On the 802.11 Turbulence of Nintendo DS and Sony PSP Hand-held Network Games

Mark Claypool
CS Department, Worcester Polytechnic Institute
100 Institute Road, Worcester, MA, 01609, USA
claypool@cs.wpi.edu

ABSTRACT

The growth in computer games and wireless networks has catalyzed the production of a new generation of hand-held game consoles that support multi-player gaming over IEEE 802.11 networks. Understanding the traffic characteristics of network games running on these new hand-helds is important for building traffic models and adequately planning wireless network infrastructures to meet future demand. This paper examines the traffic characteristics of IEEE 802.11 network games on the Nintendo DS and the Sony PSP. Analysis of a variety of games from several different genres shows that despite some overall similarities, most of the games have significantly different network characteristics. In addition, the games and hand-held platforms differ in their ability to handle degraded wireless network conditions and in the amount of broadcast traffic sent. The results should be a useful beginning to building effective traffic models for this new generation of network games.

Categories and Subject Descriptors: C.2.m [Computer-Communication Networks]: Miscellaneous

General Terms: Measurement, Performance, Experimentation

Keywords: Nintendo DS, Sony PSP, Wireless, IEEE 802.11, Network Games

1. INTRODUCTION

The computer game industry has seen tremendous growth in recent years, with the end of 2004 seeing gross revenue from U.S. computer game sales at $7.3 billion,[1] on par with domestic box office ticket sales.[2] Networked games have also seen considerable growth, spurred on by the growth in residential broadband Internet connections with high capacities and low latencies that have encouraged more and more game developers to incorporate multi-player, networked features into their products.

Through 2004, hand-held gaming consoles were primarily single player, with any multi-player networking done with a short wire between hand-helds or maybe by Bluetooth. Then, in the Spring of 2005, the next generation of hand-held gaming systems equipped with wireless networking for multi-player gameplay were released. Sony's *Playstation Portable* (*PSP*), and Nintendo's *Dual-Screen (*DS*)* support the IEEE 802.11b WLAN standard, allowing ad hoc networking between hand-helds. The 802.11 support for hand-held game players is significant since it is increasingly common for home users to access the Internet via a wireless Local Area Network (WLAN) connected to a single external broadband connection. Furthermore, growth in WLAN deployment at universities [7] and even in entire towns [5] increases the likelihood that concurrent wireless hosts compete on the same wireless channels.

The traffic generated by one host on a WLAN can have dramatic impact on the performance of other hosts on the WLAN [1, 8]. Specifically, when there is a WLAN host with weak wireless connectivity, and hence low WLAN capacity, the performance of all WLAN hosts is severely degraded. These results are especially important since the quality of wireless links for hand-helds is not guaranteed and some hand-helds are specifically designed to operate at low WLAN capacities to conserve power.

To adequately plan WLAN network infrastructures, it is important for engineers to have knowledge of the network load caused by hand-held game traffic. For wired network games, traffic from a number of popular games has been characterized to provide suitable traffic models for testing existing or planned network designs. There have been numerous studies of traffic models for popular PC games [2, 3, 6, 10, 12] and even game consoles [13]. However, to the best of our knowledge, networking characterization for hand-held games running over 802.11 networks has not been studied.

Using vicinity sniffing to capture 802.11 traffic, this paper investigates the network characteristics (the size and frequency of data sent and the overall bitrate), which we call *turbulence*,[3] for a variety of network games on the Nintendo DS and the Sony PSP.

This study seeks to answer the following questions (with

[1] 2005 Sales, Demographics and Usage Data, Entertainment Software Association, http://www.theesa.com/

[2] Total U.S. Box Office Grosses, National Organization of Theatre Owners, http://www.natoonline.org/-statisticsboxoffice.htm

[3] The term "footprint" is often used in systems work in the context of the basic size a piece of memory of some software. In a network, the size and distribution of packets over time is important, hence our word "turbulence", as first used in [11].

NetGames'05, October 10–11, 2005, Hawthorne, New York, USA.

Figure 1: Nintendo DS

Figure 2: Sony PSP

a brief answer as revealed by this study provided in parentheses):

1. *What is the network turbulence for hand-held network games?* Characterizing the frame sizes and frame inter arrival times for hand-held game consoles is a critical first step for building accurate game traffic models. (Answer: Hand-held network games make frequent sends of small frames of data, typical of network games on other platforms.)
2. *Does the network turbulence for different hand-helds (such as the PSP and the DS) differ from each other?* If the answer is no, then research efforts can study traffic on one hand-held only, saving hardware costs. (Answer: The characteristics of Nintendo DS game traffic is different than that of the Sony PSP game traffic.)
3. *Does the network turbulence for different games (such as* Ridge Racer *and* Super Mario *on the same hand-held differ from each other?* If the answer is no, then it should be possible to develop one traffic model that applies to all games on that hand-held, thus saving in model development time. (Answer: Games on the Nintendo DS have network characteristics fairly similar to each other, while games on the Sony PSP vary considerably from game to game.)
4. *Does the network turbulence for hand-held games differ from PC games?* If the answer is no, then mature, previously developed models for PC games can be used to analyze the WLAN impact of hand-held consoles. (Answer: The hand-helds send game data in sizes comparable to that of PC or console games, but hand-helds send data more frequently.)
5. *Does hand-held game traffic interfere with traditional Internet traffic on the same wireless channel?* If the answer is yes, then deployment and development of IEEE 802.11 networks must plan to accommodate the impending increase in hand-held gaming traffic. (Answer: In some cases, hand-held games can have adverse affects on the throughput for applications sharing the same WLAN channel.)

