Live Generation & Simulation of First Person Shooter Game Traffic

Carl Javier
Centre for Advanced Internet Architectures, Technical Report 100226A
Swinburne University of Technology
Melbourne, Australia
cjaver@swin.edu.au

***Abstract*—This technical report outlines techniques and theories of generating live and simulating first person shooter game traffic for N specific players. We outline a two approaches. Firstly creating live network traffic using a tool called pktgen. Secondly we outline how to simulate using Omnet Network simulator and Network Simulator with a C library called 'ArmaPktLen'. All approaches use a combination of Auto Regression Moving Average (ARMA) statistical prediction model and convolution techniques.**

I. Introduction

Consumer broadband networks are emerging to be complex systems that has yet to be completely analysed and understood by ISPs and researchers. A typical home network can now cater for various applications (such as VoIP, online gaming and video streaming) that utilise consumer grade broadband.

One application we consider to affect greatly the use of multiple home broadband application is online first person shooter (FPS) games. FPS Online games consists of home user clients connecting to a game server or another home user client. Network traffic from the user to server and vice versa varies during game play. At times, game network traffic can exhaust the link so other applications running inside the home network may perform poorly or fail. This is an issue that Industry Service Providers (ISPs) and researchers need to understand and cater for when designing and implementing consumer broadband networks.

To assist researchers and the ISPs with network simulations and predictions we have produced a live packet generation tool and C simulation library. Using existing statistical work [1] [2] our tools can create or simulate network traffic of a user playing an online first person shooter game. The packet generator can be placed into live broadband network test beds to experiement with the interactions of other home online applications. The simulation library can be used by simulation packages such as Omnet++ [3] and Network Simulator (NS) [4]. Currently the tools models server to client behaviour from 2 - N players on a server. The games that can be simulated are the following:

- Half Life Counter Strike (HLCS)
- Half Life Death Match (HLDM)
- Half Life 2 Counter Strike (HL2CS)
- Half Life 2 Death Match (HL2DM)
- Quake 3 (Q3)
- Quake 4 (Q4)
- Wolfenstein - Enemy Territory 2 Pro (ET2PRO)

II. Background

Online first person shooter games use a server to client architecture to communicate. All game clients connect to one server. The server is where the game environment and client activities are processed. All connected clients view the current game environment via periodical updates sent from the server. Constant periodical updates are also sent by the client about its activity in the game environment.

Data mining research [5] [6] has shown that server to client network packet sizes is far more variable as the number of clients connect to the server. Further research demonstrates server to client packet distributions can be modelled using signal prediction method called Auto Regression Moving Average(1,1) (ARMA) [7], [1] and [2]. However packets from the client to server distrubtions is considered periodically constant and predictable.

A. ARMA Packet Size prediction process

In [7] Crienti et al showed that a FPS server to client packet size time series is well modelled by ARMA(1,1). ARMA(1,1) is modelled by (1).

$$X_t = \phi_1 X_{t-1} + \theta_1 Z_{t-1} + Z_t \quad (1)$$

Author's email changed to cjavier@gmail.com

The residuals Z_t are independent and indentically distributed random variables.

To model online first person shooter traffic network traffic, intial data from a real online game is required. Periodical timeseries of packet lengths from a 2 and 3 player online game (server to client) is required to obtain a residual time series. The packet length time series should have zero mean. Game packet generator uses existing 2 and 3 player traffic traces from the SONG database [8].

