802.11 Wireless LAN Multiplayer Game Capacity and Optimization

Hanghang Qi and David Malone
Hamilton Institute
National University of Ireland, Maynooth, Ireland
Email:{hanghang.qi,david.malone}@nuim.ie

Dmitri Botvich
TSSG
Waterford Institute of Technology, Ireland
Email:dbotvich@tssg.org

Abstract—**Real-time multiplayer games are a popular application of networks, and as IEEE 802.11 wireless networks are widely used, games are expected to be widely played on wireless networks. However, 802.11 networks with the normal MAC (DCF) may present some challenges when supporting real-time games traffic. In this work, we present a theoretical model which can predict the performance and capacity of 802.11 WLAN for Quake 4, a real-time FPS game. Using the wireless network game model, we derive throughput, delay, jitter and MOS (mean opinion score) as the number of game clients increases in the WLAN and predict that the capacity of a default 802.11b wireless network can support around 10 players for Quake 4. In 802.11e networks, QoS (quality of service) is provided with 4 configurable MAC layer parameters. With the network game model, we show that with proper TXOP configuration at the AP and game server, the network can be optimized and its capacity improves to around 15 players.**

I. INTRODUCTION

Multiplayer real-time games have become a popular network application recently. Online games have drawn more people, and games traffic takes up a reasonable portion of the Internet traffic [1], [2], [3]. Meanwhile, wireless networks, especially IEEE 802.11 WLAN, have emerged as a common last-hop in the Internet. Playing multiplayer real-time games in a wireless network is an obvious future direction. In this paper, our interest is in the wireless LAN's performance and support for multiplayer real-time FPS games. In particular, we will consider a situation where a group of players, and the game server are all wirelessly connected by an 802.11 access point (wireless LAN party), as might happen at university dorms.

We expect 802.11 networks to face certain challenges before supporting real-time multiplayer games. 802.11 uses a CSMA/CA based MAC layer. Bianchi's model of the 802.11 MAC [4] has proven to be accurate and useful, and it has been extended to different network conditions and traffic loads (e.g. [5], [6]). Our goal is to build a model for real-time multiplayer games over 802.11 using these models.

Throughput, delay (latency), jitter and packet loss are important factors for network games [7], [8]. Wireless networks may present challenges in latency-sensitive applications, including VoIP and real-time games, as wireless could introduce extra delay and jitter because of the CSMA/CA [9]. In this model, we focus on WLAN's throughput, delay and jitter. As various compensation techniques are usually used, packet loss is considered less important than delay and jitter [8].

Once we have built our model of 802.11 network to predict loss, delay and jitter, we show how using the 802.11e MAC layer may improve the network performance. In [10], the capacity of voice in 802.11 WLAN has been considered and it is shown that the access point can become the bottleneck. We show that with Quake 4 in an 802.11 infrastructure mode network, the AP and games server are potential bottlenecks when the number of stations increases. If it is an 802.11e network and priority is given to the AP and game server by properly selecting MAC parameters, the capacity can be increased from 10 stations to 15 stations.

We use Quake 4 in this paper for two main reasons: it is a typical and popular FPS game and it is well studied by the research community. We expect our results to extend to other games which use similar topologies. The rest of the paper is organized as follows: Section II presents the traffic characteristics of Quake 4 as measured in our Wireless network. Section III presents our 802.11 model for network games, and throughput, delay and jitter are calculated. Section IV presents our optimization scheme using the 802.11e MAC parameter TXOP. In Section V, results from a normal DCF network and the optimized scheme are shown and compared. Finally, conclusions and future works are presented in Section VI.

II. QUAKE 4 TRAFFIC IN WIRELESS NETWORK

In this section we measure game traffic using our wireless testbed in a typical indoor environment in order to compare it to wired networks and derive the parameters for our model. All the stations are about 5-10 meters away from the AP. The testbed consists of 4 identical desktop PCs. One acts as the AP, one acts as the game server and other two are game clients. The Quake 4 Server is in spectate mode. Two players played the game on client PCs. Measurement is done at the AP PC using tcpdump [11]. Key features are shown in Fig. 1 to Fig. 4.

