Capability of IEEE 802.11g Networks in Supporting Multi-player Online Games

Yanni Ellen Liu*, Jing Wang†, Michael Kwok‡, Jeff Diamond§ and Michel Toulouse¶
*†¶Department of Computer Science, University of Manitoba, Manitoba, Canada
Email: {yliu, jingwang, toulouse}@cs.umanitoba.ca
‡School of Computer Science, University of Waterloo,Waterloo, Canada
Email: kfkwok@uwaterloo.ca
§TRLabs, Winnipeg, Canada
Email: diamond@win.trlabs.ca

***Abstract*— Multi-player online games have become very popular in the last few years. Meanwhile, the IEEE 802.11 wireless networks have been in wide use. In this paper, we present an experimental study on the capability of an IEEE 802.11g network in supporting multi-player online games. In particular, we focus on the highly interactive first-person-shooter games. We describe the test bed we set up and the experiments we performed. Factors such as the number of game clients and the amount of background traffic are examined. Our results show that the amount of background traffic has a significant impact on the latency and the loss ratio of the game traffic between the game clients and the game server, which in turn affect the observed game performance greatly.**

I. Introduction

Multi-player online games (MOGs) are computer games where multiple players simultaneously participate in a game session over a data network, for example, the Internet. The MOG market has developed rapidly over the last few years. Its popularity is substantiated by the availability of high-speed residential access networks and the affordable high performance personal computers. Moreover, MOGs offer an appealing entertainment experience, allowing the interaction with other human players and creating a sense of adventure. In general, MOGs can be classified into first-person-shooter (FPS) games (e.g., *Counter-strike* [1], *Half-life* [2], and *Unreal Tournament*[3]), real-time strategy games (e.g., *StarCraft*[4] and *Age of Empire*[5]), massive multi-player role-playing games (e.g, *Everquest*[6] and *Lineage*[7]), simulator games (e.g., *Grand Prix* 3[8]), and turn-based games (e.g., *Panzer General 3D: Assault*[9]).

MOGs are often implemented using a client-server architecture. Perhaps the most important requirement of a MOG implementation is to provide the game players with a consistent view of the game environment in a timely manner. Achieving so in a distributed environment involves the transmission of state update messages between the game clients and the game server. Different types of games may have different quality-of-service (QoS) demand on the underlying network. Among them, FPS games often have the most stringent requirements on network delay and loss ratio.

Most game clients today are connected to the corresponding game servers via wired networks. Over the past few years, wireless data networks such as the IEEE 802.11 (Wi-Fi) [10] networks have gained wide deployment. Wi-Fi network access points are now commonly seen in coffee shops, office buildings, university campus, airports, and many residential homes. The capacity of Wi-Fi networks has also kept increasing. The Wi-Fi network interface has become a standard built-in on today's laptop computers. In view of these advances, it is believed that participating in a MOG from a wireless environment will 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 Section II for details.) In contrast, less attention has been paid to the investigation of MOGs in wireless networks. In this work, we take an experimental approach and evaluate the capability of an IEEE 802.11g network in supporting FPS games. Specifically, the game *Half-life* [2] is experimented. We choose the IEEE 802.11g Wi-Fi networks and the FPS games because of their popularity. We choose the FPS games also because they place the highest demand on the underlying network transmissions.

In the literature, there have been other studies on the performance of FPS games in a Wi-Fi environment. However, most of these works were carried out using simulations. Differ from these studies, in this work, we utilize a real test bed and perform our study using experimentation. In order to focus on the capability of a Wi-Fi network only, in this study, we assume that the game servers are located on the same local area network as the Wi-Fi access point to which the game clients are connected. The scenario where a Wi-Fi network acts as an access network to a wide area network (WAN), while the game servers are remotely located from game clients is not considered. It should be noted however, that our results would provide useful insight to QoS support to games in such a wide-area wireless/wireline environment, when combined with results from WAN performance studies.