The rest of this paper is organized as follows: Section 2 provides background on the Nintendo DS and Sony PSP; Section 3 describes our measurement setup and methodology; Section 4 analyzes our results in relation to the questions posed above; Section 5 summarizes our conclusions and Section 6 presents possible future work.

2. NINTENDO DS AND SONY PSP

The Nintendo DS, shown in Figure 1, has a screen resolution $256x192$ pixels with 260,000 colors. The DS has dual ARM processors, with an ARM 9 processor running at 67 MHz and an ARM 7 processor running at 33 MHz, and 4 MBytes of on-board RAM. The DS has built-in speakers and a head-phone jack. For gameplay, the DS has a touch-pad on the lower screen and includes a built-in stylus, as well as buttons that are standard on the Nintendo Game Boy hand-helds. Some DS games allow networked play with just one copy of a game, with game copies downloaded to the other players during startup.

The DS uses IEEE 802.11b for all communications with a specific subset of the features of 802.11b to save power, including using: a) a short preamble; and b) wireless capacities of 1 Mbps or 2 Mbps.[4] For two-player games, the DS sends data primarily point-to-point, only using broadcast frames when starting a game session. The DS supports ad hoc mode, allowing up to 16 players for peer-to-peer multiplayer games (although most DS games released support far fewer than 16 players). The DS does not internally support an IP stack (including UDP/TCP on top of IP), making its network data unroutable on the Internet. However, there are plans for doing TCP/IP in software and doing tunneling to enable DS multi-player games on the Internet.

The Sony PSP, shown in Figure 2, has a resolution of $480x272$ pixels with 24-bit color. The screen has an aspect ratio of 16:9, specifically chosen so that movies can be viewed in wide-screen format. The PSP has dual MIPS R4000 processors running at 333 MHz, with 32 MBytes of on-board RAM. The PSP has a USB 2.0 port, supports infrared for remote operation, includes a port for a removable memory stick, and has low-volume speakers and a head-phone jack. For gameplay, the PSP has an analog joystick operable by thumb and the buttons that are standard on the Sony Playstation consoles. Some PSP game boxes claim only one copy of a game may be required for multi-player functionality, and the devices themselves include a game sharing mode to download games. However, at the time of this paper, all multi-player networked play on the PSP requires a separate copy of the game for each player.

The Sony PSP uses IEEE 802.11b for networked communication. At the time of this paper, there is insufficient documentation to ascertain whether the PSP is fully compliant with 802.11b or if it only uses a subset of the protocol. However, the analysis in this paper suggests the wireless capaci-

[4]Full IEEE 802.11b supports both long and short preambles, and capacities of 1, 2, 5.5 and 11 Mbps.

ties used by the PSP are not limited to 1 or 2 Mbps, as in the Nintendo DS, since wireless channel capacities of 11 Mbps are typically observed. Moreover, some games on the PSP use 802.11 broadcast frames to send data even when there are only two players. The PSP supports ad hoc mode for up to 16 players for multi-player network games (although most current PSP games support far fewer than 16 players). In addition, the PSP supports 802.11 infrastructure mode, allowing it to connect to a compatible wireless access point (AP) and use IPv4 to communicate (via an AP) over the Internet.

3. METHODOLOGY

3.1 Game Selection

Name	Platform	Genre
DS	Metroid Prime Hunters	First Person Shooter
	Ridge Racer	Car Racing
	PictoChat	Online Chat
	Super Mario	Third Person Action
PSP	Need for Speed	Car Racing
	World Tour Soccer	Team Sports
	Hot Shots Golf	Individual Sports
	Untold Legends	Third Person Action

Table 1: Games Studied

Table 1 shows the variety of games chosen for study, coming from several different game genres. While the intent was to obtain a game for both the DS and PSP for each genre selected, there were no suitable equivalents (in terms of gameplay) for some of the selections at the time the measurements were done. For example, World Tour Soccer was selected for the PSP but there was no soccer game for the DS, and Metroid Prime Hunters was selected for the DS but there were no First Person Shooter (FPS) games for the PSP. PictoChat is not actually a game, but is rather a built-in instant messaging program that uses the Nintendo DS touchpad and stylus. Since chat is presumed to have intermittent interactivity (a user enters a message locally for a few seconds and then sends to others in the chat session), Hotshots Golf was selected for the PSP as a game with possibly similar interactivity. While Untold Legends is a violent dungeon crawl and Super Mario is a cartoonish adventure, they were both thought to be similar by having a third-person, 3-D perspective of an avatar interacting with the world and the other players.

3.2 Wireless Sniffing Framework

Two Sony PSPs and two Nintendo DSes, as described in 2, were used in the study. The hand-held devices were running on batteries during all gameplay and the batteries were typically over half- to nearly fully-charged. The hand-helds ran in ad hoc mode where they communicated directly with each other, without the use of an access point (AP). The 802.11 communication channel used appears to be selected by the game, but only games on channels 1, 6 and 11 were observed. Once a channel was chosen, the hand-held remained on that channel for all communication until it was turned off. The hand-helds were within close proximity (10 feet) of each other except for the studies that deliberately degraded the wireless signal strength between the hand-helds by having them move further apart.