Fig. 1 illustrates the packet transmission rate. The number of packets transmissions in every second is calculated and shown. It is clear from the figure that the transmission rate is almost constant: around 65 packets per second from the clients and 14 packets per second from the server. Fig. 2 shows the distribution of interarrival times. There are high peaks observed in both client-to-server and server-to-client traffic. Looking at Figs 1 and 2 together, we see that Quake 4 traffic can reasonably be considered to have constant packet rate over
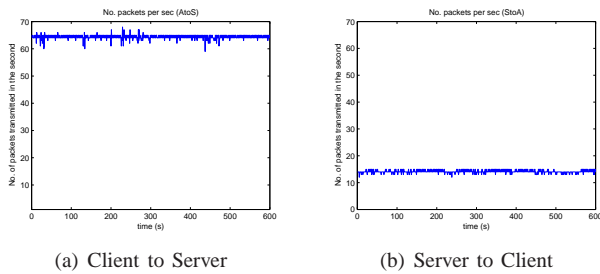
Fig. 1. Quake 4 packets transmission rate: packets per second in wireless network

(a) Client to Server   (b) Server to Client

Fig. 2. Quake 4 packet transmission interarrival time histogram in wireless network

(a) Client to Server   (b) Server to Client

Fig. 3. Quake 4 packets size histogram in wireless network

Fig. 4. Quake 4 traffic: Bytes per seconds in wireless network

Fig. 5. StoC packet size vs. players number

a good wireless network and that the rate from client to server is higher than from server to client.

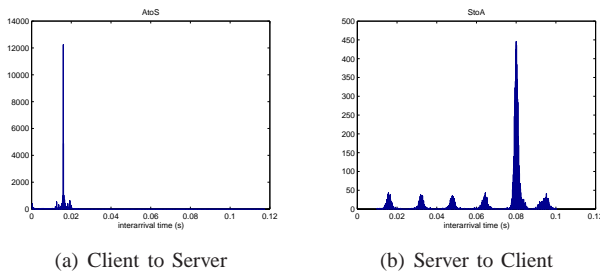Now, consider Fig. 3, which shows the packet size distribution. The packet size is spread in a range and it is larger from server to client than from client to server. It is known that the packet size exhibits correlation over time and can be modeled as an ARMA(1,1) process [12]. Fig. 4 illustrates bytes per second. We can see that the network throughput fluctuates in a considerable range.
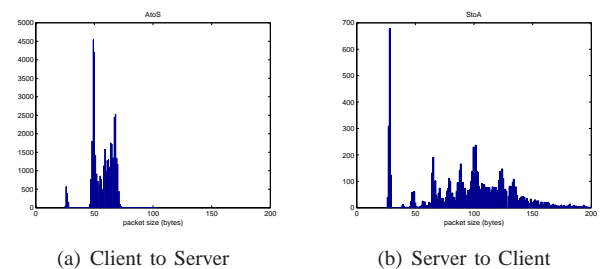
Comparing our wireless results with previous studies from wired networks [13], [14], [15], we see that the game traffic behavior is similar (as expected). In our two-player testbed, game performance is good. Our assumption will be that if the network can support the traffic indicated by scaling-up appropriately, then game performance will be good. When network becomes congested, as the number of clients (players) increases, delay and jitter will become too large, and the network will fail to support the games.
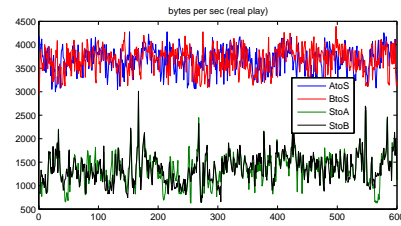
As we have confirmed that game traffic is similar in wireless and wired networks, we will take the results from wired network that server to client packet size distribution increases linearly as the number of players increases [9], [12], [16]. Important quantities for the model include average packet size EP and average collision packet size EP*. EP* [4] is defined as $EP* = E[max(EP1, EP2)]$. Using data from the SONG database [17], packet size distribution can be acquired, and EP and EP* are calculated. Fig. 5 shows the result of the average packet size and average collision packet size from 2 players to 7 players. Linear fits are made to predict results for larger numbers of players.
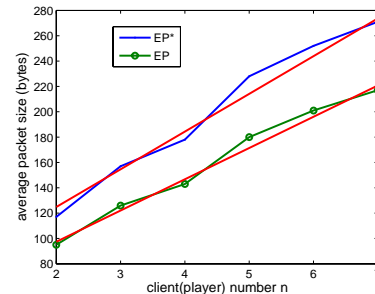
To summarize, game traffic characteristics are constant packet rate with varying packet length (ARMA) transmission between Server and Clients. Transmission rate and packet size distributions are different between server and client. Packet size distribution does not change much from client to server, while sizes are increased with the number of players from server to client. The Quake 4 traffic parameters are summarized in Table I. They are the traffic input in our network model.