The next step is for the residual time series to be are generated. From the residual time series a residual probability distribution function (PDF) is formed. Taking the residual PDF of a 2 (Figure 1) and 3 (Figure 2) player residual, a convolution (2) of the two creates a 5 player PDF of residuals (Figure 3).

$$Z_{5,t} = Z_{2,t} * Z_{3,t} \tag{2}$$

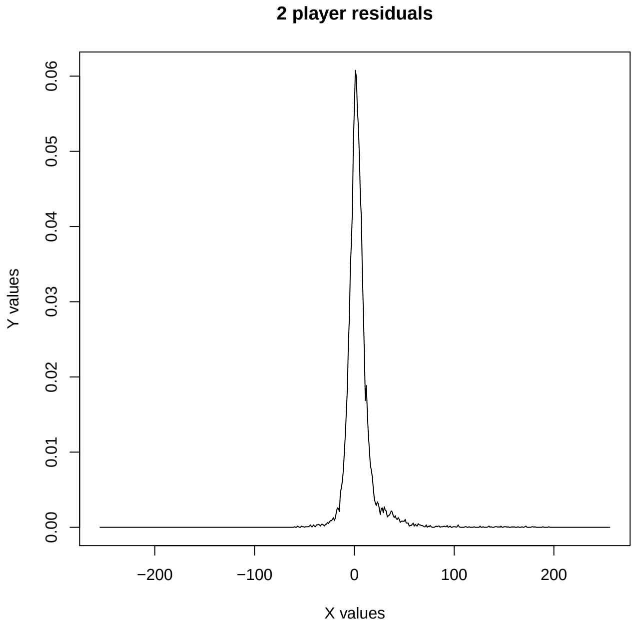

Fig. 1. 2 player PDF of residuals.

From the 5 player residual, we can extract and predict a time series for a 5 player game (Figure 4) by using the cumulative distribution function (CDF) and a random (0 - 1) generator.

To obtain other N player time series, we simply use the combinations and results of a 2 and 3 player residual PDFs. For example if a 10 player packet length time series was created, then the residual PDFs of a 5 and 2

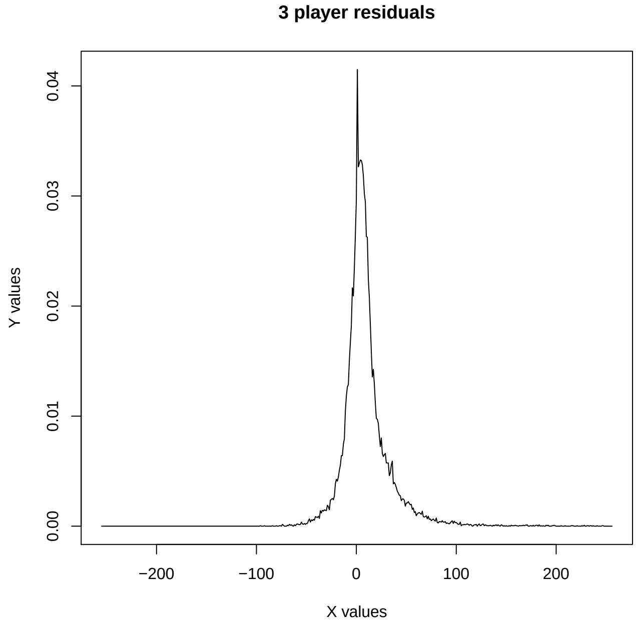

Fig. 2. 3 player PDF of residuals.

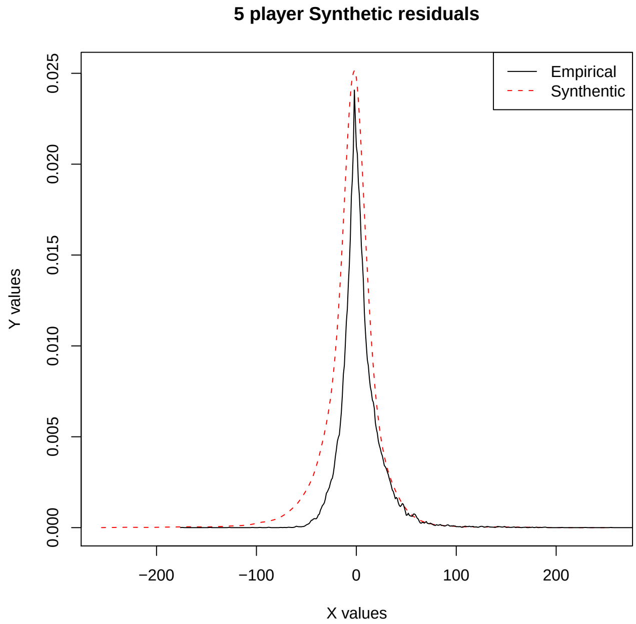

Fig. 3. Convolving residuals from a 2 and 3 player game are used to construct the 5 player residual distribution.

