Do Nintendo handhelds play nice? An analysis of its wireless behavior*

Adam C. Lusch, Adele V. Fleury and Surendar Chandra
University of Notre Dame, Notre Dame, IN 46556, USA
{alusch1,afleury,surendar}@nd.edu

ABSTRACT

Nintendo DS[1] handheld game units are popular. They are capable of wireless game play using IEEE 802.11 networks. In this paper, we investigated the local Nintendo wireless game traffic and the effect of this game traffic on other wireless users. We analyzed the wireless traffic generated by up to six players for a number of different games. Our analysis showed that the local wireless traffic differed significantly from wide-area wireless traffic for the same games and hardware. The bit rates used made the small game packets appear large to higher speed networks that shared the wireless channel. Also, Nintendo used a point coordination function (PCF) to arbitrate channel access for local games. Ideally, the contention free slots provided by PCF can provide better quality of service for the game traffic. However, this added extra overhead for the coordination traffic, about 98% of polls for Pictochat produced no game data. Also, this PCF coordination packets were unaware of the coexistence of distributed coordination function (DCF) traffic from typical wireless access points; catastrophically interfering with the wireless traffic of other wireless LAN users.

Keywords

Nintendo DS, IEEE 802.11 wireless, Point coordination function (PCF), Distributed coordination function (DCF)

1. INTRODUCTION

Nintendo DS Lite handheld game units are inexpensive and popular, selling 508K units in the USA alone in March 2007 [11]. Their wireless capabilities allow them to play games with users over the Internet using the free Nintendo Wi-Fi connection service. Users can also wirelessly play games with up to fifteen other players in the immediate vicinity (up to 100 ft) without connecting through an access point. Local players can even download the games from a single host without requiring all the players to own a game card. It is relatively easy to spontaneously play a game with other Nintendo users who are in the vicinity; in a cafe, airport, school or at work without the need to get special permission from the local site. Nintendo has sold over forty millions game units so far; such an ad hoc collaboration is increasingly likely to find other Nintendo players in many public spaces.

Nintendo DS uses the IEEE 802.11 [9] wireless LAN protocol and operates in the 2.4 GHz ISM band. This band is also used by various wireless LANs (IEEE 802.11 b/g/n) as well by personal area networks (IEEE 802.15) and is congested [6]. This paper aims to understand the effects of Nintendo's local wireless traffic on other users who might be sharing the same wireless medium.

1.1 Comparison with prior work by Claypool

Earlier, Claypool [3] had analyzed the wireless Nintendo DS game traffic. They showed that games consumed a constant amount of data (around 200 kbps). These game units frequently transmitted packets: within a few milliseconds of each other. The packets were also small: around 100 bytes each. These durations were smaller than what was observed by Claypool [2] in wired gaming scenarios, where they measured durations of around 100 ms. Their preliminary analysis also showed that the interference of the game traffic on 802.11 LAN performance was minimal. In this work, we extend their analysis and investigate the reasons behind the decreased inter-packet durations as compared to wired gaming scenarios.

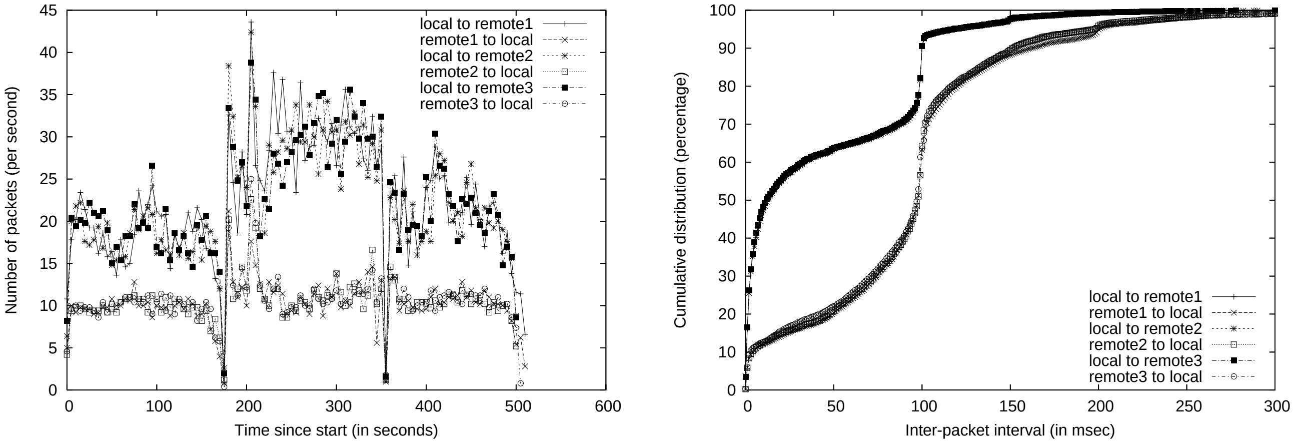
Ideally, we would like to analyze this discrepancy by investigating the wired and wireless behavior for the same game and gaming platform. Unfortunately, the DS does not support wired game play. Hence, we first analyzed the packet dynamics for playing games with other remote users using the Nintendo WiFi connect service. We presume that the game data requirements (unlike other control traffic) are identical between playing a game over the Internet and for playing the game with local players. We based our presumption on the observation that the gaming experience itself was identical. For our experiments, we played the Mario Kart DS game with three other remote players over the Internet. We also hired four students to play Mario Kart using the local play feature. Detailed experiment setup will be described in Section 2.