Various factors affect the game performance in a Wi-Fi network. These include the number of wireless clients, the amount of non-game traffic, the wireless protocol, and the physical environment parameters (such as the distance and the clearance of sight between wireless clients and access points, the humidity, and the interference with other wireless devices). In this paper, we study the first two factors as our first step and quantify the impact of each on the game performance. In what follows, we will refer to the non-game traffic as the *background traffic*. Our initial results show that in terms of latency and loss ratio experienced by the game traffic, the amount of background traffic has the major impact. In our test bed environment, when there are up to seven game clients, the amount of background traffic needs to be kept below 16 Mbps in order to ensure a good game play quality.

The rest of the paper is organized as follows. In Section II, we provide some background information on the FPS games and on the IEEE 802.11 networks. In Section III, we describe the test bed we set up and the experiments we performed to evaluate the capability of an 802.11g network in supporting FPS games. The results are presented and discussed in Section IV. Finally, Section V contains our concluding remarks and some suggestions for future research.

II. Background and Related Work

In this section, we discuss the traffic characteristics and performance requirements of FPS games, and provide some background information on the IEEE 802.11 Wi-Fi networks.

A. FPS

In a FPS game, the game world is rendered as what is seen by each player's eyes. A player undertakes a specified task during a game session, and may encounter and open fire at enemies during the task course. When a player makes a move, e.g., moving one step forward or firing a bullet at an enemy player, two state update messages are generated and sent in sequence: one is sent from the player's computer to the game server, and the other is sent from the game server to the computers of those players who are affected by this move. The shorter the network delay in delivering the state update messages, the more realistic the playing experience. Due to their fast paced actions, FPS games often place the highest demand on the performance of the underlying networks.

Many previous works have been carried out to study the traffic characteristics and the performance of FPS games in wired networks. For example, in [11], the traffic characteristics of the game *Counter-strike* was measured over a wired local area network. It was shown that from a game server to individual game clients, the mean size of packets is 127 bytes, and the arrival rate of these packets is 16 packets/s. From a game client to the server, the mean packet size is 82 bytes and the packet arrival rate is 24 packets/s. As another example, in [12], the game *Half-life* was studied. It was found that from a game server to individual clients, the packet size ranges from 60 to 300 bytes, whereas from clients to the server packets are of size between 60 and 90 bytes. The arrival rates of packets from a server to clients and from a client to a server are very close to those shown in [11]. Similar results were also observed by many other researchers in the field. In our experiments, we therefore assume that typical FPS games generate small sized packets at a relatively low bit rate.

Researchers have studied the relation between the network-level performance and the user-perceived game performance, and established the requirements on network-level performance for adequate support of FPS games. In [13], Beigbeder *et al.* experimented with the game *Unreal Tournament*. Several game clients and a game server are connected in a controlled LAN environment. A traffic emulator is located on the paths between the game clients and the game server to emulate the delay and loss within the wide-area Internet. This way the messages exchanged between the game clients and the game server appear to have traversed a wide area network upon their arrivals at the receivers. Three levels of performance, namely the user-level, the application-level, and the network-level, were collected. They found that an average round-trip-time of 75 ms or lower between a game client and the game server is noticeable. An average of 100 ms or above makes the game less enjoyable. A packet loss ratio of 3% or higher becomes evident in affecting the game performance. These results are consistent with another study in [14] which found that a latency of 60 ms or less is desired. We will use the 60 ms latency and the 3% packet loss ratio as the network-level performance targets in our study.

B. IEEE 802.11 Wi-Fi Networks

The IEEE 802.11 standard suite includes multiple modulation techniques, all of which use the CSMA/CA media access control (MAC) protocol. The most widely used ones are 802.11b and 802.11g standards. The 802.11b networks have a maximum raw data rate of 11 Mbps, while the 802.11g networks have a maximum raw data rate of 54 Mbps. In practice, due to protocol overhead, the maximum throughput that an application can achieve is much lower than the above figures [15], [16], [17]. For example, in [17], the throughput of UDP traffic in an 802.11g network was measured. It was found that in almost all cases the observed throughput of UDP traffic is well below 50% of the 802.11g maximum data rate.