player would need to be used. Equation (**??** is the basic formulae for predicting N player residuals.

$$Z_{n,t} = Z_{a,t} * Z_{b,t}, \quad where\ a + b = n \tag{3}$$

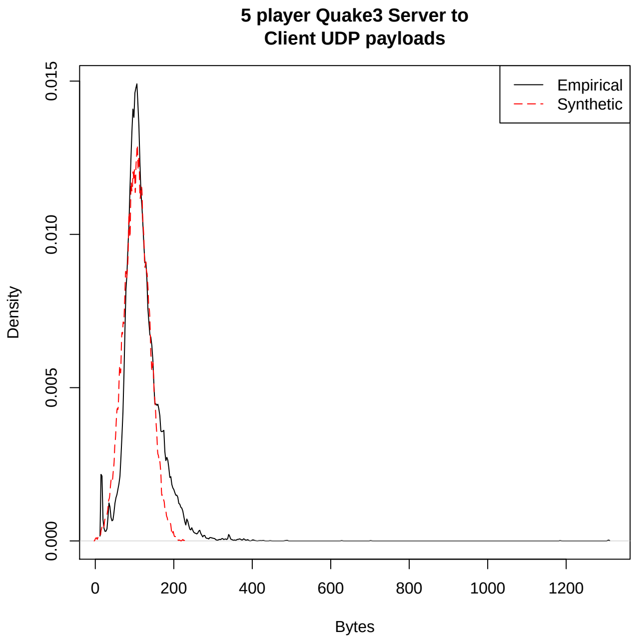

Fig. 4. 5 player packet distrubtion comparison of empirical and synthetic game traffic. The residuals from a 2 and 3 player game were used to construct the 5 player packet distribution.

Figures 5, 6 and 7 demonstrate the packet length distributions for Quake 3. Other FPS games we have tested have similar results when comparing to the empirical data. Unfortunately only emprical 9 player statistics could be used to compare with our generator.

III. Live Game Traffic Generation

This section describes the different parts of the game generator tool called ‘pktgen’. Any specific processes or tweaks in the tool is mentioned in this section.

‘pktgen’ is part of the Broadband Internet Traffic Simulation & Synthesis(BITSS) project at the Centre for Advanced Internet Architecture at http://caia.swin.edu.au/bitss/ .

‘pktgen’ is designed to synthesise network game traffic a typical home broadband user would encounter. Synthetic network traffic can be used by ISPs, network engineers, and researchers to test the capabilities of networks.

Synthetic game traffic can be injected into the network or be saved as pcap file for later analysis

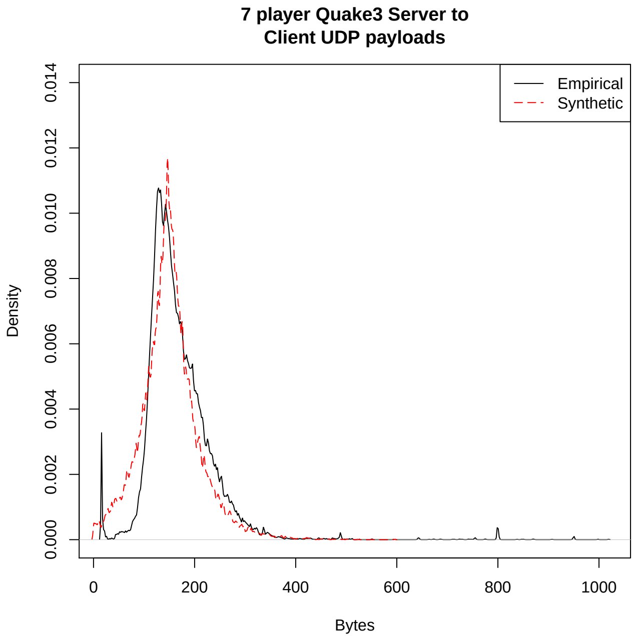

Fig. 5. 7 player packet distrubtion comparison of empirical and synthetic game traffic. The residuals from a 4 and 3 player game were used to construct the 7 player packet distribution.