Each game session lasted for about three minutes and we reported the behavior for three consecutive games. We plotted the number of packets that were sent between the game unit and the various remote players as well as the cumulative distribution of inter-packet intervals in Figure 1. Each of the data packets were unicast between each players and were of a constant size of 60 bytes each. From Figure 1(a), we note that the local game unit transmitted more network packets to the remote site than what was actually received from the remote site. The packet rates from remote nodes appear to be restricted by packet rates supported by the Internet. Even though the number of received packets were

*Supported in part by the U.S. National Science Foundation (CNS-0447671)

[1] DS Lite is a newer version of Nintendo DS. We use the terms interchangeably. We used DS Lite for our experiments

NetGames'07, September 19-20, 2007, Melbourne, Australia
.



(a) Number of packets sent between the game unit and the various remote players. Each data packet was 60 bytes long. The data represents three different gaming runs (of about 3 minutes each)

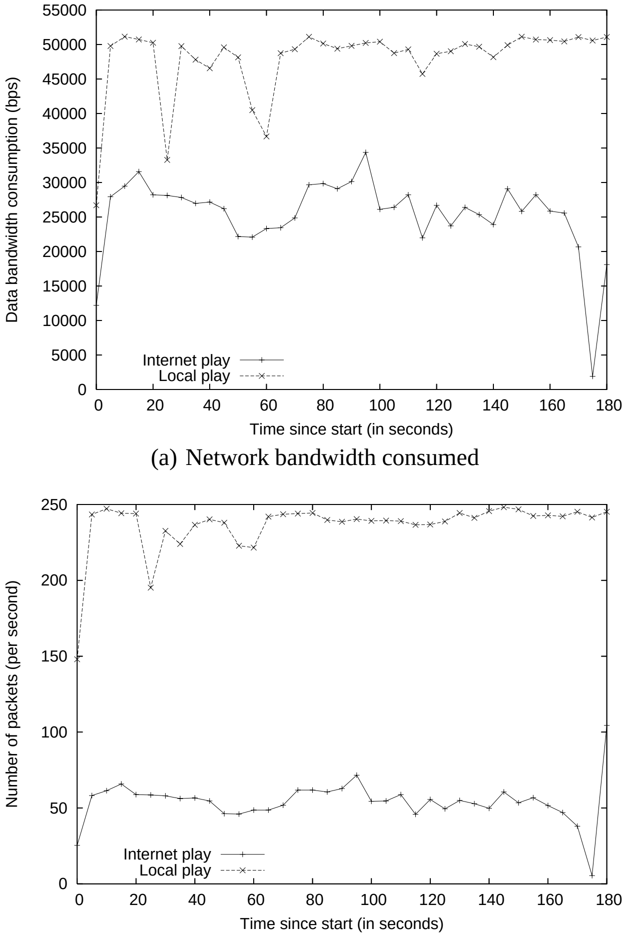
(b) *Packet intervals between the game unit and the various remote players. Packets were unicast to each individual player*

Figure 1: Packet dynamics for playing Mario Kart with three other players over the Internet

smaller than the number that was sent out, the gaming experience itself was adequate with no noticeable discrepancy. From Figure 1(b), we observed that the inter-packet durations for Internet players were higher than what was observed by Claypool [3]. Claypool observed a new packet every millisecond. However, for the Internet game play scenario, 40% of the packets exhibited inter-packet durations of less than 100 msec, about 35% using a duration of 100 msec with 25% exhibiting even higher durations.

