

# A Statistical Network Traffic Model for First-Person Shooter Games

Behnoosh Hariri and Shervin Shirmohammadi

**Abstract**—Online games have now become significant contributors to Internet network traffic. As such, research effort is now devoted to the analysis and modeling of this network traffic to aid network designers to provision for gaming traffic in the networks they design. While many research studies have performed measurements on gaming network traffic, the results are game-specific and have not been unified into a framework. In general, network traffic can be represented by packet size and packet interarrival time parameters; therefore, a synthetic model can be built to deal with these two parameters as traffic features. In this paper, we present a technique for constructing a Hierarchical Hidden Markov Model that provides a packet level statistical model for First Person Shooter (FPS) gaming network traffic, which allows generation of traffic for various numbers of users through different game states. The proposed solution has been implemented for Counter-strike and Quake as two of the most popular online FPS games. The results derived from the models have then been used to successfully predict certain game related statistics.

**Index Terms**—Network traffic generation, gaming traffic simulator, game state modeling.

## I. INTRODUCTION

For the past few years, studies and reports about online games conclude that computer games, now a significant economic sector and subject to both academic and industry research and development, continue to grow in popularity at a substantial rate. Online games, which enable hundreds of thousands of players to interact with each other over the network, are now widely used not just for entertainment, but also for socializing, business, commerce, scientific experimentation, and many other practical purposes. Literally millions of people spend their time and money in game worlds such as Second Life, Eve Online, Guild Wars, or World of Warcraft, the latter alone hosting more than 10 million subscribers [1] with a peak of 500,000 players online at a given time [2], or Entropia Universe, which holds the number 1 and number 2 record for the most expensive “virtual” real estate player-to-player sale, each over US\$300,000 per virtual property. A detailed analysis by In-Stat indicates that the revenue generated from such sales of virtual goods has increased 245%: from \$2.1 billion in 2007 to \$7.3 billion in

2010, and that such revenues will rise to \$14 billion by 2014 [3]. These are in addition to the player’s gaming purchases and subscription fees, which the NPD Group reports to be US\$3.3 billion just in the fourth quarter of 2011 and only taking into account the U.S., U.K., France, and Germany’s non-physical retail sales [4]. In fact, the Electronic Retailers Association reports that in UK alone video game sales hit £1.93 billion (US\$3 billion) in 2011 [5]. In China, another large and rapidly growing market, online games generated a revenue of 11.27 Billion Yuan (US\$1.77 billion) in the first quarter of 2012 alone [6]. The computer gaming market is therefore huge, and although the industry itself at approximately 30 years old is quite young, it has already surpassed the much longer-established film and music industries, generating more revenue than Cinema since 2009 [7] and more revenue than DVD/BlueRay since 2011 [5]. Considering that the gaming market is still growing at a very fast pace, and that it is remarkably tolerant against global economic crises due to its nature of “hundreds of hours of entertainment for pocket money” [8], we can see the importance and significance of this economic sector for many years to come.

As online games become more popular and significant contributors to Internet traffic, research efforts are devoted to the analysis and modeling of network game traffic. The importance of such research is twofold: for game designers to understand the traffic their game is producing and to control/shape it for better performance over the network, and for network providers such as ISPs to understand game traffic characteristics and to make provisions to support consumers. However, most of the previous work have measured real world traffic and tried to fit some standard distribution over it, where the distribution parameters follow several game and player related parameters. Such measurements usually do not address the issue of general traffic modeling and only apply to a specific game in a specific state and cannot be extended to other cases.

In this paper, we propose a generalized framework to cover all the previous measurements that have been performed for First Person Shooter (FPS) games: a popular genre of online games with the most stringent network requirements. This model can later be used as a traffic synthesizer for any FPS game and is useful as a modeler for all gaming related network protocols and design. Our modeling strategy is to describe the overall game states using Hidden Hierarchical Markov Model (HHMM) and to exploit the fact that traffic characteristics would depend on the game state, hence modeling the overall game as a statistical model that can then be used as a traffic synthesizer.