	Client to Server	Server to Client
Packet transmission rate	λ_c=65(packets/s)	λ_s=14(packets/s)
Average packet length	EP_c=57.24(bytes)	EP_s=24.8n+45.4(bytes)
Average collision packet length	$EP*_c$=61.32(bytes)	$EP*_s$=30n+60(bytes)

TABLE I
QUAKE 4 GAME TRAFFIC CHARACTERISTICS. n IS THE NUMBER OF PLAYERS.

III. IEEE 802.11 MAC SCHEME AND NETWORK MODEL

A. IEEE 802.11 CSMA/CA

An IEEE 802.11 infrastructure network with the DCF (distributed coordination function) MAC uses a CSMA/CA (carrier sense multiple access with collision avoidance) scheme with binary slotted exponential backoff. Briefly, when stations with packets to send sense the wireless medium is idle for a period of $DIFS$, each station goes to a count down state, and counts down a uniform random number chosen from the interval [0,CW-1]. While the medium remains idle, each station decreases the number by 1 after a slot time δ, until some station reaches 0 when the station transmits the packet (802.11e allows more than one packet to be transmitted). If the packet is successfully transmitted to its destination, the destination sends an ACK frame after a period of $SIFS$. Once the other stations receive the ACK frame, they know the medium is idle again, and they resume their count down. If two or more stations happen to reach 0 at the same time they transmit their packets simultaneously and a collision happens. Destinations are not successful in receiving and thus no ACK frame is sent back. After a period if those stations do not receive an ACK frame, they know collision happened. They will try to resend their packet. They come into a new count down state, where CW is doubled. After a successful transmission, CW is reset to the value CW_{min}. CW can be doubled to CW_{max} and then it will not change if there are further collisions. If number of failures of a particular packet reaches a limit, the packet is dropped and a new packet is processed. 802.11e MAC enables the values of the MAC parameters $DIFS$ (called $AIFS$), CW_{min}, CW_{max} and $TXOP$ to be set on a per class basis for each station, with a maximum of four classes. We will be interested in the parameter $TXOP$, which allows a variable sized group of packets to be transmitted when the backoff counter reaches zero.

Note that the 802.11 MAC in infrastructure mode also requires that stations do not communicate directly, but forward all packets through the access point. Thus, when calculating the load on our network, we must factor in the extra work done by the AP. While the 802.11e MAC does include an extension, DLS (direct link setup), to allow direct station-to-station communication, we do not consider that in this paper.

B. Two-dimensional Markov Chain model of 802.11 MAC

The MAC of an 802.11 network can be modeled as a two-dimensional Markov Chain model (shown in Fig. 6), which can be used in saturated or unsaturated heterogeneous networks [4], [5], [6].

Each state $\{s(t), b(t)\}$ in the 2-dimensional Markov Chain represents the count down state of one station. Each row represents one stage of the backoff process. Every time a packet fails to transmit, the MAC moves to next backoff stage. The first row models the non-saturated case which allows heterogeneous packet arrival rates to be considered [6]. Different packet arrival rates lead to different q parameters for
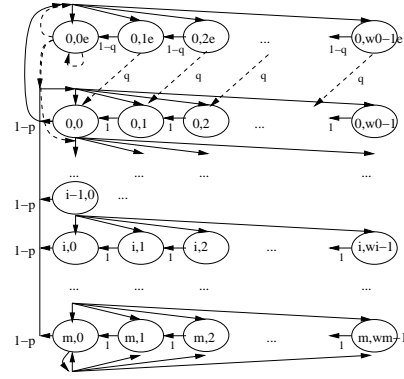

Fig. 6.   Markov chain model of 802.11 MAC

each class of station, and so the collision rate p is different. In the wireless network games problem, we consider 3 classes of station: AP, game server and client. Each class has different input rates and also performance. Generally, the AP and game server act like hubs of the network, where they have higher packet arrival rates than the clients.