Sniffing of the 802.11 traffic between the hand-helds was performed using a Dell Inspiron 8600 laptop with an 802.11 wireless network card, running in close proximity to both the hand-helds (or at least one in the degraded wireless study). The wireless card is a Netgear WG511 802.11b/g PCMCIA card, with Intersil's PRISM GT chipset[5]. The laptop runs Linux Fedora Core 3 with kernel version 2.6.5, using the Prism54 v1.2 driver[6] for the wireless card. The wireless sniffer uses Ethereal[7] version 0.10.3 with the wireless card in "monitor" mode, enabling the capture of management, control and data frames. The sniffer is able to capture the entire frame, including the MAC headers and the extra physical/MAC layer information provided by the Prism driver.

With vicinity sniffing, a challenge is that the sniffer cannot record all frames that are tranmitted over the network. There are three times this may happen: 1) frames are not recorded because one side of the communication is hidden from the sniffer; 2) frames are dropped because of bit-errors; and 3) frames are dropped because the sniffer is over-loaded. For 1), by having at least one of the hand-helds in close proximity to the sniffer, and often both, the likelihood of hidden communication is small. For 2), with bit-errors, the frame is dropped by the wireless network card before the sniffer sees it, and therefore likely before the hand-held destination sees it, as well. For 3) ethereal reports the number of packets missed due to being over-loaded, and for all experiments in the paper, it reported no drops.

3.3 Gameplay and Sniffing

Each game was played first in single player mode to provide some familiarity with the gameplay. Then, a two-player version of the game was played in order to do any necessary re-adjustments to the style of play. At that time, the sniffer was set to the right channel to capture all packets between the hand-helds. A full game was then started again, with multiple runs done in all cases to provide consistent data for analysis.

All gameplay (and sniffing) took place on the ground floor of a wooden house. For competing traffic, there were three 802.11 Access Points (AP) visible by the sniffer. One AP was physically located in the house itself and had a strong signal while the other two APs were quite distant and had weak signals. All three APs broadcasted beacons approximately every 100 milliseconds (the default) and communicated on 802.11 channel 6. Upon analyzing the sniffing data, it was verified that there was no additional traffic, aside from the beacons, sent to or from the APs.

4. ANALYSIS

4.1 Game Phases

After analyzing complete game traces across multiple games, it was determined that network games go through different phases, with each phase having its own frequency and duration. Our assertion is the fundamental phases are:

- *Setup* – The game parameters are selected by the users as is appropriate for the game. For example, in a race

[5]Initially Prism was owned by Intersil http://www.intersil.com, but now Prism is owned by Conexant, http://www.conexant.com/

[6]http://www.prism54.org/

[7]http://www.ethereal.com/

game, the hosting player would select the track and number of laps and each player would select a car, while in a soccer game the hosting player would pick the field and the number of minutes per half while each player would select a team. Some games have multiple setup phases, such as a car race circuit or soccer tournament, while others have one setup phase, such as a multi-level fighting game. The Setup phase typically has minimal network traffic since little communication is required between the hand-helds.

- *Synchronization* – During the transitions between Setup and when gameplay actually starts, some games communicate state and settings from one hand-held to another. For example, a map may be sent from the game host to other the hand-helds, or the sports team selections and formations chosen may be exchanged by all hand-helds. The Synchronization phase is generally marked by high bitrates with frequent and large frames in order to exchange data as fast as possible to proceed on to gameplay.

- *Play* – The game is actually played, with players responding to the game state and the hand-helds communicating player actions to the other hand-helds, as appropriate. For example, for a fighting game, the directional movement and attack moves may be communicated while a golf game may communicate the ball direction and velocity. The Play phase generally has moderate bitrates with frequent but small frames since low delay between updates is required for real-time interaction, but often there is not a lot of data to exchange.

- *Transition* – In between periods of game play, some games have transition phases where game information is loaded and processed locally from the game disk[8] into memory. For example, in a dungeon crawl game, the underground map may be loaded and the location of the monsters and treasure determined, while in a sports game, the attributes of each player could be loaded and processed. The Transition phase is generally marked by low network bitrates because most data is processed locally from the game disk and not sent over the network.

Figure 3 depicts an example of the game phases for the Sony PSP game Untold Legends. During the Setup phase, players load characters from previous games or create new characters and the hosting player decides whether to continue the story or re-start the story. During the Synchronization phase, game state information, such as the quests that have been completed and magic items found, are communicated among the other players. During the Transition phases, the players have decided to move between major locations in the world (such as out of a town into the wilderness or from the wilderness into a cave) and local game data is loaded and processed from the PSP UMD disk into memory. During the Play phases, the players control their characters through exploration, combat and inventory management.

[8]The Sony PSP has a UMD disk and the Nintendo DS has a Game Card disk

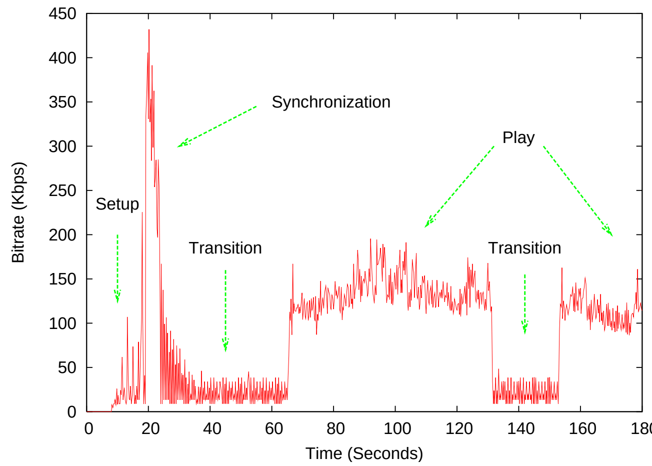


Figure 3: Example of Network Traffic for Several Game Phases (the game is Untold Legends)

In general, the length and frequency of each phase depends upon the game and often on the users actions and preferences. In this study, insufficient data was gathered to accurately determine the length of each phase since this requires a large user sample. In particular, the length of most of the phases depends upon user preferences (for example, the length of the half chosen in World Tour Soccer) and even for choices made during the game (for example, how often a player returns to town for healing in Untold Legends). Instead, the focus of our analysis is on network turbulence during the Play phases as the Play phase does not vary much within the same game and is typically the phase of most concern to users. All subsequent analysis is on a 50 second slice of a Play phase for each game in order to provide a uniform sample with which to compare the games and platforms.