Section III includes a general concept of applying HHMM

Manuscript received September 25, 2013; revised December 20, 2013.

B. Hariri was with the Distributed and Collaborative Virtual Environment Research (DISCOVER) Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Canada. She is now with Google Inc. (e-mail: behnoosh@google.com).

S. Shirmohammadi is with the Distributed and Collaborative Virtual Environment Research (DISCOVER) Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Canada (e-mail: shervin@eecs.uottawa.ca).

to modeling online gaming traffic patterns, while Section IV and Section V describe the application of this concept to the famous Counter-strike and Quake online games. Section VI includes some experimental evaluations, proceeded by summary and conclusions in Section VII. We begin our discussions by a brief look at some related work.

## II. RELATED WORK

For FPS games, measurements have been reported for many games [9], such as Counter-strike, Quake, and Half-life. However, most of these works concentrate around the measurement results for a specific game and try to fit some standard distribution over the results. As such, the results cannot be used in a generalized case. Very few research works exist that work on predictive network traffic model generation for games, such as [10] which demonstrates the possible use of a predictive model for FPS traffic patterns where the distribution for multiple client games proposed to be found by convolving the distribution of lower player games. In this paper, we try to come up with a generic statistical model for FPS game traffic.

## III. HHMM APPLICATION TO GAME STATE MODELING

A detailed description of HHMMs is beyond the scope of this paper, and readers are referred to [11] for details. In brief, Hierarchical Hidden Markov Models (HHMM) are structured multi-level stochastic processes that can describe the internals of a system's states and probability transitions between those states. The states of an HHMM emit sequences rather than a single symbol. This process of recursive activations ends when we reach a special state that is known as a production state. These production states are the only states that actually emit output symbols through the usual HMM state output mechanism and are hence observable. A complete specification of an HHMM requires estimation of its model parameters, transition probabilities, and its probability distributions.

Most games utilize a client-server model. Every client's actions are sent in short messages to the server, and every client is regularly updated with the actions taken by other players. Generally, the game traffic can be modeled by independent traffic streams from each client to the server and a traffic stream from the server to the clients assuming that clients behave independently and client traffic is independent of the server traffic. Players run around shooting and interacting with each other. The game may also be paused as the server changes maps or restarts a previous map. A client transmit cycle consists of reading a server packet, processing it, rendering the client's current view on the screen, sampling input devices, then transmitting an update packet to the server for global updates. The client packet size usually depends on the player action such as walking, running, attacking and state in the game but independent of the number of players, computer hardware, or map type [9].

As the first step in the HHMM definition is the hierarchy level, we have arranged game states into a two-level hierarchy of events. The highest level is related to the game states where

the transitions are modeled using a markov model. The states in the lowest hierarchy level are production states: states that result in update message generation. The observation for the production states is assumed to be the packet size. In addition, a markov model is also used to model the number of players in each game state [12].

The choice of parameter values for the HHMM is an estimation problem and its usual criterion is maximum likelihood; i.e., finding the values of parameters which maximize the probability of the observed data. This is the problem that the Baum-Welch computation addresses. Baum-Welch training [13] is an expectation-maximization algorithm for training the emission and transition probabilities in an HMM structure. What's important about the Baum-Welch Training Method is that we can feed in observation sequences, and as long as we present enough data, we will get an HMM that is pretty close to optimal for the given training sequences. In this paper, we use Baum-Welch algorithm to estimate the HHMM parameters for the traffic model. The algorithm starts with an initial estimate of HHMM parameters and converges to the nearest local maximum of the likelihood function. Initial state probability distribution in each hierarchy level is considered to be uniform, that is, if there are  $N$  states, then the initial probability of each state is  $1/N$ .