Through the Markov chain, the collision probability p and transmission probability τ are entangled together. After solving a group of nonlinear equations, collision probability p and transmission probability τ of each class can be calculated (see [6] for details). Using these, we can get $P_{tr} = 1-(1-\tau)^n$, the probability that there is at least one transmission in a state, and

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}$$

the probability of a successful transmission occurring on the channel, conditioned on at least one station transmitting. The normalized system throughput can then be expressed as the ratio

$$S = \frac{E[\text{payload information transmitted in a state time}]}{E[\text{length of a state time}]}.$$

Using $E[P]$, the average packet payload size, σ, the duration of an empty state time, T_s, the average time the channel is considered busy during a successful transmission, and T_c, the average time the channel is considered busy during a collision, we can write:

$$S = \frac{P_s P_{tr} E[P]}{(1-P_{tr})\sigma + P_{tr}P_s T_s + P_{tr}(1-P_s)T_c}. \quad (1)$$

We also calculate the channel access delay from the model. We work in terms of the length of a state L and the number of states it experiences D. Then $delay = E[D] = E[\sum_{i=1}^N L_i] = E[N]E[L]$ as N and L are independent and L_i are i.i.d. random variables. $E[L]$ is the average length of a state time. $E[N]$ is average number of slots which is expressed after some algebra as

$$E[N] = \frac{W_0}{2}\frac{1-(2p)^{m+1}}{1-2p} + \frac{W_0 p(2p)^m}{2}\frac{1}{1-p} + \frac{1}{2}\frac{1}{1-p}. \quad (2)$$

This result is the same as Chatzimisios's [18].

We can also use this technique to calculate the jitter of the network channel access time. Jitter is defined as the variance

of delay:

$$jitter = V[D] = E[D^2] - E[D]^2 = E[(\sum_{i=1}^{N} L_i)^2] - E[D]^2$$

$$= E[N]E[L^2] + E[N^2]E[L]^2 - E[N]E[L]^2 - E[N]^2E[L]^2$$

where $E[L^2] = (1 - P_{tr})\sigma^2 + P_{tr}P_sT_s^2 + P_{tr}(1 - P_s)T_c^2$ and $E[L]$ and $E[D]$ are known, $E[N^2]$ is to be calculated as $E[N^2] = \sum_i N_i^2 p(N_i^2)$

After some algebra, including applying $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$ and combining items in the same column in two-dimensional Markov chain, the expression finally reduces to

$$E[N^2] = [\frac{1}{3}\frac{1 - (4p)^{m+1}}{1 - 4p} + \frac{1}{3}\frac{2^{2m}p^{m+1}}{1 - p} + \frac{1}{2}\frac{4p(1 - (4p)^m)}{1 - 4p}$$
$$- \frac{p(1 - (2p)^m)}{1 - 2p} + \frac{1}{2}\frac{p^{m+1}}{1 - p}(2^m - 1)2^m + \frac{1}{2}2^{2m}\frac{p^{m+1}}{(1 - p)^2}]W_0^2$$
$$+ [\frac{1}{2}\frac{1 - (2p)^{(m+1)}}{1 - 2p} + \frac{1}{2}2^m\frac{p^{(m+1)}}{1 - p} + \frac{p(1 - (2p)^m)}{(1 - 2p)^2}$$
$$- \frac{1}{2}\frac{m(2p)^{m+1}}{1 - 2p} + \frac{p(1 - (2p)^m)}{1 - 2p} - \frac{1}{2}\frac{p(1 - p^m)}{1 - p}$$
$$+ \frac{1}{2}\frac{p^{m+1}}{1 - p}((m + 1)2^m - 1) + 2^m\frac{p^{m+1}}{(1 - p)^2}]W_0$$
$$+ \frac{1}{6}\frac{1}{1 - p} + \frac{1}{2}\frac{p(1 - p^m) + mp^m(p - 1)}{1 - p} + \frac{1}{2}\frac{p^{m+1}}{(1 - p)^2} \quad (3)$$

Now with $E[N^2]$, $E[L^2]$ and $E[D]$, we can get jitter $V[D]$.