Next, we performed local game play experiments that matched our Internet scenario. We played Mario Kart DS among four local players (Claypool also analyzed Mario Kart) and plotted the bandwidth consumed as well as the number of packets transmitted by comparing the local game play and the Internet game play in Figure 2. If our presumption that the network dynamics of the local and Internet play are the same were to be true (albeit with differences in packet delays introduced by the Internet), we expected the bandwidth consumption and the number of packets transmitted to be identical. However, we noticed that the local games used more network bandwidth (50 kbps vs 30 kbps) and used more packets (250 packets per second vs 50 packets per second) than remote games.

Finally, we also investigated whether the local wireless traffic interfered with other LAN users. Note, that the DS's were transparent to established wireless users and access points because the ESSID was not transmitted with the beacons. Any interference would be felt even though the access points may not be able to coordinate their wireless usage with the Nintendo DS. For this experiment, one user played the Nintendo Pictochat game (described in Section 2) between three game units. Unlike other Nintendo games which randomly chose the wireless channels to play their games, Pictochat was predictable in its choice of wireless channels. As was also observed by Noland [10], Chat rooms A, B, C and D chose wireless channels 1, 7, 13 and 7, respectively. We played the game on chat room A (wireless channel 1 was exclusively allocated for research purposes in our department) and simultaneously setup an access point in channel 1 and measured the TCP and UDP throughput between a wireless laptop and a wired server. We used the iPerf tool ([12]) for our measurements and plotted the measured bandwidth for TCP and UDP tests in Figure 3. We observed that the Nintendo DS had negligible effect on UDP traffic. However, the interference was dramatic for TCP traffic. The measured throughput dropped



(a) Network bandwidth consumed

(b) Number of packets

Figure 2: Comparing the behavior of local game play and Internet game play that used the Nintendo Wi-Fi connect service

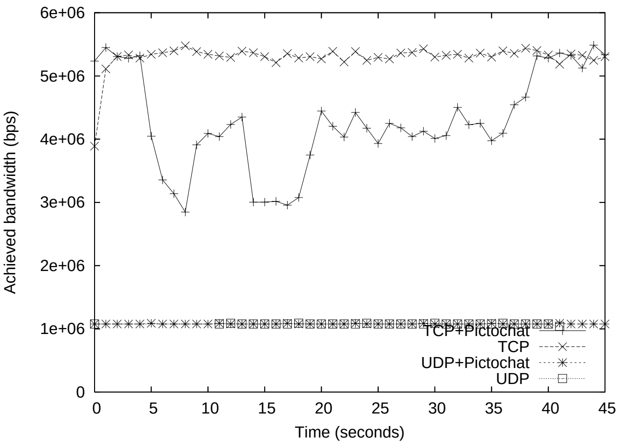


Figure 3: Pictochat interfering WLAN network (measured using iperf). The drop in bandwidth is drastic for TCP connections with no observable change for UDP connections

from over 5 Mbps to between 2 and 3 Mbps (even though the volume of game data itself was not high). As the users left the game, the measured bandwidth gradually recovered. Claypool reported no interference between a *http* flow and Super Mario game. It is likely that their two flows were operating under different, non-interfering wireless channels.

Clearly, there was a discrepancy between the Internet scenario and the local game play scenario. Game play with other Internet users appear more similar to wired scenarios [8, 13] than they are to the network characteristics of local game play mode. Local play transmitted many more packets at close intervals. Even for simple games such as Pictochat which essentially broadcasted a bitmap to other gamers in the vicinity, the local wireless traffic can drastically reduce the network bandwidth achievable by competing TCP flows that operate using regular access points. This paper investigates the reasons for such discrepancy and the effects of such a discrepancy on other wireless users.