In the following two sections we describe the application of such HMM Model to two FPS games: Counter-strike and Quake III. These two games are chosen among the most popular representative of FPS shooter games. However it should be noted that the same modeling strategy can be applied to any other FPS game as well. We will then use traffic measurements from counter-strike servers to train the corresponding counter-strike HHMM model using Baum-Welch Training Method as described before. Finally we will compare the trained model synthesized traffic versus another measurement set to verify the model validity and accuracy.

## IV. COUNTER-STRIKE

Counterstrike (counterstrike) is a tactical first-person shooter game. It is a gaming environment in which a team of counter-terrorists fight against a team of terrorists in a series of rounds. The game's communication model follows the client server approach and uses UDP packets for the exchange of small update information. Traffic consists of client and server generated packets. Thus, the game traffic can be modeled by independent traffic streams from each client to the server and a burst traffic stream from the server to the clients. However it should be noted that client generated traffic is mostly constant in terms of both packet size and packet interarrival time and does not vary that much depending on the game conditions and change in the number of players [14]. Since client streams can be assumed as independent processes, the overall server incoming process can be found by superimposing these independent traffic sources. Traffic rates from the server to one client do not vary that much [9] and were mainly designed to saturate the narrowest last-mile link; therefore, clients can similarly play

at modem speeds. Therefore we model the server outgoing traffic mainly characterized by packet size and packet interarrival time in order to define a packet oriented model. But first, let us look at the traffic pattern already measured by other studies.

A. Traffic Measurement

The transition pattern within a session depends on players' arrival process and session distribution. User interarrival process is observed to have a heavy-tailed nature, and shows a strong time-of-the-day effect [9]. The latter suggests that a time series model such as ARIMA (Autoregressive Integrated Moving Average) can be chosen as a good fit [15] since the probability of returning from each state to the previous states depends on both the arrival process and the session time.

It has been shown that a significant number of players play only for a short time before disconnecting, and that the number of players that play for longer periods of time drops sharply as time increases [16]; However the game duration process has a Heavy-tailed distribution due to a number of long-time players [15]. The two-parameter form of Weibull distribution with  $\beta= 0.5$ ,  $\eta= 20$ , and  $\gamma= 0$ , closely fitting the probability distribution function (PDF) of measured session times for the trace PDF [14],[16].

The server traffic is constantly high during a game turn, but between turns the server traffic may drop to zero for a short time. A large high-frequency component is observed in the total server traffic (incoming and outgoing) at a frequency of around 20Hz (50 ms) with periodic variations occurring twice an hour, in addition to other low frequency components [14][16]. The frequency component measured around 50ms is the result of the synchronous operation of the game server logic itself which is written to deterministically send state updates to the clients. There is also a frequency component around 30 minutes due the 30 minute map time of the server during which a dip in traffic occurs when the server is doing local tasks to perform the map change over. Other low frequency components happen due to round changes. Around 70% of all packets are sent immediately one after the other, around 25% of all packets show an interarrival time of 20 ms, 4% are sent after a gap of 40 ms, and the remaining 1% arrive every 60 ms. The burst interarrival mean is measured to be 64ms [14] and 79ms in [17].

Server packet size itself has a wide distribution and linear dependence on the number of clients' packets [14]. Therefore, the server packet size depends on the number of clients. The output distribution has been also considered to have heavy tail behavior. Extreme (a (n)=(34.5 + 4.2n), b(n)= (9 + 3 n)) distribution where n is the number of players seems to be a good approximation or server packet size in the range {20:1000} [14].

B. HHMM Traffic Model for Counter-Strike

As the measurement results show a strong dependence of server outgoing traffic on the number of users, modeling the users' join and leave process is the first step towards the traffic modeling. Fig. 1 shows the Top-level discrete markov model for modeling the users joining and leaving process where state Ni represents i-1 number of active users in the game and k+1 is the maximum number of users that can be

served by a server.