The MAC parameters of the 802.11b network (11Mbps) used in the model are listed in Table II. We also note that assumptions of this model include Poisson arrivals and a small amount of buffering at each station. The model has been shown to approximate constant packet rate traffic [6]. As we will see in the Section III-C, games have been shown to be more sensitive to delay than loss, so a small buffer should be well adapted to games traffic.

802.11b parameters	Durations(μs)
Slot time, σ	20
Propagation delay, δ	1
$CW_{min} = 32\sigma$	640
DIFS(AIFS=0)	50
SIFS	10
PLCP Header@1Mbps	192
MAC Header 24 Bytes@1Mbps	17.5
CRC Header 4 Bytes@1Mbps	2.9
IP Header 20 Bytes@11Mbps	14.5
MAC ACK 14 Bytes@1Mbps	11.2

TABLE II
802.11 NETWORK MAC PARAMETERS

C. Game MOS with Delay and Jitter

A good experience for players is an essential part of game play. The effect of objective factors such as delay (latency), jitter and packet loss on game performance have been widely studied [7], [8]. Other than these objective factors, people's subjective experience of games has been measured with Mean Opinion Score (MOS). MOS is a subjective measurement of games which is based on people's judgment [19], [20], [21]. Recently, there has been work to use objective factors in order to understand MOS [20], [21]. We will use the work in [21], where the MOS score is predicted based on two quantities, the *ping average* and the *jitter average*.

The ping average is the average of 100 pings from client to server and 100 pings from server to client. These both represent full round trip time for the systems, we thus estimate them from our model using $ping\_average = D_C + D_{AP} + D_S + D_{AP}$, where D_C is the delay predicted for a client, D_{AP} is the delay predicted for an AP, and D_S is the delay predicted for the server.

In [21] the jitter average is calculated as follows; 300 packets are sent from client to server, 50ms apart. The arrival times are noted and the shortest time shifted to zero. Then the mean of these is taken giving the mean client to server delay less the min client to server delay. This process is repeated with 300 packets from server to client, and the mean of the two results is taken. In our case, since the min delay is relatively small, we estimate this by $jitter\_average = (D_C + D_S)/2 + D_{AP}$.

Following [21], the *network impairment* is given by $X = 0.104 * ping\_average + jitter\_average$, where the units are milliseconds. The mapping for the MOS is then given as $MOS = -0.00000587X^3 + 0.00139X^2 - 0.114X + 4.37$.

IV. AVOIDING THE BOTTLENECK AT AP AND SERVER

In a normal 802.11 infrastructure network with the DCF MAC, all the stations, including the AP, have an equal opportunity to access the channel. As the MAC is based on chances to transmit single packets, when the network becomes busy each station can transmit roughly equal numbers of packets, regardless of their size. With the different packet loads at the Server, Client and AP, this equal sharing of transmission chances may become a problem. The AP and server will become the bottleneck of the network as they have more packets to transmit than the clients and the situation will become worse as the number of clients increases. Our idea is to effectively give more packets transmission opportunities to the AP and server using 802.11e's TXOP mechanism and see how it improves the network capacity and performance.

When there are n players playing game, we set the AP and server's TXOP to be n times the Clients' TXOP. TXOP effectively allows the AP and server to have longer packets which consist of many small packets[1]. Thus, in the model, the AP has the same transmission opportunity rate as clients, and Server has fixed transmission opportunity rate. The effect is that AP, Server and Client still have the same opportunity to access the medium, but AP and server transmit more packets in each opportunity. It has been shown that network performance improves for voice traffic using a similar scheme [10]. In following sections we show that it also improves the network performance substantially for multiplayer games.