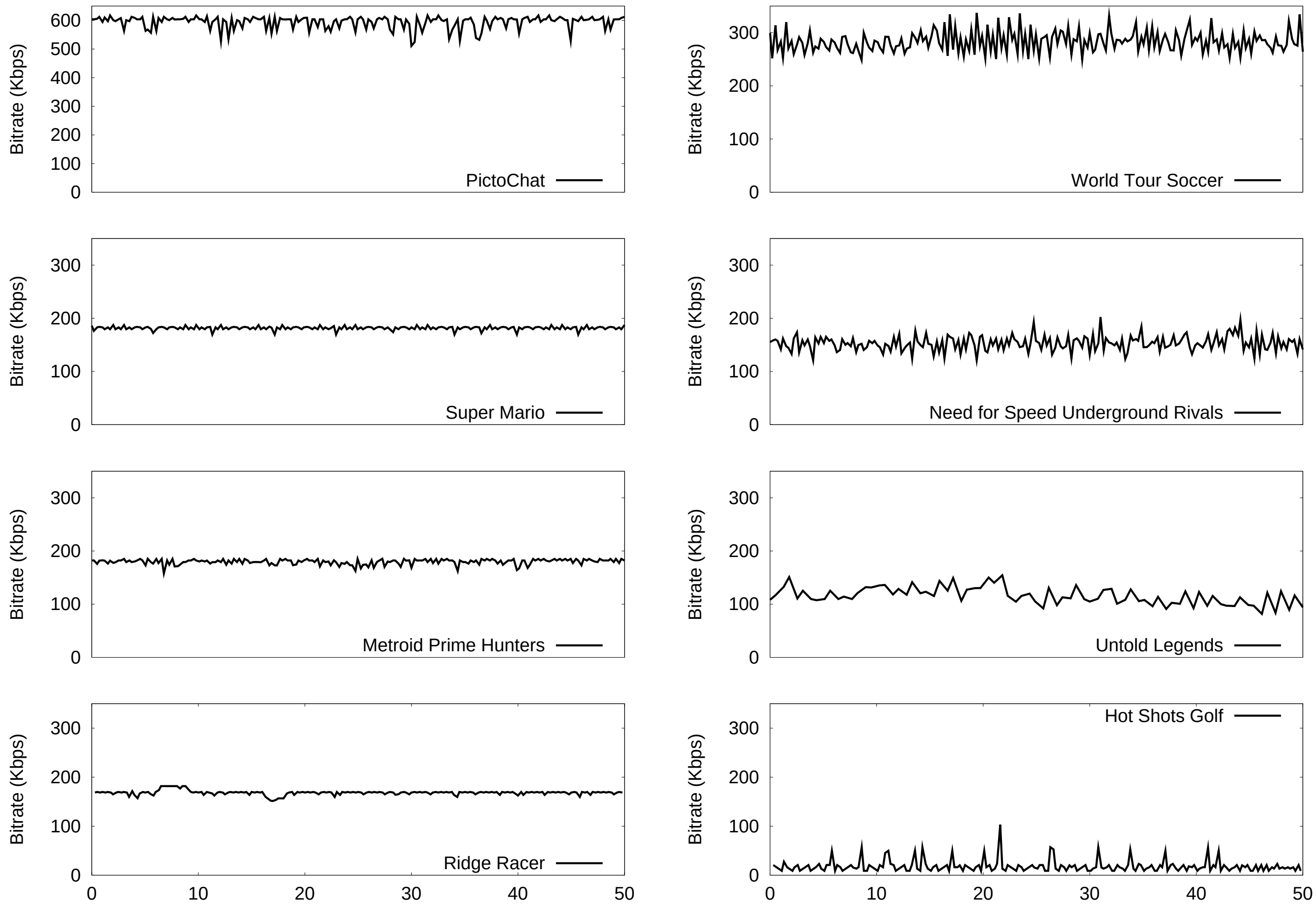
4.2 Turbulence

Figure 4 depicts the bitrate (in Kilobits per second) for the eight games used in the study, measured in 250 millisecond intervals, and Table 2 provides the means and standard deviations. Although the x-axes are all labeled from 0 to 50 seconds, the traces shown are for 50 consecutive seconds during a Play phase, but not necessarily at the start of the game.