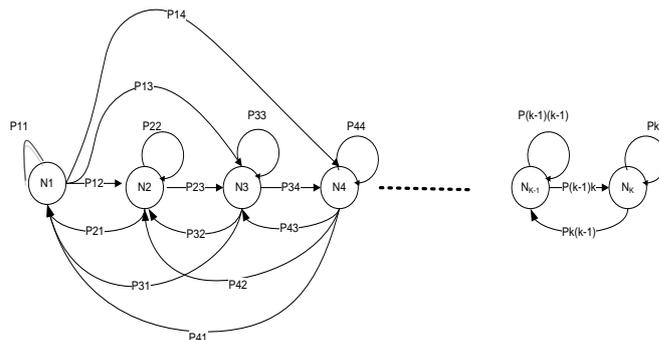


Fig. 1. Top-level markov model for modeling the players entering and leaving a session.

The server traffic model itself can be categorized in the following states:

Traffic Bursts to clients: this models the interval when the server sends the packet in burst to clients. Server usually divides this burst interval into 3 subintervals [14], as listed below, where each of the intervals update a number of clients and the length of the burst depends on the number of clients:

- 1) The time when server starts sending packets from a new burst cycle. This state would therefore model the burst interarrival time.
- 2) The traffic dip state that occurs due to map change.
- 3) The last state that model the packets transmissions that don't fall within regular in-game traffic.

The game transitions among subsequent game rounds can therefore be modeled with a lower order markov model with 4 states where states G1-G4 are game related states corresponding to the above packet interarrival times. Fig. 2 shows the previously explained state diagram. All states will eventually end in production states where packets are generated and sent. State G1 results in another production state that models the packet transmission during the burst. State G2 models the burst interarrival time while state G3 models the server traffic dip according to server local operation during a map change and does not end in any production states. Finally, state G4 goes to a production state where packets in response to random clients' request are generated.

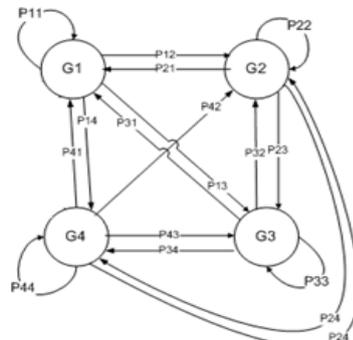


Fig. 2. Higher level Markov model for modeling the game rounds.

Two sets of observations from the production states are defined for the Markov model: packet size, and Log of packet interarrival time. The Log of the packet interarrival time is performed in order to compensate for the wide distribution of numbers that have been measured for the packet interarrival

time and make them more evenly spaced. We assume that packet size and packet interarrival time are statistically independent, and we have chosen their distributions according to traffic measurement results described earlier (i.e., real-world measurements). Normal distribution, represented by its mean and variance in each of the production states is chosen for states G1-G3. This can be explained as follows: state G1 models the packet interarrival time within a burst where packet transmission is mostly limited by the bandwidth connection among the client and server and processing limitations. Assuming the clients have roughly the same processing and networking powers (guaranteed by minimum system requirements of the game), the server packet interarrival time will be distributed over a mean value. State G2 models the regular burst interarrival time that can also be approximated with another normal distribution, and state G3 would also model the game traffic during the map change which is another normal distribution with a smaller average. Finally the packet interarrival distribution for state G4 is assumed as Poisson due to the random and uneven distribution of client requests. We have assumed the packet size to have an extreme distribution for states G1-G3 according to the measurement results. Therefore the two extreme distribution parameters will represent the output in these two states. The packet size distribution for state G4 is assumed as uniform due to the various payload sizes in response to different client requests where distribution interval is considered as the state output parameters.

## V. QUAKE III

Quake is another FPS game that, in its multiplayer mode, players on several computers connect to a server where they can play against each other. The goal is for each player to kill the other players as many times as possible. UDP is used here too as the transport protocol. In order to describe the HHMM model, we have followed the same approach for counter-strike. First, we present measurement results for the game made by other studies, then we describe the model that best fits to these results.