Both 802.11b and 802.11g support two operating modes: the base station (or infrastructure) mode and 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.

The support of Wi-Fi networks to real-time applications have been studied previously. In [16], a media streaming application was experimented on an 802.11b network. It was observed that the maximum throughput achieved was 4.6 Mbps. In [18], the performance of two FPS games, namely *Half-life* and *Quake 3* [19] was measured on an 802.11b network. They found that twenty *Half-life* or ten *Quake 3* game players take more than 3.5 Mbps of bandwidth, even though the actual required bandwidth is less than 1 Mbps. In [20], an interesting study has been carried out to quantify the impact of the MAC-layer protocol in IEEE 802.11g networks on the

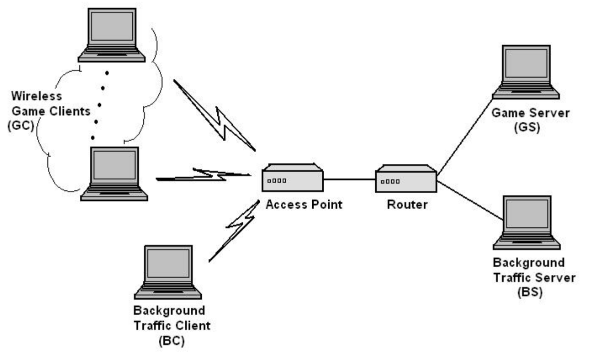

Fig. 1. Experimental test bed

upper-layer performance, when the network is carrying in-home entertainment traffic. Various types of traffic including FPS games, video chat, streaming media, and file transfer were utilized. Differ from our study, a number of MAC-layer protocol and physical environment related parameters were considered. Performance on delay, jitter, and TCP throughput were reported. The cases when with one game player were studied.

In this paper, the newer 802.11g network will be studied. The base station mode will be considered because this mode is commonly seen in the deployed 802.11g networks. Multiple game clients will be included.

III. Test Bed and Experiments

In this section, we describe the test bed we set up for our evaluation as well as the design of our experiments.

A. Test Bed Description

Our test bed environment, depicted in Figure 1, consists of 10 machines. These include:

- (i) A game server (GS)—has a Pentium III 1.7 GHz CPU, 512 MB RAM, and 60 GB hard disk.
- (ii) Seven game clients (GCs)—each has a Pentium III 733 MHz CPU, 256 MB RAM, and 13 GB hard disk.
- (iii) A background traffic server (BS)—has a Pentium III 1.7 GHz CPU, 1 GB RAM, and 40 GB hard disk.
- (iv) A background traffic client (BC)—has a Pentium III 733 MHz CPU, 256 MB RAM, and 13 GB hard disk.

All ten machines are installed with the Linux operating system (Fedora 3, Kernel 2.6.9-1.667).

The client machines (the GCs and the BC) are equipped with Linksys WMP54G Wireless-G PCI adapters, and are associated with a wireless access point (CISCO AIRONET 1200 series, model#: AIR-AP1231G-A_K9). The GS and the BS are on a wired network, and are connected together with the wireless access point via an U.S. Robotics 8054 Router.

The test bed was set up in a relatively isolated environment where there is no physical obstacles between the wireless access point and all the client machines. All wireless clients formed a circle with a diameter of around 3 meters. The access point was positioned in the center of the circle. The access point was configured using its default settings except that the access control by MAC addresses on the wireless access point was turned on and the broadcasting SSID was disabled. This way, only the wireless clients on our test bed can access our Wi-Fi network, other wireless machines or devices in the area are prevented from getting onto our Wi-Fi network. Nevertheless, in addition to 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 rated 3 to 5 out of 10 as seen from our test bed, which is deemed low, compared to the signal strength in our Wi-Fi network, which is rated 10. To reduce the interference from those two networks, we configured our access point to make use of the least busy channel and performed our experiments in evenings and weekends. We consider such an environment adequate for our study because in a targeted wireless gaming environment, co-existence of multiple Wi-Fi networks may be likely and some (low) degree of interference may be present.