Platform	Game	Bitrate (Kbps)	Stddev
DS	PictoChat	595	22.0
	Metroid	180	5.4
	Super Mario	182	3.4
	Ridge Racer	168	4.6
PSP	Soccer	284	19.8
	Need for Speed	153	14.0
	Legends	116	16.2
	Golf	18	12.5

Table 2: Bitrates by Game

Note the PictoChat graph (Figure 4(a), top) has a higher y-axis scale than all the other graphs since PictoChat uses significantly more capacity. Surprisingly, PictoChat has bi-



(a) Nintendo DS

(b) Sony PSP

Figure 4: Bitrate Comparison for Different Games

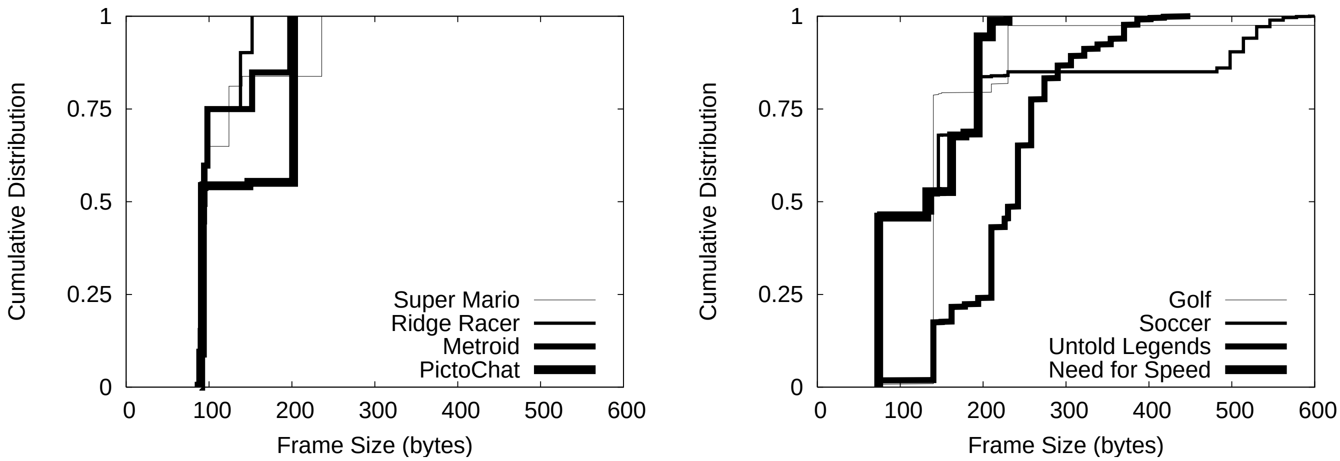
trates over 3 times that of the other DS games. This could perhaps be explainable if the images drawn on the stylus during chatting were large and uncompressed, but the bitrates are independent of the actual images drawn.

With the exception of PictoChat, all the Nintendo DS games have visually similar bitrates, around 175 Kbps, with a low standard deviation. The Sony PSP games have a much wider range of bitrates, with World Tour Soccer generating 50% more traffic than the DS games. The PSP racing game Need for Speed has about the same bitrate as the DS games, with Untold Legends having slightly lower bitrates, and Hotshots Golf having the lowest bitrate by far, about 10% of most of the others. Generally, the PSP games have a slightly higher variance in their network bitrate than do the DS games.

Comparing within genres, both the car racing games (Need for Speed on the PSP and Ridge Racer on the DS) have about the same bitrates but Need for Speed has a higher variance, while the 3rd person action games are slightly different (Untold Legends on the PSP has a slightly lower bitrate and more variance than Super Mario on the DS). The initial assumption that golf may have network turbulence similar to that of chat is totally incorrect as PictoChat has the highest bitrate and over 30 times that of Hotshots Golf, which has the lowest bitrate.

Figure 5 and Figure 6 depict cumulative distributions functions (CDFs) for the frame sizes and inter frame times, respectively. Many of the frames are quite small, around 100 bytes and many frames are sent within 1 millisecond of a previous frame. Visual inspection shows most of these small, closely space frames are empty acknowledgments (ACKs). Given that the 802.11 protocol requires an ACK for every data frame sent, the expectation is that about 50% of all frames recorded should be empty ACKs. All of the Nintendo DS games do have a steep rise in the frame size distribution around 100 bytes (Figure 5(a)), but two of the PSP games, Hotshots Golf and Untold Legends, do not. Looking at these traces in-depth, the PSP games Golf and Untold Legends appear to send much of their game data with broadcast frames which are not acknowledged in the 802.11 MAC protocol. Table 3 provides the percentage of frames that are sent by broadcast.

The use of broadcast is natural for a game with three or more players as only one `send()` is required for each data



(a) Nintendo DS

(b) Sony PSP

Figure 5: Frame Size Comparison for Different Games

Platform	Game	Percent
DS	PictoChat	1%
	Metroid	10%
	Super Mario	3%
	Ridge Racer	10%
PSP	Soccer	6%
	Need for Speed	8%
	Legends	96%
	Golf	98%

Table 3: Percentage of Frames Sent by Broadcast

packet sent, while directly addressable frames need a separate `send()` for every other player. However, the side effect of broadcast is that the data is received not only by all other game players, but by all other computers on the same 802.11 channel. The other computers not playing the game ignore the broadcast packet, but their processing is still interrupted so they can receive the packet, transfer it to main memory and then determine that they do not need it. Directly addressable traffic is discarded by the network card before interrupting the processor. The processor time wasted by other computers handling broadcast packets can be significant, and broadcast traffic generated by the game Doom plagued wired LANs in the mid-1990s [9], even prompting some companies and universities to implemented specific anti-Doom policies to reduce congestion on their LANs.