### A. Traffic Measurement

In Quake, the packet lengths from the individual clients to the server is independent of map type or number of players and is limited between 50-70 bytes. The server updates the clients in regular burst similar to counter-strike. Measurements demonstrate that although server packet rate depends on the number of players, the burst interarrival does not change with the number of players. The mean of server interarrival time for individual clients is about 50 ms [18], [19]. Therefore  $(100*(L-1)/L)\%$  of the server packets interarrival are close to 0 ms and the rest are concentrated around 50 ms where L is the number of active clients [19].

The packet length generated by the server has a strong dependence on the map type and number of players with a distribution that grows wider as the number of players in the game increases. Each map has a base packet length distribution, and each additional player on the map creates a certain amount of activity that increases the packet lengths by 12 or 13 bytes [19]. The dependency of the packet map can be

ignored since the study shows that the number of clients is the dominant parameter. The distribution is then modeled as an addition of a base distribution plus a second distribution that models the increase per client added, where the base distribution is lognormal (79.340543, 0.24507092) and the additive distribution per player is an exponential distribution with a mean of 13 [19]. Finally measurement results show that players' duration times fit an exponential distribution, while interarrival times fit a heavy-tailed distribution, and the session interarrival time fits an ARIMA (Autoregressive Integrated Moving Average) model [20]. The above information is used to develop the HHMM traffic model for Quake, which is presented next.

### B. HHMM Traffic Model for Quake III

Due to Quake's server packet size and packet rate being strongly dependent on the number of active players, we will use a Markov model to represent the players' joining and leaving process in the same way as the model that we previously used for counter-strike in Fig. 1. This will be a good choice considering both the heavy-tailed distribution and the autoregressive behavior. The second order Markov chain deals with the traffic transmission states of the server. The server to client packet inter-arrival times is very regular in Quake compared to counter-strike and transmission cycles can be assumed to be one of the following two cases:

- 1) Traffic Bursts to clients: this models the interval the server sends the packet in burst to clients.  $(100*(L-1)/L)\%$  of the packets are sent in this state where L is the number of active clients.
- 2) The time when server starts sending packets from a new burst cycle. This state would therefore model the burst interarrival time that is expected to be very close to 50ms in Quake.

The second order Markov chain would therefore be the same as the one shown in Fig. 2 except that it has two states instead of four. The state G1 presents the packet transmission within a burst while state G2 models the burst interarrival time. The two Observation sets are again assumed to be packet size and Log of packet interarrival time. The normal distribution has been chosen as the best match for packet interarrival time in state G1 for to the same reasons that was explained for counter-strike. However we choose the deterministic 50 ms value for the interarrival time in state G2 instead of a Gamma distribution to keep things simpler considering the fact that measurement results show that packet interarrival time is densely distributed around 50ms. For both states, we assume the packet size to have a mixed distribution of lognormal and K number of Poison distribution where K is the number of players in that state.

## VI. EXPERIMENTAL EVALUATION

In the previous sections we have proposed HMMM models that have discrete states. The Observations have been assumed to be continuous bi-dimensional vector where the first observation is the distribution of packet size and the second observation is the distribution of the Log of packet size. The whole model is specified with the horizontal and vertical probability of transition among the states and one or

two parameters to specify each output element distribution that can be the mean and variance or only the mean or other parameters depending on the type of distribution.

This HHMM model was then trained to describe the states in counter-strike as proof of. We also designed the HHMM model for Quake III but were not able to train it since we were unable to find a log file for that game. However, the same modeling and training strategy can be applied to Quake III or any other FPS game.

After training the model, we verified it by predicting the probability distribution of packet size and interarrival over time. Fig. 3 and Fig. 4 show the results of the predicted data in comparison with the real-world measured results. The results clearly demonstrate the fact that the model is a good fit for the measurements, for all practical purposes.