B. Traffic Model

Two types of traffic are generated and sent in our experiments: game traffic and background traffic. The game traffic is sent between individual GCs and the GS. We developed a traffic emulator for the game *Half-life* to generate the game traffic. The traffic model in [12] is used. From the GS to individual GCs, on average 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 are more than one GCs in the game session, at each timeout, the GS sends one packet to each GC in a row. From each GC to the GS, on average one packet is sent every 41.5 ms, its size follows a normal distribution with parameters (71.57, 6.84) in bytes. At the beginning of an experiment, the GS waits for all GCs to initiate a connection. After they do so, the GS sends packets to the GCs back-to-back in a row at each timeout, following the order in which the connections were first initiated.

Background traffic, on the other hand, is generated and sent from the BS (wired) to the BC (wireless). This direction was chosen because our initial experiments indicated that the network-level performance is much more significantly affected by the background traffic sent from the BS to the BC, when compared to that in the opposite direction. The generated background traffic shares the Wi-Fi network bandwidth with the game traffic. A C program was written to generate and send messages with a fixed size at regular time intervals using UDP. Various levels of offered load were experimented. To decide on the length of the regular time interval, we first evaluated the accuracy of the operating system timer that we used. By way of Ethereal [21], we found that when using a 10 ms interval, the background traffic bandwidth achieved is only 90% of what is specified. With a 50 ms interval, the accuracy reaches 99%; thus, we used 50 ms in our experiments. The message size is calculated based on the level of offered load of the background traffic. For example, to achieve an offered background traffic load of 16 Mbps, a message size of 100 Kbytes was used. If the size of a message is larger than the maximum UDP payload length, the message is fragmented into multiple UDP segments which are then sent back-to-back to the BC. Background traffic is started at the beginning of each experiment.

C. Performance Metrics

The performance metrics of our experiments are: (i) the packet loss ratio, LR_{s2c}, for game packets from the GS to the GCs, (ii) the packet loss ratio, LR_{c2s}, for game packets from all the GCs to the GS, and (iii) the average round-trip-time, RTT, from a GC to the GS. As mentioned in Section II, these metrics have also been used in existing studies.

To calculate LR_{s2c}, two counters, Ns and Nc are used. Ns records the total number of game packets that are sent by the GS to all the GCs in an experiment. Nc is the total number of game packets received by all the GCs from the GS in an experiment. 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 calculated by $(Nc' - Ns')/Nc' * 100\%$.

Finally, RTT is collected using the "ping" utility that is provided by the operating system. To reduce the negative impact brought 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 ms time interval between consecutive game packets that are sent to the network. This interval is not too large either; so it can capture the actual RTT encountered by the game traffic throughout an experiment. When there is more than one GCs in an experiment, the GC that last joins the game session is used to collect the RTT results. Our experiments showed that the difference in RTT performance experienced by all the GCs is minimal.

D. Experimental Design

As the first step of our research, we experimented with two factors, namely the number of GCs (NC) and the amount of background traffic (BT), and studied their impact on the game performance. NC denotes the total number of GCs that are simultaneously accessing the GS via the experimental Wi-Fi network. BT denotes the bandwidth usage of the background traffic. We are interested to learn that in a typical shared 802.11g network, how many game clients and how much background traffic can be accommodated.

We first used a 2^2 factorial experimental design to determine the relative importance of each factor. The lower and upper bound levels for NC were chosen to be 1 and 7 respectively. The upper bound of 7 was selected mainly based on our available resource. The lower and upper bound levels for BT were chosen to be 16 and 32 Mbps respectively. The lower bound of 16 Mbps was chosen because when the amount of background traffic is below this level, we observed that the network-level performance meets the target values — an average round-trip-time of 60 ms and a packet loss ratio of 3%. We chose an upper bound of 32 Mbps to represent a heavy load condition.