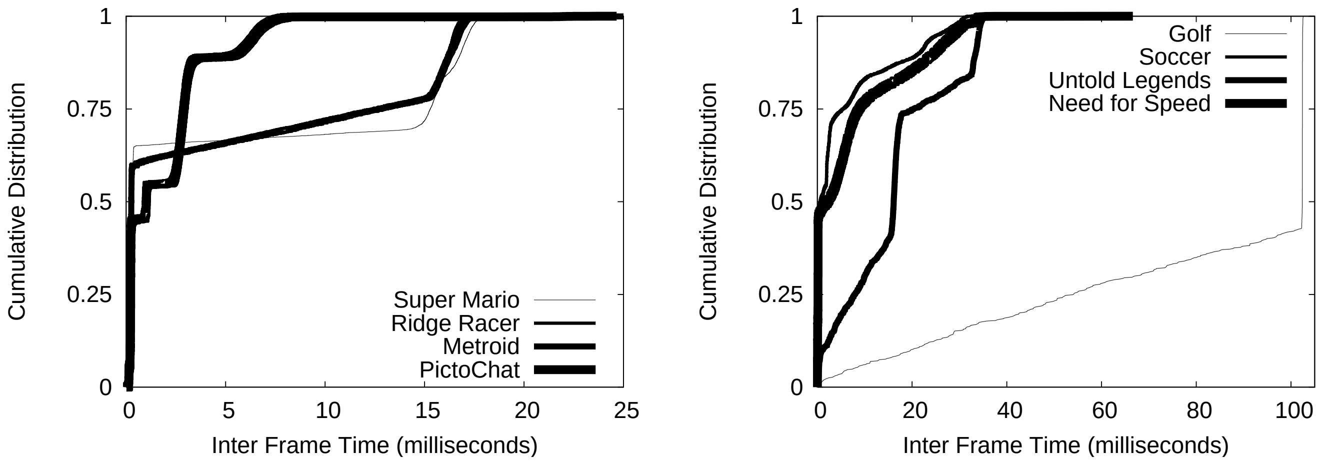
In general, the frames for all games studies are small and frequent, befitting the interactive nature of computer games. The frame sizes for the DS games are slightly smaller than the frame sizes for the PSP games, however only the PSP games World Tour Soccer and Untold Legends have a significant fraction of frames around 200 bytes. With the exception of the PSP Hotshots Golf game, frames are sent every 40 milliseconds at the most and as often as every 15 or 20 milliseconds.

Compared with the turbulence from other Internet games, the hand-helds send frames considerably more frequently than a typical real-time strategy (RTS) game, which has a signature of sending data every 100 ms [4], and more frequently than a typical first-person shooter (FPS) game, which has a signature of sending data about every 40-50 ms [3, 13]. The payload sizes for the hand-helds are of comparable size (small) to the payload sizes for both FPS and RTS games.

The evidence suggests it may be possible to use one model to represent 802.11 traffic for all Nintendo DS games. However, the Sony PSP games have a wider range of network turbulence making it unlikely one traffic model will accurately fit all games. Similarly, the disparity between the turbulence the DS games and the turbulence by the PSP games, not to mention the differences in their use of the 802.11b protocol, make it a challenge to find a unified model for the DS and the PSP. Furthermore, it is unlikely that a model developed for a PC or console game will accurately fit the network turbulence from a hand-held game.

4.3 Wireless Link Conditions

Since hand-held games are designed to be played in a variety of locations with varied wireless network conditions, preliminary analysis of hand-held networking performance under degraded connection signals is provided. Neither the Nintendo DS nor the Sony PSP provide a quantifiable wireless connection quality, such as signal-to-noise ratio. However, PictoChat on the DS provides "bars" indicating connection quality, visually similar to the bars found on many mobile phones. The PictoChat bars were used to generally assess the connection quality for 3 different indoor locations for pairs of hand-helds. A "3-bar", or *Excellent* connection, the highest, had the two hand-helds within 10 feet of each other in the same room. A "2-bar", or *Good* connection, had the two hand-helds about 25 feet away in separate rooms, where the players could not see one another but a doorway was open between the rooms. A "1-bar", or *Poor* connection, had the two hand-helds about 25 feet from each other and had one player around the corner outside the adjoining room from the first player.



(a) Nintendo DS

(b) Sony PSP

Figure 6: Inter Frame Time Comparison for Different Games

Figure 7 depicts the bitrates for separate runs (players are stationary for the entire game, so no mobility) of the same game under three different wireless network conditions. In Figure 7(a), the Nintendo DS bitrates noticeably increase in variance (and average, as seen in Table 4) with an decrease in the wireless connection quality. This can likely be attributed to frame loss on the wireless channel, resulting in retransmissions and perhaps even periodic re-synchronization of data sent by the games themselves. In Figure 7(b), the Sony PSP bitrates do not visually vary as much with a decrease in wireless connection quality, but from Table 4 there is a slight increase in bitrate average and variance with a decrease in wireless connection quality.

Moreover, although not depicted in the data, the game connection for the Nintendo DS is noticeably more robust than the connection for the Sony PSP. With the PSP, the Need for Speed game on the Good connection would break with an error of "Connection Lost" about 1 out of 2 times before the end of a race (about 2 minutes) while the Bad connection would break about 4 out 5 times before the end of the race. The connection never broke for the Nintendo DS Metroid game, although the Bad connection had intermittently unresponsive gameplay.

The bitrate differences and intermittent connections for PSP under degraded conditions may be attributable to the differences in the 802.11 WLAN capacities used. As mentioned in Section 2, the Nintendo DS uses only 1 or 2 Mbps for the WLAN capacity while the Sony PSP can use up to 11 Mbps for the WLAN capacity. Analysis of the Good and Bad traces shows that the DS does not autorate down to the lowest capacity (1 Mbps) when the wireless connection is bad, and instead always sends at 2 Mbps. The PSP does autorate down, but with a Bad connection, still sends about 85% of its frames at 11 Mbps with only 5% frames sent at each of 5.5 Mbps, 2 Mbps and 1 Mbps.