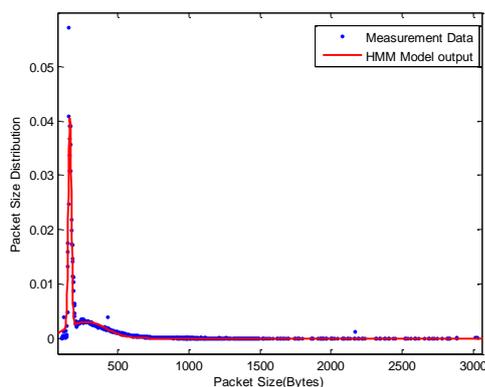


Fig. 3. Packet size distribution: measurement results and model output.

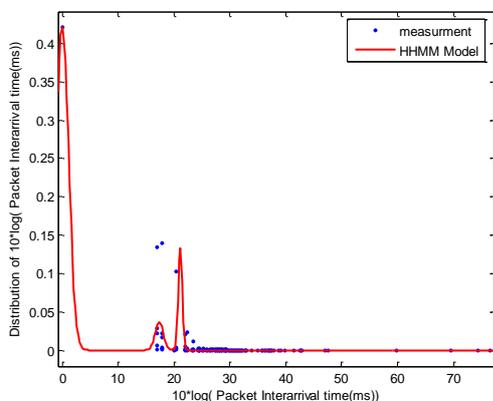


Fig. 4. (10\*log) function of Packet interarrival time distribution: measurement results and model output.

We also applied the goodness of fit test, using the coefficient of determination,  $R^2$ , which is the proportion of variability in a data set that is accounted for by a statistical model as defined in (1).

$$R^2 = 1 - \frac{SS_E}{SS_T} \quad (1)$$

where  $SS_E$  is the sum of square errors between the predicted and the observed values, and  $SS_T$  is the total sum of the squares; i.e., the squared sum of observed point deviation from their mean. For Fig. 3 and Fig. 4, the  $R^2$  coefficients were calculated to be 0.9294 and 0.6998 respectively. The reason that the latter value doesn't seem as high as the accuracy seen in Fig. 4, is the wide and non-continuous distribution of the measured data in the figure (blue curve), leaving some points which have characteristics that don't fall within any of the

Markov states. But, as it can be seen clearly from the graph, the predicted values follow very closely the measured values and the result is satisfactory.

## VII. SUMMARY AND CONCLUSION

The objective behind this work was to propose a generalized statistical framework to cover previous measurements that were performed on FPS game traffic. We aimed at proposing a statistical model for gaming traffic synthesis, motivated by the fact that gaming traffic is generated by client-server interactions during game sessions and therefore it should be possible to model the game states and later use this model to extract the traffic pattern. In order to deal with the complicated state sequence, we proposed to incorporate a hierarchical structure in representation of activities, using HHMM, in which a state is represented by a combination of multiple basic units. A series of measurement data for some popular online FPS games have also been used as the model training data, and the model was trained using the classic Baum-Welch algorithm. The model has then been used to extract some traffic-related information, which was found to closely follow real-world measurements. Based on this, traffic generators can be built that would help game designers and backbone providers simulate games for a large number of geographically-distributed players.