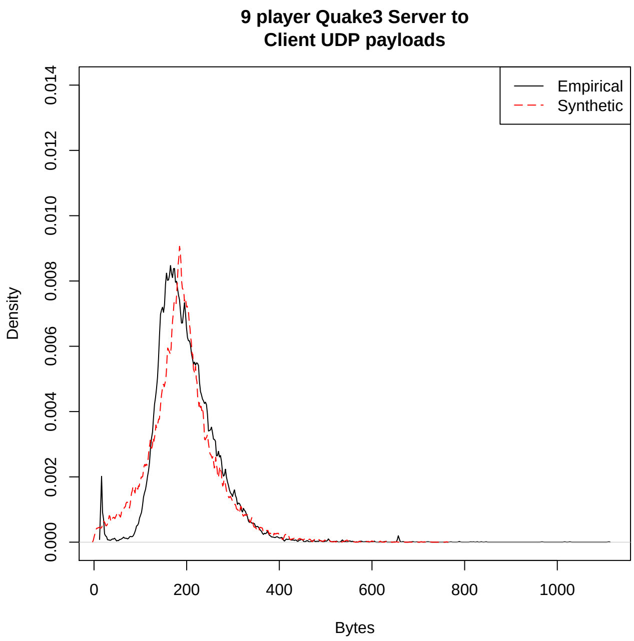

Fig. 6. 9 player packet distrubtion comparison of empirical and synthetic game traffic. The residuals from a 4 and 5 player game were used to construct the 9 player packet distribution.

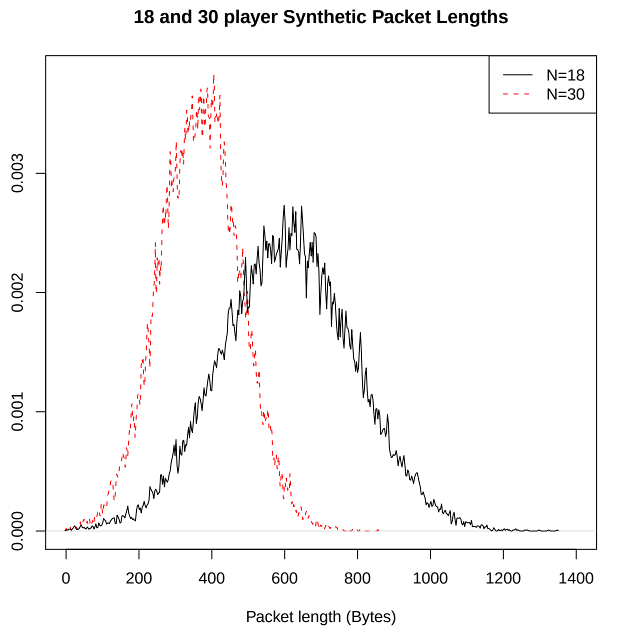

Fig. 7. 18 and 30 player packet distrubtion based on synthetic game traffic.

SWITCH	DESCRIPTION
game	Specify q3, hl2cs, hl2dm, hlcs, hldm, et2pro, q4
N	No. of Players (4 - 32)
IP	Destination address to send packets to.
port	Destination port to send packets to, else default game port is used
phi	Overwrite the predefine Phi value
theta	Overwrite the predefine Theta value
iat	Specify inter-arrival time
cap	A 2 and 3 player pre-recorded packet capture file to be used as stats.
write	Write generated packets to pcap file
IPsrc	Source IP when generating pcap files, default tries to resolve hostname else loopback is used
hrs,mins,secs	Hours, Minutes or Seconds packet generation should last for.
pkts	Number of packets packet generation should last for.
debug	Writes information in the debug folder about the packet length generation process.