4.4 Impact on Other Traffic

In order to begin the study the impact of hand-held games on other 802.11 traffic, a preliminary investigation examines

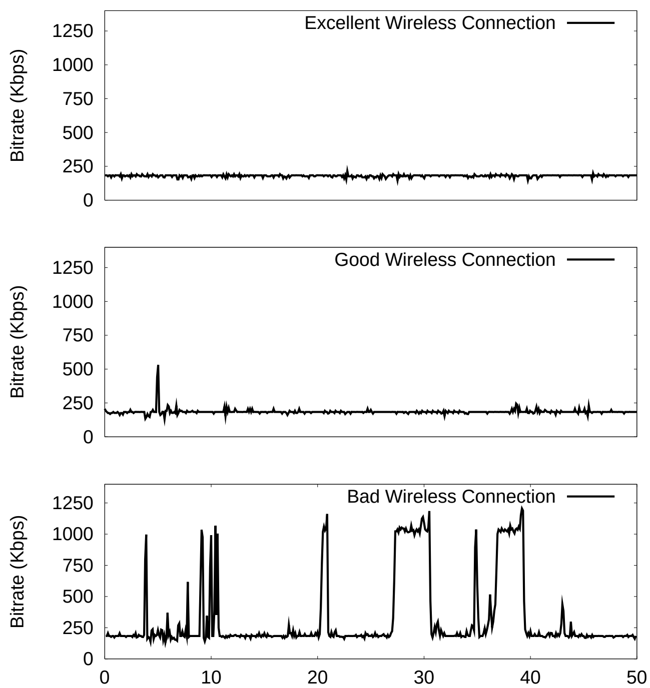
Game	Connection	Mean (Kbps)	Stddev
Metroid	Excellent	180	5.4
	Good	185	22.0
	Bad	325	302.5
Speed	Excellent	153	14.0
	Good	166	19.8
	Bad	171	15.4

Table 4: Bitrates by Wireless Connection Quality

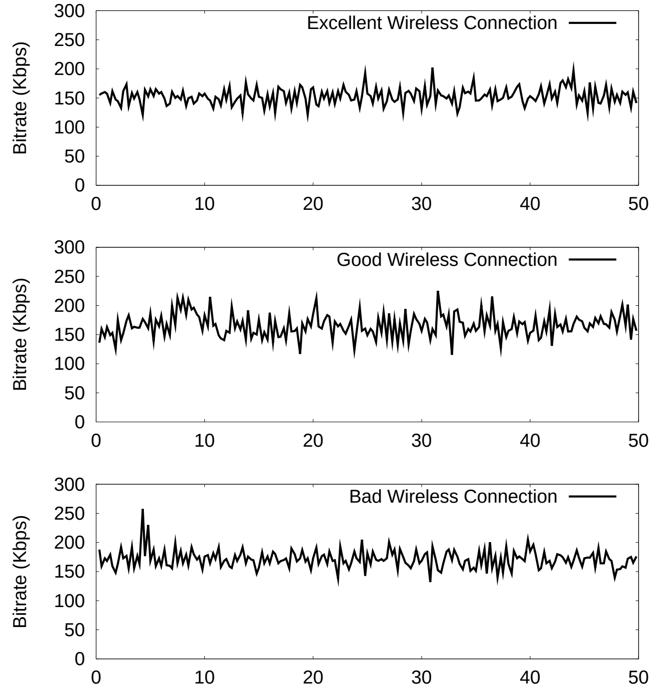
the impact of a Nintendo DS game on throughput from a TCP download sharing a 802.11 channel. At the start, a file transfer (using *wget*, a publicly-available TCP download application[9]) between two wireless laptops was started. A short time after, a two-player session of Super Mario on the Nintendo DS was started, doing synchronization by transferring data from one DS hand-held to the other. As shown in Figure 8, there is an immediate drop in the throughput for the download from slightly over 6 Mbps to just over 1 Mbps, less than half the throughput that might be expected if the DS and the download shared the 802.11 channel equally. When the synchronization phase of the Super Mario game completes, the throughput returns to the former levels. The other phases of Super Mario, most notably the Setup and Play, do not visibly affect the download.

As found in [8], the drop in the throughput is likely due to the IEEE 802.11 Distributed Coordination Function (DCF). Since DCF provides all hosts with an equal probability to access the wireless channel, hosts operating with a higher channel capacity wait nearly as long on average between sending frames as hosts operating at lower channel capacities. Thus, the average throughput of all hosts is reduced to the throughput of the host with the lowest channel capacity. These results illustrate how hand-held game consoles may affect Internet traffic sharing the same channel.

[9]http://www.gnu.org/software/wget/wget.html



(a) Nintendo DS – Metroid Prime Hunters



(b) Sony PSP – Need for Speed Underground

Figure 7: Bitrate Comparison for Different Quality Wireless Connections

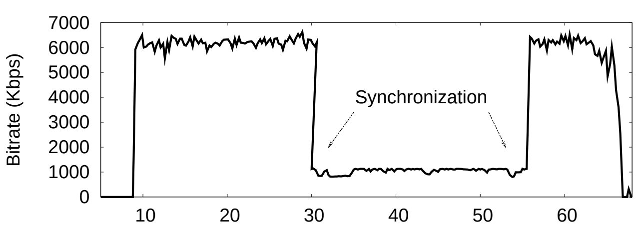


Figure 8: Impact of Hand-held Game Traffic on a Simultaneous TCP Download

5. CONCLUSIONS

The growth in the computer game market and the corresponding increase in networked games brings importance to the characterization of network game traffic. The spread of IEEE 802.11 wireless networks, most recently to the newest generation of hand-held game consoles, has made an understanding of the turbulence caused by hand-held games increasingly relevant. This paper analyzes the characteristics of eight 802.11 network games on two popular hand-held game consoles, the Nintendo DS and the Sony PSP.