TABLE I
RESULTS OF THE 2^2 FACTORIAL DESIGN

	BT = 16 Mbps	BT = 32 Mbps
NC = 1	$RTT = 4.19$ ms $LR_{s2c} = 2.61\%$ $LR_{c2s} = 0.00\%$	$RTT = 12.92$ ms $LR_{s2c} = 9.23\%$ $LR_{c2s} = 0.00\%$
NC = 7	$RTT = 4.54$ ms $LR_{s2c} = 3.32\%$ $LR_{c2s} = 0.00\%$	$RTT = 13.81$ ms $LR_{s2c} = 11.83\%$ $LR_{c2s} = 0.06\%$

TABLE II
EXPLANATIONS OF THE VARIATIONS IN RESULTS

Factor	RTT	LR_{s2c}	LR_{c2s}
NC	0.47%	4.50%	33.33%
BT	99.44%	94.03%	33.33%
$NC\&BT$	0.09%	1.47%	33.33%

Each experiment is performed for a duration of 8 minutes. This length was considered representative of a typical FPS game session [22]. Each experiment is repeated multiple times. We computed the 95% confidence intervals of these results, which turn out to be extremely narrow, and in many cases are barely visible in the graphs. Therefore, we only report on the sample mean results.

IV. Results

In this section, we report on the results of our experiments and present the major findings of our study.

A. Results of 2^2 Factorial Design Experiments

The raw performance results of the 2^2 factorial design are shown in Table I. In Table II, for each of the three performance metrics, we calculate the amount of variation explained by each factor and by the interaction of the two factors to show their relative importance. We observe that (1) for RTT, the amount of variation explained by factor BT is over 99%. Factor NC and the interaction of the two factors together explain less than 1% of variation in the results. Thus, we conclude that the RTT performance in our testing scenarios is dominated by the amount of background traffic. (2) The game packet loss ratio from the GS to the GCs is also mainly affected by the amount of background traffic. (3) However, each factor and the interaction of the two factors explains 1/3 of variation for the loss ratio of game traffic from the GCs to the GS. Hence, there is no dominating factor for this performance metric.

We now study in detail the effect of each factor on the network-level performance of game traffic.