## REFERENCES

- [1] A. Holisky, "World of warcraft subscriber numbers remain at 10.2 million," *Joystiq News*, May 9, 2012.
- [2] S. Shirmohammadi and M. Claypool, "Massively multiplayer online gaming systems and applications," *Multimedia Tools and Applications*, vol. 45, no. 1, pp. 1-5, 2009.
- [3] K. Nissen, "Virtual goods in social networking and online gaming," Stat Research Report, November 2010.
- [4] *The NPD Group Announces Fourth Quarter Consumer Spending on Video Game Content Outside of New Physical Retail Sales in U.S. and Key European Markets*, NPD Group Press Release, March 22, 2012.
- [5] Game Central, "UK game sales beat film and TV for the first time," *Metro Magazine*, March 22, 2012.
- [6] *Tencent Lead China Client Online Game Market 2012 Q1*, Analysys International, May 23, 2012.
- [7] T. Chatfield, "Videogames now outperform hollywood movies," *The Guardian*, September 27, 2009.
- [8] A. Allen, "Gamer's unaffected by recession," *The Extreme Pulse*, June 20, 2012.
- [9] S. Ratti, B. Hariri, and S. Shirmohammadi, "A survey of first-person shooter gaming traffic on the internet," *IEEE Internet Computing*, vol. 14, no. 5, pp. 60-69, September/October 2010.
- [10] P. Branch and G. Armitage, "Towards a general model of first person shooter game traffic," Center for Advanced Internet Architecture, University of Swinburne Technical Report, 2005.
- [11] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: analysis and applications," *Machine Learning*, vol. 32, pp. 41-62, 1998.
- [12] B. Hariri, S. Shirmohammadi, and M. R. Pakravan, "Hierarchical HMM model and measurement of online gaming traffic patterns," in *Proc. IEEE International Instrumentation and Measurement Technology Conference*, Victoria, Canada, May 12-15, 2008, pp. 2195-2200.
- [13] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Ann. Math. Statistics*, vol. 41, no. 1, pp. 164-171, 1970.
- [14] J. Farber, "Traffic modeling for fast action network games," *Multimedia Tools and Applications*, vol. 23, pp. 31-46, 2004.
- [15] A. Sinha, K. Mitchell, and D. Medhi, "Network game traffic: a broadband access perspective," *Computer Networks*, vol. 49, pp. 71-83, 2000.

- [16] W. Feng, F. Chang, and J. Walpole, "A traffic characterization of popular on-line games," *IEEE/ACM Trans. on Networking*, vol. 13, no. 3, pp. 488-500, 2005.
- [17] D. LaPointe and J. Winslow, "Analyzing and simulating network game traffic," Project Report, Worcester Polytechnic Institute, 2001.
- [18] H. Park, T. Kim, and S. Kim, "Network traffic analysis and modeling for games," *Lecture Notes in Computer Science*, vol. 3828, pp. 1056-1065, 2005.
- [19] T. Lang, P. Branch, and G. Armitage, "A synthetic model for Quake 3 traffic," in *Proc. ACM Conference on Advances in Computer Entertainment Technology*, Singapore, June 3-5, 2004, pp. 233-238.
- [20] T. Henderson and S. Bhatti, "Modeling user behavior in networked games," in *Proc. ACM International Conference on Multimedia*, Ottawa, Canada, September 30 - October 5, 2001, pp. 212-220.



**Behnoosh Hariri** received her PhD degree in electrical engineering from Sharif University of Technology, Iran, in 2009, and she was a MITACS elevate postdoctoral fellow at the Distributed and Collaborative Virtual Environment Research Laboratory (DISCOVER Lab), University of Ottawa, Canada, from 2009 to 2011. Her research was focused on gaming systems and networks, especially massively multiuser virtual environments, as well as

vision based multimedia systems. She currently works at Google Inc.



**Shervin Shirmohammadi** received his Ph.D. degree in electrical engineering from the University of Ottawa, Canada in 2000, where he is currently a full professor at the School of Electrical Engineering and Computer Science. He is co-director of both the Distributed and Collaborative Virtual Environment Research Laboratory (DISCOVER Lab), and Multimedia Communications Research Laboratory (MCRLab), conducting research in multimedia systems and networking, specifically in gaming systems and virtual environments, video systems, and multimedia-assisted biomedical engineering. The results of his research have led to more than 200 publications, over a dozen patents and technology transfers to the private sector, and a number of awards and prizes. He is associate editor-in-chief of *IEEE Transactions on Instrumentation and Measurement*, associate editor of *ACM Transactions on Multimedia Computing, Communications, and Applications*, and was associate editor of Springer's *Journal of Multimedia Tools and Applications*, and chairs or serves on the program committee of a number of conferences in multimedia, virtual environments, and games. Dr. Shirmohammadi is a University of Ottawa Gold Medalist, a licensed professional engineer in Ontario, a senior member of the IEEE, and a professional member of the ACM.