Our analysis finds that hand-held network games go through distinct game phases: setting up, synchronizing data, playing and transitioning between other phases. The length and frequency of each phase varies from game to game, with the synchronization phase typically having the highest bitrates and the playing phase potentially lasting the longest. The focus of the analysis of this paper is on the playing phase as this represents the perceived "steady state" for a networked game and can be used to answer the question posed in the Introduction (Section 1):

1. Typical of other network games, hand-held network games make frequent sends of small frames of data. Typical hand-held frame payloads are less than 100 bytes and typical inter frame times (excluding acknowledgments) are less than 20 milliseconds.
2. In general, the traffic characteristics of Nintendo DS games are different than those of the Sony PSP games, with the DS games sending smaller frames than the PSP games, and the PSP having some games send mostly broadcast traffic while all the DS games studied use direct addressing.
3. Games on the Nintendo DS have network turbulence similar to each other, while games on the Sony PSP vary considerably in bitrate, frame size, frame frequency, and fraction of broadcast traffic.
4. The data frames sent by hand-held games are comparable in size to those of PC or console games, but the hand-helds send data more frequently, resulting in bitrates that are higher overall than typical PC or console games. This has implications for Internet links in the near future when hand-held game traffic is tunneled or sent over IP on the Internet.
5. Preliminary evidence suggests some game phases for hand-held games may have adverse affects on through-

put for Internet applications sharing the same 802.11 wireless channel.

Additional analysis shows some games for the Sony PSP primarily use 802.11 broadcast as a means of communicating between the hand-helds. This has implications for other wireless devices that share the same channel since broadcasts cause interruptions for all computers that receive them, even if they are not the intended recipients of the data. Also, under conditions of poor connection quality, the Nintendo DS game studied sends data at higher bitrates when the connection is degraded, while the Sony PSP game increases bitrates only slightly but has difficulty staying connected when the connection is degraded.

6. FUTURE WORK

This study concentrated on the network characteristics for two-player games during the play phase of the game. Since the hand-held devices themselves support up to 16 players, and several of the games tested support up to 8 players, future work could study the network characteristics for play phases with multiple players. Future work could also include a more detailed study of the network characteristics during other phases of multi-player networked games.

The widespread growth of 802.11 suggests wireless devices may increasingly compete for bandwidth on the same channel frequency. Future study that rigorously expands upon the preliminary interactions between Internet traffic and hand-held wireless game traffic may prove important.

The Sony PSP supports infrastructure mode, allowing connection to the Internet via a wireless access point. At the time of this paper, hand-held gameplay over the Internet was not available. As infrastructure games become available, future work could include studying the traffic characteristics during wide-area (Internet) play, as opposed to this local area play studied in this paper.

ACKNOWLEDGMENTS

I would like to thank Mingzhe Li for his help in setting up the laptop for wireless sniffing, and Akaash and Saahil for their help in gathering the experimental data. This research was partially supported by NSF grant CNS-0423362.

7. REFERENCES

[1] G. Bai and C. Williamson. The Effects of Mobility on Wireless Media Streaming Performance. In *Proceedings of Wireless Networks and Emerging Technologies (WNET)*, pages 596–601, July 2004.

[2] M. S. Borella. Source Models of Network Game Traffic. *Elsevier Computer Communications*, 23(4):403 – 410, Feb. 2000.

[3] W. chang Feng, F. Chang, W. chi Feng, and J. Walpole. Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, Nov. 2002.

[4] M. Claypool. The Effect of Latency on User Performance in Real-Time Strategy Games. *Elsevier Computer Networks (special issue on Networking Issues in Entertainment Computing)*, 2005. To appear.

[5] A. Cox. Cities find Wi-Fi future. *CNN.com*, Oct. 2004.

[6] J. Faerber. Network Game Traffic Modelling. In *Workshop on Network and System Support for Games*, Apr. 2002.

[7] T. Henderson, D. Kotz, and I. Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proceedings of International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, Sept. 2004.

[8] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proceedings of IEEE INFOCOM*, 2003.

[9] D. Kushner. *Masters of Doom*. Random House, 2003. ISBN 1588362892.

[10] T. Lang, P. Branch, and G. Armitage. A Synthetic Traffic Model for Quake 3. In *ACM SIGCHI Advances in Computer Entertainment (ACE)*, Singapore, June 2004.

[11] M. Li, M. Claypool, and R. Kinicki. MediaPlayer versus RealPlayer – A Comparison of Network Turbulence. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, pages 131 – 136, Marseille, France, Nov. 2002.

[12] T. Lang and G. Armitage and P. Branch and H-Y. Choo. A Synthetic Traffic Model for Half Life. In *Australian Telecommunications Networks & Applications Conference (ATNAC)*, Melbourne, Australia, Dec. 2003.

[13] S. Zander and G. Armitage. A Traffic Model for the Xbox Game Halo 2. In *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Stevenson, WA, USA, June 2005.