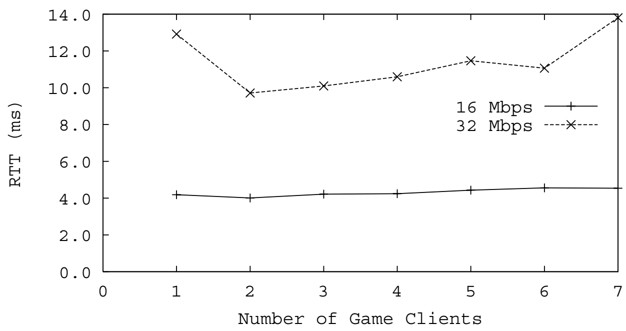

Fig. 2. Round-trip-time vs. the number of game clients

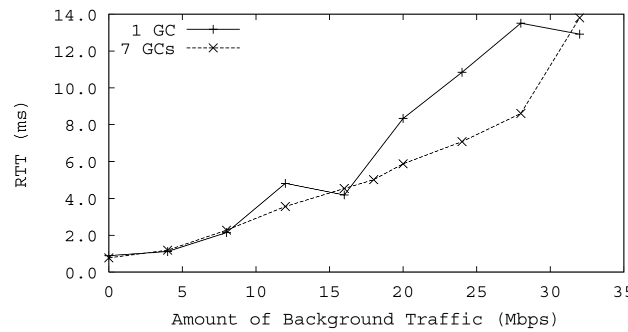

Fig. 4. Round-trip-time vs. the amount of background traffic

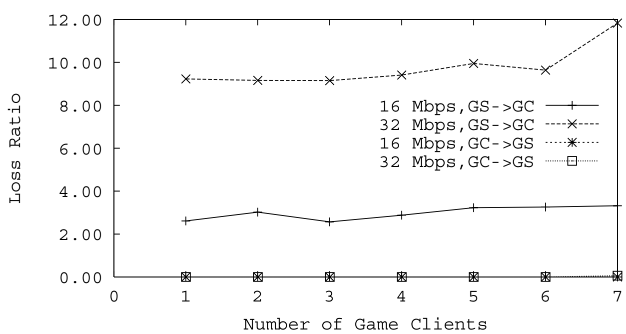

Fig. 3. Loss ratios vs. the number of game clients

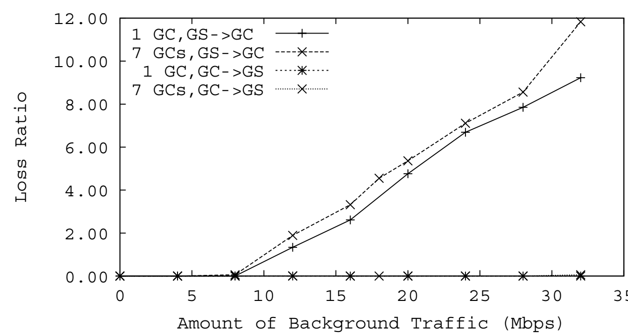

Fig. 5. Loss ratios vs. the amount of background traffic

B. Effect of the Number of Game Clients

The effect of NC on performance is shown in Figures 2 and 3. In Figure 2, we plot the RTT performance of the game traffic against the number of game clients, for BT = 16 and 32 Mbps. We observe that (1) the RTT when BT = 16 Mbps is much lower than that when BT = 32 Mbps. This is as expected because the higher the load of the background traffic, the less capacity that is left to the game traffic, thus the higher latency. (2) For both values of BT, RTT is well below the target 60 ms level. (3) RTT is not significantly affected by the number of game clients for the scenarios that we experimented. We thus conclude that in terms of game traffic RTT, when the number of game clients in an 802.11g network is seven or below, a large amount of background traffic can be offered to the network without jeopardizing the game performance.

In Figure 3, we plot the loss ratios as a function of the number of game clients for both the GS→GC and the GC→GS directions. The two curves corresponding to the loss ratios along the GC→GS direction are collapsed to the x-axis; when there is no background traffic along this direction, the loss ratios for game traffic are always near zero. On the contrary, the other two curves showing the loss ratios along the GS→GC direction for BT = 16 and 32 Mbps are above the x-axis. We observe that (1) the loss ratio slightly increases as the number of game clients increases beyond 3. The increase is more noticeable when the amount of background traffic is higher. We thus conclude that in terms of the loss ratio for traffic along the GS→GC direction, which is also the direction of the background traffic, the number of game clients has noticeable but slight impact on the performance.

We also noticed that when there are more than one GCs, looking at the individual GCs loss ratio results, the loss ratio increases following the order in which the GS sends packets to the GCs. This is because the packets sent by the GS at each timeout form a train of arriving packets at the access point, the ones sent earlier in the train are more likely to be able to enter the queue, and are less likely to be dropped, the ones sent later in the train are more likely to encounter a full queue, and thus be dropped. We further noticed that the difference between the highest and the lowest loss ratio results among the GCs increases as the number of game clients is increased.

C. Effect of the Amount of Background Traffic

The effect of BT on performance is shown in Figures 4 and 5. In Figure 4, the average RTT experienced by game traffic as a function of the amount of background traffic is plotted. The cases with NC = 1 and 7 are presented. We observe that (1) as the amount of background traffic increases, for both cases, RTT increases. (2) Regardless of the amount of background traffic, the values of RTT are all well below the target 60 ms level. We thus conclude that in terms of average RTT, the experimental 802.11g network is adequate for supporting the FPS game of our choice, even though the network may be heavily loaded with background traffic.