2. EXPERIMENTAL SETUP

We used six Nintendo DS Lite game units for our experiments. We collected traces with up to six players for various configurations: two, three, four, five, five active and one inactive and six players. We collected game traffic for playing Pictochat, Brain Age and Mario Kart DS. We used undergraduate students who were compensated (by the hour) for playing these games. The students were proficient in playing video games in general even though many had not actually played these particular games. Pictochat sends a small bitmap painted by a player to other players. Even though the gamers did not know about PictoMation animation tricks, the players found the game to be addictive. Brain age tests the speed in which the gamers solved simple math operations (such as addition, multiplication etc.) and type in their solutions. By comparison, Mario Kart is a fairly complex racing game.

Even though the Nintendo DS used IEEE 802.11 wireless networks, their behavior was slightly different than that of regular wireless clients. The game hardware itself did not provide mechanisms to configure the SSID and other wireless parameters for accessing the Nintendo WiFi connection. It was the responsibility of the individual game to choose these configuration parameters. For local game play, there were no explicit configuration parameters to choose the wireless channel (as described earlier, Pictochat used certain predetermined channels for each chat room). The games do not broadcast their SSIDs making them invisible to normal clients. For our experiments, we captured the wireless traffic as follows: first, we used a Kismac client running on a Mac Powerbook and using a Linksys card to identify the channel used by the particular game session. Our department had dedicated channel 1 for research. However, even though channel 1 was available, the DS chose the wireless channels in a random fashion. After the current operating channel was identified, we used another laptop running Redhat Linux that used a Linksys card to capture wireless traffic. We configured the card to operate in monitor mode to capture all the network packets sent during the particular gaming session.

3. ANALYZING THE LOCAL WIRELESS TRAFFIC

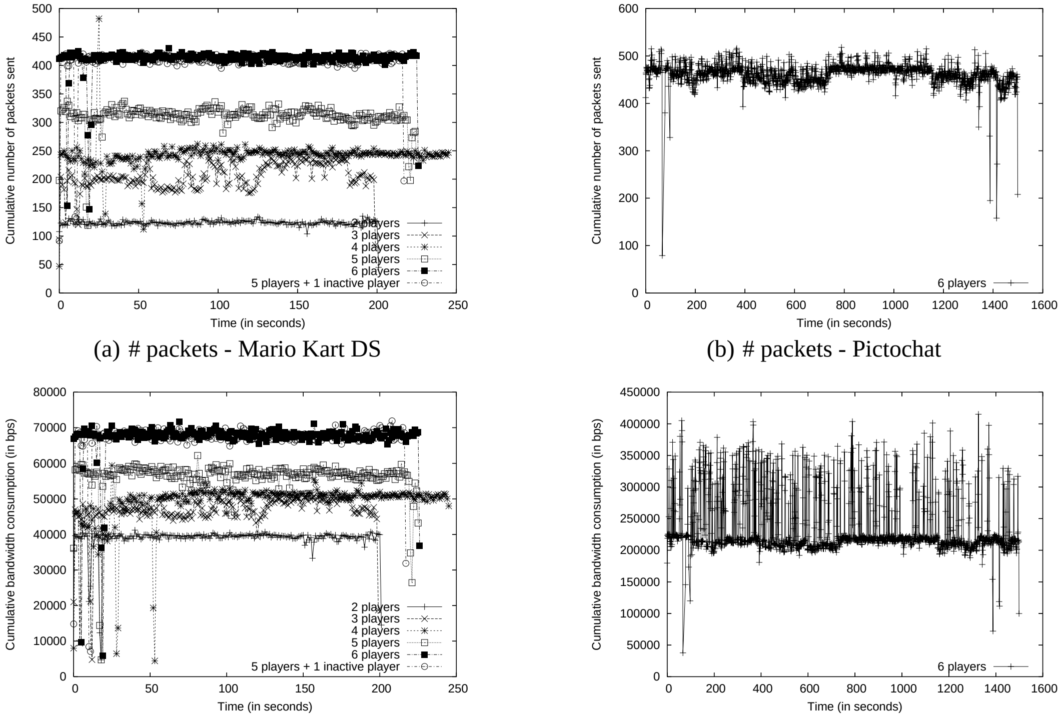
We analyzed the captured data to understand the reason for the discrepancy in the network packet dynamics for local game play and their implications for users who share the wireless spectrum.