---

[1]Though note that if a collision occurs, then the TXOP terminates after the first packet because of the missing ACK. Indeed, we model TXOPs as large packets that have short collisions.

Since the games server knows the number of players, it can easily set its TXOP correctly. A question remains about how an access point would determine the number of players. One possibility is that a mechanism such as UPnP could be used by the game server to configure the AP's 802.11e settings. Another option is that the access point could identify game traffic using port numbers, or some similar mechanism, and apply an appropriate 802.11e configuration.

V. Results and Performance

A. Basic network with DCF

As we noted, a default infrastructure network with DCF, gives the AP, game server and clients all the same opportunity to access the medium. It is this basic DCF network we consider here. Fig. 7(a) shows the network structure and packet arrival rate at each station. Packet sizes in different classes are as indicated in Table I. With our model, which assumes Poisson arrivals of the same rate, throughput, throughput efficiency, delay, jitter and MOS are calculated and shown from 7(b) to 7(f).

(a) Structure.

(b) Total throughput

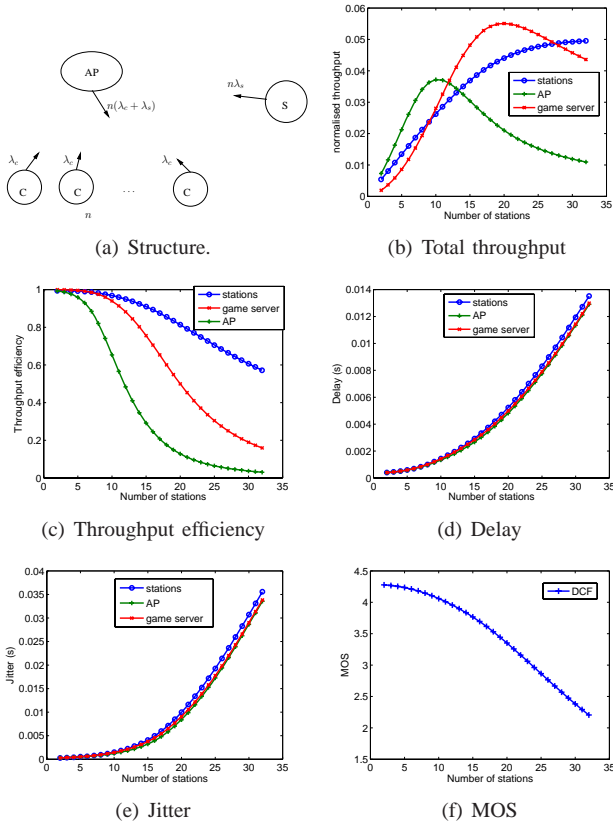(c) Throughput efficiency

(d) Delay

(e) Jitter

(f) MOS

Fig. 7.   No prioritization

Fig. 7(b) and (c) show the throughput of each class of the three in two forms: total throughput and throughput efficiency. In Fig. 7(b), AP reaches the peak at about 9 players and then drops; server reaches the peak at about 18 players and then drops. It indicates that they become congested and then with continued increase in number of clients, their throughput

decreases. Stations' total throughput reaches peak at about 24 players and then stays almost constant. In general, the network becomes congested before 9 players as the AP's throughput begin to decrease. Fig. 7(c) shows the throughput efficiency which is the ratio of output throughput and input throughput. It is shown that the throughput efficiency of AP drops below 0.6 at about 10 players. While previous research has shown that games are resilient to loss, 40% loss indicates severe congestion and is the highest level of loss considered in [21].

Fig. 7(d) and (e) show the delay and jitter. They initially increase gradually, but become steeper as the number of players increases. Fig. 7(f) shows the MOS. It decreases gradually for fewer than 10 players, and then drops more steeply up to 20 players and then more decreases gradually again. From this figure, it is observed that the MOS becomes less than 4 over 7 to 8 players and 3.8 over 10 players. Overall, based on the results observed in Fig. 7, the capacity of a default DCF 802.11b network is about 10 players.

