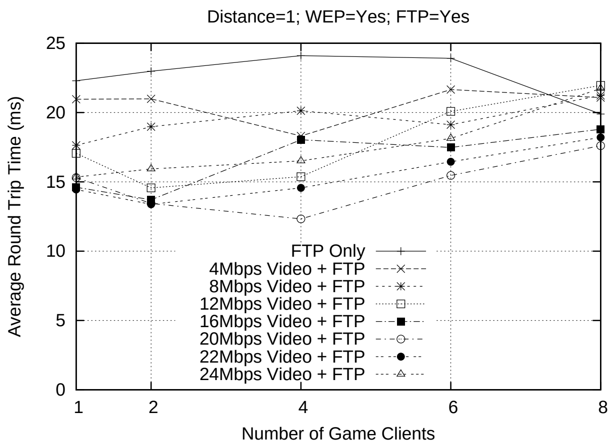

Figure 9: RTT vs. number of GCs (with FTP)

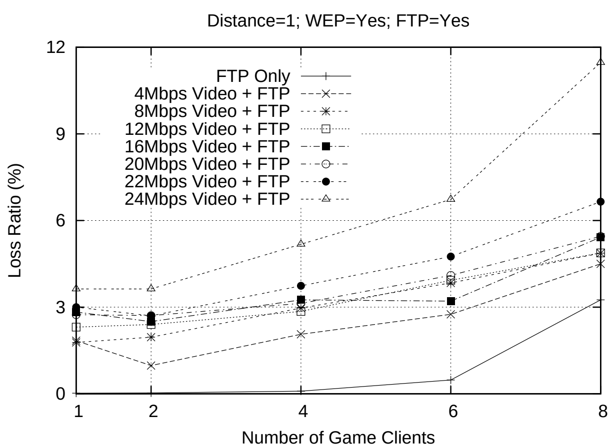

Figure 10: Loss ratio vs. number of GCs (with FTP)

5. QOS STRATEGIES IN A WI-FI GAMING ENVIRONMENT

In this section, we discuss possible QoS strategies that one may devise to better support games in a Wi-Fi environment that may be shared by FTP and video background traffic. Assuming no change is to be made at the transport layer, i.e., with TCP and UDP, we discuss strategies at the application layer and inside the network respectively.

At the application layer, a possible strategy is to throttle both FTP and video background traffic when game sessions are on-going. This may imply certain restrictions on the applications that other wireless clients sharing the same Wi-Fi network with the game clients can run. From our study, we have observed that both FTP and video traffic significantly affect the game traffic performance, especially when the aggregated load level is high; therefore, throttling their traffic may leave ample bandwidth on a Wi-Fi network to game traffic and better support the game sessions.

Inside a Wi-Fi network, QoS strategies refer to the queue management and channel scheduling algorithms inside the AP at the MAC layer. A possible strategy is to employ priority-based scheduling in which a higher priority is given to the game traffic. In terms of implementation, a separate queue may be maintained for the game traffic, and the other types of traffic share another queue. Whenever the channel becomes idle, the channel scheduler serves the traffic in the game traffic queue, if it is non-empty; otherwise, the other queue is attended. Within each of these two queues, FCFS scheduling may suffice. Prioritizing game traffic may pose a problem when the total bandwidth consumed by the game traffic is high because this may lead to the game traffic starving the other traffic. However, FPS games have been identified to have low bit rate. Thus, this strategy can be viable with moderate game traffic load. In addition, this strategy may require cross-layer processing in order to identify the game traffic carried within MAC-layer frames. Evaluating the performance implications of this overhead is a topic of further research.

Comparing the above two strategies, the one at the application layer does not require any modification at the AP. The downside is that it may result in a low network utilization. The strategy at the MAC layer, on the other hand, requires modification at the AP, but may yield higher network utilization. Further investigation of these two strategies is part of our future work.