3.1 Observation 1: Scalability of game traffic - packets scaled with more players

First, we analyzed the effect of increasing the number of players on the traffic generated by the game units. Many games allow for up to 16 players per game. Pictochat allowed for 16 players per room for a total of 64 players in four rooms. We plot the total number of packets sent and the bandwidth consumed by these packets in Figure 4. From Figure 4(a), we noted that the number of packets were proportional to number of players. The bandwidth consumed for the Mario Kart (Figure 4(c)) did not scale with the number of players. For the Pictochat game with six players, the number of packets (Figure 4(b)) were high: around 500 packets per second. The bandwidth consumed also varied between 200 kbps and 400 kbps (Figure 4(d)). These observations were similar to the the packet dynamics observed by Claypool [3]. These figures were puzzling because Pictochat is a fairly simple game: the players painted a bitmap which was transmitted to all other players whenever the player hit the *Send* button. Given the rates at which the players drew items on the screen (on the order of tens of seconds), it was puzzling that the game would create so many packets.

3.2 Observation 2: Packets were deceptively small

Like Claypool [3], we also observed (Figure 5) that most game packets were small. However, a closer analysis showed that the broadcast packets observed by Claypool [3] were wireless Beacons and not data packets. The game unit which initiated a particular game acted as the access point and transmitted Beacons describing the available games to other Nintendo clients. The Beacons advertised that the game unit supported low data rate 1 and 2 Mbps modes of operation. Wireless LAN protocols such as IEEE 802.11b chose different encoding for achieving maximum throughput of 1, 2, 5.5 or 11 Mbps based on the current operating conditions. Transmissions at each of the 1, 2, 5.5 and 11 Mbps modes were encoded at 1, 2, 4 and 8 bits per carrier symbol. 1 Mbps and 2 Mbps modes operated at 1 MSps (symbols per second) while high bitrate 5.5 Mbps and 11 Mbps modes operated using Complementary Code Keying (CCK) at 1.375 MSps. For backward compatibility, the link-layer headers were always transmitted at 1 Mbps. Similarly, 802.11g networks used protection bits to co-exist with slower 802.11 networks. For example, transmitting a 1 byte data at 11 Mbps would require transmitting 24 bytes of headers at 1 Mbps and transmitting the 1 data byte+10 frame headers at 11 Mbps. The



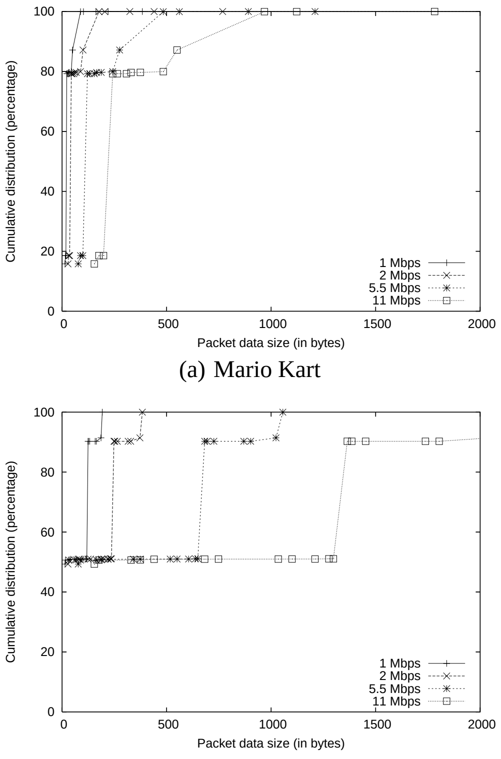
(a) # packets - Mario Kart DS

(b) # packets - Pictochat

(c) BW consumed - Mario Kart DS

(d) BW consumed - Pictochat

Figure 4: Scalability of game traffic (Brain age behaved similar to Pictochat. Not illustrated for lack of space)



(a) Mario Kart

(b) Pictochat

Figure 5: Cumulative distribution of wireless packet sizes (scaled from Nintendo operating in 1 Mbps mode to the various IEEE 802.11b modes (1, 2, 5.5 and 11 Mbps))