B. AP and Server priority with TXOP

In this scheme, priority is given to the AP and server, as described in Section IV. The larger $TXOP$, which depends on player number n, is given to AP and server so that they can transmit multiple packets consecutively. The network structure and effective packet transmission rate are shown in Fig. 8(a). Again, the same performance indicators as in Fig. 7 are derived and shown in Fig. 8.

Fig. 8(b) and (c) show the total throughput and throughput efficiency respectively. Compared to Fig. 7, the AP and server's performance are greatly improved. The AP's throughput reaches a turning point at a little over 15 players and server almost maintains increasing throughput. Clients reach their peak at about 17 players. In Fig. 8(c) of throughput efficiency, all the 3 classes start to drop over 10 players and drop below 0.6 over 15 players. Compared to the basic scheme, it appears that it can support more players before the network becomes congested.

Fig. 8(d) and (e) show the delay and jitter. Compared with Fig. 7(d) and (e), the transition to the network being congested is more obvious and consistent across AP, server and clients. For MOS, shown in Fig. 8(f), it now decreases very slowly initially and then drops quickly after about 15 players. The MOS drops below 4 at about 13 players and 3.8 at 15 players. Throughput efficiency also remains above 0.6 at 15 stations. All in all, the capacity of the network with optimization seems to be around 15 players.

For the comparison, the MOS of both schemes are drawn in Fig. 9. From this figure, we can see that optimized network MOS is above default DCF network within 12 players at MOS of 3.9. Though the predicted MOS is lower for the optimized scheme above 12 players, it is possible that the substantially lower levels of loss achieved by the optimized scheme may result in better game play in practice.

VI. Conclusions and Future Work

In this paper, we present a theoretical model to predict the performance of 802.11 infrastructure WLAN with multiplayer
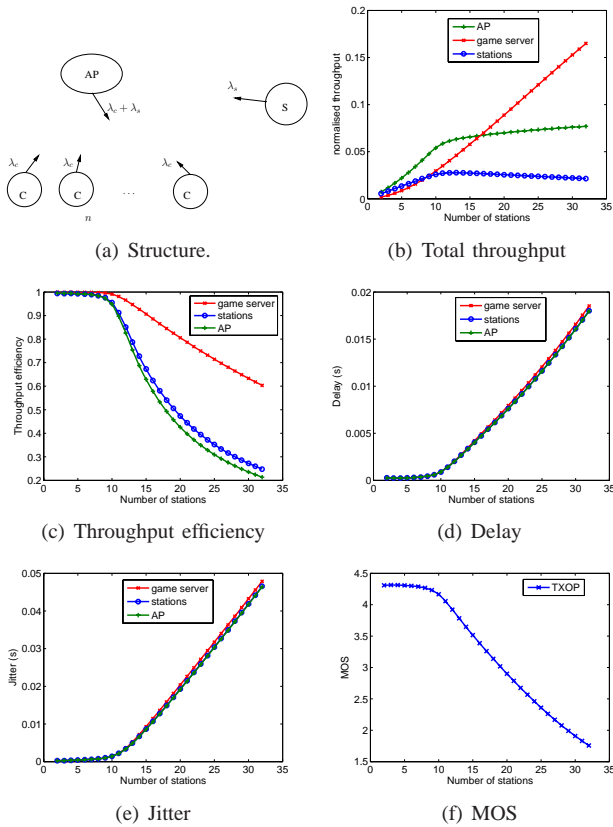
(a) Structure.

(b) Total throughput

(c) Throughput efficiency

(d) Delay

(e) Jitter

(f) MOS

Fig. 8. AP and Server priority with TXOP

Fig. 9. MOS compare of schemes

gains can also be made when the game server is located in a wired network beyond the AP. Using the model, we can also consider the advantages of 802.11e's DLS or the use of multicast for distributing game state in a WLAN. We also plan to implement the TXOP based scheme in our testbed to demonstrate the gains beyond the assumptions of our model. We have also investigated mechanisms to protect game traffic from competing TCP traffic.