6. CONCLUSIONS

In this paper, using an experimental approach, we have evaluated the impact of various factors, namely the distance between a game client and the wireless access point, the enabling of data encryption, the inclusion of FTP and video streaming background traffic, and the number of game clients on the game traffic performance in terms of RTT and loss ratio performance on an IEEE 802.11g network. We have found that FTP and video streaming traffic significantly affect the performance. The number of game clients is important when the aggregated background traffic load is high. In particular, it has higher impact on loss ratio than on RTT. Comparatively, distance and WEP have minimal effect. Finally, based on our observations, we have suggested two QoS strategies at the application and MAC layers, that may be used to improve the support of a Wi-Fi network to a MOG. As suggestions for future research, proper pricing schemes for QoS strategies in a Wi-Fi gaming environment can be studied. Furthermore, experimentation that links network-layer performance and user-level performance such as user satisfaction can be carried out.

7. ACKNOWLEDGMENTS

We gratefully acknowledge TRLabs Winnipeg and the Department of Computer Science at the University of Manitoba for their support to this project. We also thank anonymous reviewers for providing very insightful feedback and suggestions, which helped improve the quality of our paper. Many thanks to John Tromp who helped with polishing the graphs and some text.

8. REFERENCES

[1] Counter-strike. http://www.counter-strike.net/.

[2] First-person shooter. http://en.wikipedia.org/wiki/Firstperson_shooter.

[3] Half-life. http://half-life.sierra.com/.

[4] Quake 3. http://www.idsoftware.com/games/quake/quake3-gold/.

[5] Unreal Tournament. http://www.unrealtournament.com.

[6] Atheros Communications. *Methodology for testing wireless LAN performance with Chariot*. http://www.atheros.com/.

[7] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in Unreal Tournament 2003. In *Proceedings of the 2004 ACM SIGCOMM Workshop on NetGames (NetGames 2004)*, Aug. 2004.

[8] X. Cao, G. Bai, and C. Williamson. Media streaming performance in a portable wireless classroom network. In *Proceedings of the IASTED European Workshop on Internet Multimedia Systems and Applications (EuroIMSA)*, pages 246–252, Feb. 2005.

[9] M. Claypool, D. LaPoint, and J. Winslow. Network analysis of Counter-strike and Starcraft. In *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference*, Apr. 2003.

[10] F. H. Fitzek and M. Reisslein. Mpeg-4 and h.263 video traces for network performance evaluation. http://www-tkn.ee.tu-berlin.de/research/trace/ltvt.html.

[11] IEEE-SA Standards Board. *ANSI/IEEE Standard 802.11*. IEEE, 1999.

[12] J. F. Kurose and K. W. Ross. *Computer networking: a top-down approach featuring the Internet*. Addison Wedley, 3rd edition, 2005.

[13] T. Lang, G. Armitage, P. Branch, and H. Choo. A synthetic traffic model for Half-life. In *Proceedings of the Australian Network and Telecommunications Conference 2003*, 2003.

[14] Y. E. Liu, J. Wang, M. Kwok, J. Diamond, and M. Toulouse. Capability of IEEE 802.11g networks in supporting multi-player online games. In *Proceedings of the 2nd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME 2006)*, Jan. 2006.

[15] T. Nguyen and G. Armitage. Quantitative assessment of IP service quality in 802.11b and DOCSIS networks. In *Proceedings of the Australian Network and Telecommunications Conference*, 2004.

[16] C. E. Palazzi, G. Pau, M. Roccetti, S. Ferretti, and M. Gerla. Wireless home entertainment center: Reducing last hop delays for real-time applications. In *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2006)*. ACM Press, June 2006.

[17] C. E. Palazzi, G. Pau, M. Roccetti, and M. Gerla. In-home online entertainment: Analyzing the impact of the wireless MAC-transport protocols interference. In *Proceedings of the IEEE International Conference on Wireless Networks, Communications, and Mobile Computing (WIRELESSCOM 2005)*, June 2005.

[18] H. Park and T. Kim. Network game traffic modeling. In *Proceedings of the 1st Workshop on Internet ad Network Economics (WINE 2005)*, Dec. 2005.

[19] P. Quax, P. Monsieurs, W. Lamotte, D. Deleeschauwerand, and N. Degrande. Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. In *Proceedings of the 2004 ACM SIGCOMM Workshop on NetGames (NetGames 2004)*, Aug. 2004.

[20] A. Wijesinha, Y. Song, M. Krishnan, V. Mathur, J. Ahn, and V. Shyamasundar. Throughput measurement for UDP traffic in an IEEE 802.11g WLAN. In *Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks*, pages 220–225, May 2005.