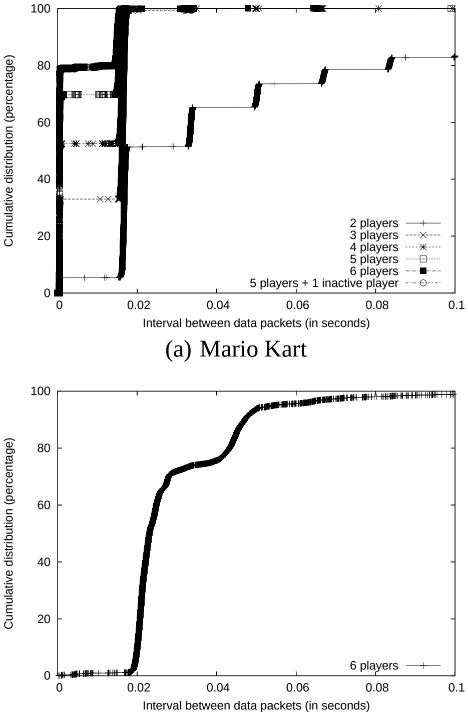
effective throughput is far smaller for small packet payloads [6].

Assuming that the Nintendo DS's were operating in the 1 Mbps mode (beacons advertised their capability to operate in 1 Mbps or 2 Mbps), we calculated the equivalent packet size for the observed game packets on an 802.11b client operating at 1 Mbps, 2 Mbps, 5.5 Mbps and 11 Mbps (taking into consideration the 24 byte frame header operating at 1 Mbps and 10 byte header operating at the higher rates). We plotted a cumulative distribution of the various sizes in Figure 5 for playing Mario Kart and Pictochat among six players. These results show that though the data packets appear small, their effect on IEEE 802.11b networks can be fairly large. For example, Pictochat would appear to use 1,250 byte packets (rather than 100 byte packets) for 50% of the time using a 11 Mbps 802.11b network. The results are even worse for considering 54 Mbps IEEE 802.11g networks that used OFDM encodings. The Nintendo game traffic are not small in practice.

3.3 Observation 3: DS uses Point Coordination Function

We observed that the DS used the 802.11 Point Coordination Function (PCF) mode for contention free access of the wireless channel. The hosting node controlled the time slots during which the other players can transmit their data. PCF is optional within the IEEE 802.11 specification and few access points support this mode of operation. Though PCF can allow for QoS access to the wireless medium, the standard itself does not specify the mechanism for allocating access rates and hence PCF was never widely deployed. Even its successor IEEE 802.11e failed to become popular.

The Nintendo DS wireless protocol also used an abnormal (though still complaint) variation that allowed for MAC layer broadcast of game data to all other players without being resent from the coordination host. The unit that started the local game chose the



(a) Mario Kart

(b) Pictochat

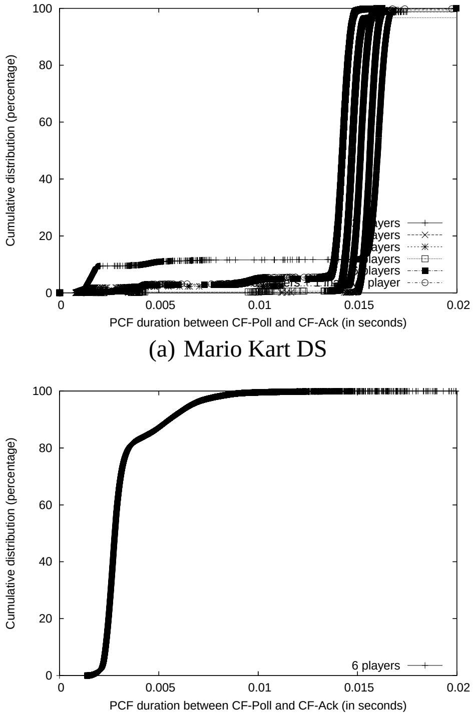
Figure 6: Cumulative duration between game packets (does not include the CF-Poll and CF-Ack from the coordinator station)

wireless channel, transmitted Beacons and acted as the PCF coordinator (PC). The PC sent a CF-Poll frame to the MAC address 03:09:bf:00:00:00. Game units that had data sent a CF-Ack frame to 03:09:bf:00:00:10 while the PC terminated the contention-free interval with a CF-Ack frame to 03:09:bf:00:00:03. The MAC addresses were fictitious and presumably treated as broadcast frames by the game units. The CF-Poll and CF-Ack from the PC itself were overhead transmissions.