REFERENCES

[1] N. Shah and C. Haigh, "The video game industry, an industry analysis, from a vc perspective," 2005.
[2] Enterrainment Software Association, "2005 sales, damographics and usage data - essential facts about the computer and vedeo game industry," 2005.
[3] S. Gargolinski, C. St. Pierre, and M. Claypool, "Game server selection for multiple players," in *NetGames '05*, New York, NY, USA, 2005, ACM.
[4] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, Mar 2000.
[5] K. Duffy, D. Malone, and D.J. Leith, "Modeling the 802.11 distributed coordination function in non-saturated conditions," *Communications Letters, IEEE*, vol. 9, no. 8, pp. 715–717, Aug 2005.
[6] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *Networking, IEEE/ACM Transactions on*, vol. 15, no. 1, pp. 159–172, Feb. 2007.
[7] M. Dick, O. Wellnitz, and L. Wolf, "Analysis of factors affecting players' performance and perception in multiplayer games," in *NetGames '05*, NY, USA, 2005, ACM.
[8] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, 2006.
[9] G. Armitage, M. Claypool, and P. Branch, *Networking and online games: understanding and engineering multiplayer Internet games*, John Wiley & Sons, Ltd., Chichester, England, 2006.
[10] P. Clifford, K. Duffy, D. Leith, and D. Malone, "On improving voice capacity in 802.11 infrastructure networks," in *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, 2005.
[11] "Tcpdump," Tech. Rep., http://www.tcpdump.org, 2006.
[12] A. Cricenti and P. Branch, "ARMA(1,1) modeling of Quake4 server to client game traffic," in *NetGames '07*, NY, USA, 2007, ACM.
[13] W. Feng, F. Chang, W. Feng, and J. Walpole, "A traffic characterization of popular on-line games," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 488–500, 2005.
[14] C. Ibrahim, E. Melike, and O. F. Sema, "Analysis of multi-player online game traffic based on self-similarity," in *NetGames '06*, New York, NY, USA, 2006, ACM.
[15] Michael S. Borella, "Source models of network game traffic," *Computer Communications*, vol. 23, pp. 403–410, 2000.
[16] P. Branch and G. Armitage, "Extrapolating server to client IP traffic from empirical measurements of first person shooter games," in *NetGames '06*, NY, USA, 2006, ACM.
[17] Smart Internet Technology CRC, "Simulating online networked games (SONG) database," *http://caia.swin.edu.au/sitcrc/song/download_traces.html*.
[18] P. Chatzimisios, A.C. Boucouvalas, and V. Vitsas, "Packet delay analysis of IEEE 802.11 MAC protocol," *Electronics Letters*, vol. 39, no. 18, pp. 1358–1359, Sept. 2003.
[19] C. Schaefer, T. Enderes, H. Ritter, and M. Zitterbart, "Subjective quality assessment for multiplayer real-time games," in *NetGames '02*, NY, USA, 2002, ACM.
[20] B. Carrig, D. Denieffe, and J. Murphy, "Supporting first person shooter games in wireless local area networks," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, Sept. 2007, pp. 1–6.
[21] A. F. Wattimena, R. E. Kooij, J. M. van Vugt, and O. K. Ahmed, "Predicting the perceived quality of a first person shooter: the Quake IV g-model," in *NetGames '06*, NY, USA, 2006, ACM.
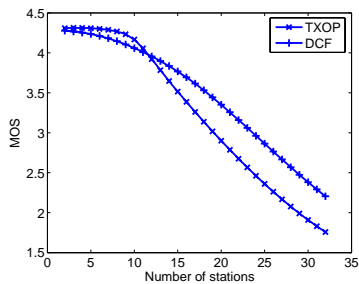
real-time games (Quake 4 in this instance). We used traffic characteristics measured from a wireless games testbed to establish traffic parameters and combined this with MAC and MOS modeling. We then demonstrated how the distribution of transmission opportunities given by the MAC does not match the traffic load for a multiplayer game in a WLAN. Based on our model, we see that the 802.11e parameter TXOP can be used to prioritize AP and game server, and improve game performance. The game wireless network capacity can improve from 10 players with normal DCF to 15 players with TXOP prioritised in the AP and the game server.

We have begun to consider network topologies beyond the simple WLAN described here. Our initial results indicate that