The results for the game traffic loss ratios when the amount of background traffic is varied are shown in Figure 5. We can observe that (1) the two curves for the results along the GC→GS direction when NC = 1 and 7 are very close to the x-axis. This implies that when there is no background traffic, with game traffic only, the loss ratio is near zero. (2) Nonetheless, along the GS→GC direction, when with background traffic, the loss ratios are far from zero. When $NC = 1$, the loss ratio increases from zero to above 9% as BT increases from zero to 32 Mbps. When $NC = 7$, the loss ratio increases to almost 12%. We thus conclude that the amount of background traffic greatly affects the loss performance when the background traffic is sharing the wireless network capacity with the game traffic. If using the 3% target loss rate for good game performance, the amount of background traffic should be kept below 16 Mbps for the cases when there are up to seven game clients.

One way to tackle the inferior loss performance encountered by the game traffic when presented with large amount of background traffic could be by way of prioritized buffer management. Our delay performance results indicated that as long as the game traffic is not dropped at the access point, the delay performance will be adequate for games. Thus a simple remedy could be for the access point to give game traffic lower dropping precedence; whenever possible, the scheduler always drops other types of traffic when the queue is full, rather than dropping the game traffic. However, for this scheme to work, cross-layer processing may be needed in order to identify the game traffic among other types of traffic.

V. Conclusions

We investigated the capability of an 802.11g wireless network in supporting a FPS game using an experimental approach. Factors such as the number of wireless game clients and the amount of background traffic were examined. It was found that in terms of the latency and loss ratio performance experienced by the game traffic, when there are up to seven game clients in the wireless network, the amount of background traffic has the major impact on performance. In order to ensure a good game playing experience, the amount of background traffic should be kept below 16 Mbps.

Due to the limitation on the amount of available resources, we only experimented with up to seven GCs. As part of our future work, more game clients can be added. In addition, more factors such as those related to the physical environment may be tested. In this work, we only experimented with one type of background traffic, in future, a wider variety of background traffic, e.g., the HTTP traffic, can be included. Furthermore, the support to multiple FPS games in one Wi-Fi environment can be studied.

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 feedbacks and suggestions, which helped improving the quality of our paper.

References

[1] "Counter-strike," Sierra, http://www.counter-strike.net/.
[2] "Half-life," Sierra, http://half-life.sierra.com/.
[3] "Unreal Tournament," Atari, http://www.unrealtournament.com.
[4] "Starcraft," Blizzard, http://www.blizzard.com/starcraft/.
[5] "Age of empire," Microsoft, http://www.microsoft.com/g̃ames/ẽmpires/.
[6] "Everquest," Sony, http://www.everquest.com/.
[7] "Lineage," NCsoft, http://www.lineage.com/.
[8] "Grand Prix 3," MicroProse, http://www.grandprixgames.org/.
[9] "Panzer General 3D: Assault," Ubi Soft Entertainment, http://panzergeneral3.com/pzg3d/.
[10] IEEE-SA Standards Board, *ANSI/IEEE Standard 802.11*. IEEE, 1999.
[11] J. Färber, "Network game traffic modeling," in *Proceedings of the 1st ACM Workshop on Network and System Support for games (NetGames2002)*, Apr. 2002.
[12] 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.
[13] 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 (NetGames2004)*, Aug. 2004.
[14] 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 (NetGames2004)*, Aug. 2004.
[15] *Methodology for testing wireless LAN performance with Chariot*, Atheros Communications, http://www.atheros.com/.
[16] 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)*, Feb. 2005, pp. 246–252.
[17] 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*, May 2005, pp. 220–225.
[18] 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*, 2004.
[19] "Quake 3," idSoft, http://www.idsoftware.com/games/quake/quake3-gold/.
[20] 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.
[21] "Ethereal," http://www.ethereal.com.
[22] 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.