Next, we plotted the cumulative distribution of the interval between game data transmissions (without including the CF-Poll and CF-Ack from the coordinator) in Figure 6. We also plotted the cumulative distribution of duration between a CF-Poll and CF-Ack from the PC as well as the duration between CF-Ack and the subsequent CF-Poll (duration allowed for contention traffic similar to DCF) in Figures 7 and 8, respectively. We observed that these durations were small; the Nintendo did not allow for other traffic between its contention free intervals, even though no other client can belong to this particular basic service set (BSS). Though the standard does not offer any guidance on the choice of these parameters, the practical implications are catastrophic for competing traffic.

Finally, we plotted the distribution of the number of data packets sent during the content free window in Figure 9. The PC itself can piggy-back its game data with a CF-Poll or CF-Ack. We noted that for Mario Kart, many game units did transmit data during this interval (about 90% for six player game). The number were similar whether the sixth player was active or inactive. For Pictochat, we observed that very little data was actually transmitted. In over 95% of the contention free intervals no game data was sent: most of the observed traffic can be attributed to PCF control traffic.

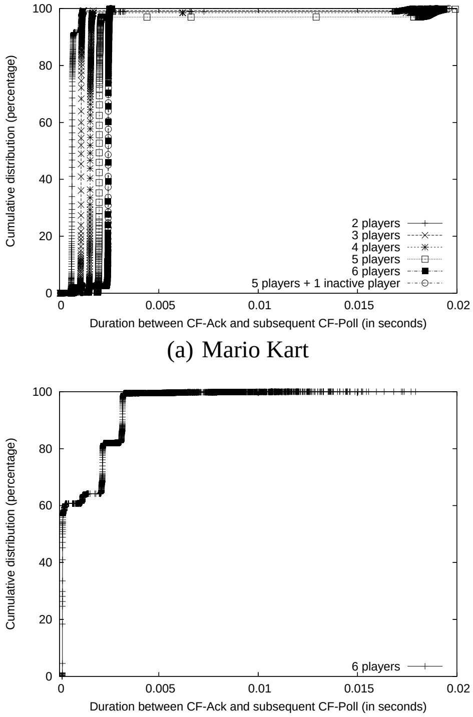
4. RELATED WORK



(a) Mario Kart DS

(b) Pictochat

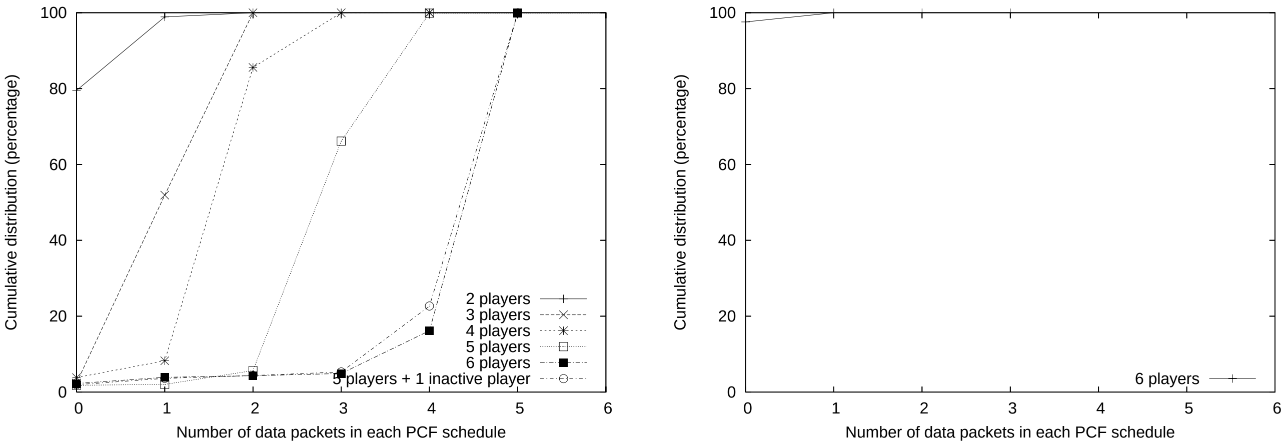
Figure 7: Duration between CF-Poll and CF-Ack from the coordinator station for varying game size



(a) Mario Kart

(b) Pictochat

Figure 8: Duration between CF-Ack and the subsequent CF-Poll from the coordinator station for varying game sizes



(a) Mario Kart

(b) Pictochat (98% of CF-Poll:CF-Ack elicited no game data from other players

Figure 9: Number of messages sent from the game stations during a CF-Poll from the coordinator station

As described in Section 1.1, Claypool [3] had already analyzed the traffic from Nintendo DS. In this work, we extend their analysis to show the implications for wireless medium. There has been a number of prior efforts [7, 1] on quantifying the game dynamics for wired gaming scenarios. Our work shows that the wireless behavior can be more complex. Heusse et. al. [4] showed the anomaly of low rate 802.11 clients on the overall throughput of DCF based 802.11b networks. A DS using Nintendo WiFi service and operating at low rates likely brings down the achieved throughput for the entire wireless network. Howitt et. al. [5] showed the interference of Bluetooth networks and 802.11 LANs. Our work highlighted similar interference between PCF game traffic and DCF networks.

5. DISCUSSION: ARE DS'S MISBEHAVING?

Compared to general network flows, game traffic were known to be small and frequent. PCF was arguably the preferred mechanism to allocate the network channel for such traffic. PCF was known to give higher priority to contention free traffic. However, the confluence of chosen bit-rates and PCF based BSS co-existing with wireless LANs creates unintended interaction between PCF and DCF based system. Arguably, these problems will not occur had the DS chosen an exclusive channel. However, it is likely for users to take a DS into an established hot-spot and their game traffic will cause more damage than the amount of traffic created should warrant. Further investigation on the interaction of overlapping PCF and DCF systems are warranted among the networking community.

6. REFERENCES

[1] CEVIZCI, I., EROL, M., AND OKTUG, S. F. Analysis of multi-player online game traffic based on self-similarity. In *Netgames '05* (Oct. 2006), p. 25.

[2] CLAYPOOL, M. The effect of latency on user performance in real-time strategy games. *Comput. Networks 49*, 1 (2005), 52–70.

[3] CLAYPOOL, M. On the 802.11 turbulence of nintendo ds and sony psp hand-held network games. In *Netgames '05* (Oct. 2005), pp. 1–9.

[4] HEUSSE, M., BERGER-SABBATEL, F. R. G., AND DUDA, A. Performance anomaly of 802.11b. In *INFOCOM* (Mar. 2003), vol. 2, pp. 836–843.

[5] HOWITT, I. Bluetooth performance in the presence of 802.11b wlan. *IEEE Transactions on Vehicular Technology 51*, 6 (Nov. 2002), 1640–1651.

[6] JARDOSH, A. P., RAMACHANDRAN, K. N., ALMEROTH, K. C., AND BELDING-ROYER, E. M. Understanding Congestion in IEEE 802.11b Wireless Networks. In *Proceedings of ACM Internet Measurement Conference* (Berkeley, CA, Oct. 2005), pp. 279–292.

[7] KIM, J., CHOI, J., CHANG, D., KWON, T., CHOI, Y., AND YUK, E. Traffic characteristics of a massively multi-player online role playing game. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games (NetGames '05)* (2005), pp. 1–8.

[8] LANG, T., BRANCH, P., AND ARMITAGE, G. A synthetic traffic model for quake3. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '04)* (2004), pp. 233–238.

[9] LAN/MAN STANDARDS COMMITTEE OF THE IEEE COMPUTER SOCIETY. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE, 3 Park Avenue, New York, NY 10016, 1999.

[10] NOLAND, M. A. Inferring a proprietary wireless l3 protocol from packet traces. Tech. rep., UNC Chapel Hill, `http://www.cs.unc.edu/~jasleen/Courses/Fall05/projects/wmb_presentation.pdf`, Dec. 2005.

[11] PUGET SOUND BUSINESS JOURNAL. Nintendo game sales stay in front. `http://seattle.bizjournals.com/seattle/stories/2007/04/16/daily37.html`, Apr. 2007.

[12] TIRUMALA, A., QIN, F., DUGAN, J., FERGUSON, J., AND GIBBS, K. *iPerf: TCP/UDP bandwidth measurement tool*, 2.0.2 ed. NLANR, May 2005. `http://dast.nlanr.net/Projects/Iperf/`.

[13] ZANDER, S., AND ARMITAGE, G. A traffic model for the xbox game halo 2. In *Proceedings of the international workshop on Network and operating systems support for digital audio and video (NOSSDAV '05)* (2005), pp. 13–18.