TABLE I
PKTGEN COMMAND LINE WWITCHES.

A. *pktgen.py*

The main section of the tool resides in pktgen.py. Running pktgen.py without any switches will display the help screen. There are two ways to use pktgen.py. One is to use it as a live packet generator on a real network. Secondly it can write packets to a pcap file for use in simulators such as Ixia's NX2 [9].

The minimum switches needed to send live network game packets is the game type, number of players and the IP address to send it to. The following is an example:

```
./pktgen.py -game q3 -N 5 -IP 192.168.4.23
```

To create a pcap file of generated traffic, you simply specify the filename and include any other switches that you would use to generate live packets. The following is an example :

```
./pktgen.py -game q3 -N 5 -IP 192.168.4.23
    -write pcap.dmp
```

Table I explains the rest of the switches.

1) Customising game packet generation: There are a few statistical features that can be customised or catered to when using the game generater. The values of phi, theta, inter arrival time (iat), game port number which are predefined in [1] [7] [2] can be changed by using the switches `-phi -theta`.

Table II shows the default values for ϕ, θ, inter arrival times and port numbers.

Game	ϕ	θ	IAT(sec)	port
ETPRO	0.98	0.68	0.5	27960
HL2CS	0.79	0.12	0.3	27015
HL2DM	0.97	0.71	0.3	27015
HLCS	0.96	0.77	0.5	27010
HLDM	0.97	0.84	0.5	27010
Q3	0.97	0.77	0.5	27960
Q4	0.95	0.60	0.5	28004

TABLE II
PREDEFINE GAME STATISTICS FOR GENERATING N PLAYER DATA TAKEN FROM [1] [7] [2].

'pktgen' also can take in two trace files to generate an N player game. Trace files of 2 and 3 player game can be used to perform the same convolution and ARMA methods to produce an N player game. The trace files should be filtered to show actual game play. Game play is when every client has connected to the game server and there is no short connection/synchronization packets are being sent. The following is an example on how to use two tracefiles.

```
./pktgen.py -game q3 -N 5 -IP 192.168.4.23
```

```
    -cap2 2player.dmp -cap3 3player.dmp
```

The amount of packets or duration of 'pktgen' can also be specified. Specifying the hours, minutes, seconds or specifically the number of packets is possible. The following is an example on both specifying the time or the number of packets. If no specification is given then the default of each game is given.

```
./pktgen.py -game q3 -N 5 -IP 192.168.4.23
    -hrs 1 -mins 50 -secs 32
```

```
./pktgen.py -game q3 -N 5 -IP 192.168.4.23
    -pkts 20000
```

B. *pktlen.py*

The ARMA(1,1) process and timeseries prediction is carried out in pktlen.py. This module uses the SciPy [10] and NumPy [11] python modules.

Initially the mean of both time series are removed creating a zero mean timeseries. Once zero meaned, the residuals of the timeseries are calculated using 3. The residual time series are then used to create probability distribution functions (pdfs). The pdfs are then convolved multiple times to acquire the N player residual specified.

After each convolution of PDFs, the result is rotated around to produce a pdf that has its peak in the x axis.

C. *pktsend.py*

The creating of network packets and pcap files is handled in pktsend.py. Pktsent module has features to send live game packets to a specified host address or create PCAP files.

Simple socket programming is used to create live games packets. PCAP files are created using the standard LibPCAP structure [12]

IV. Simluating Traffic Generation

This section describes the C library called 'ArmaPktLen' and how to use it in simulation software such as Omnet++ and Network Simulator. It requires a dependency on the GNU Scientific Library [13]. 'ArmaPktLen' uses static files that hold 2 and 3 player data of multiple first person shooter games located in the 'residuals' folder.

A. *ArmaPktLen.cc*

ArmaPktlen.cc can be used to calculate a packet length using the technique outlined in section II-A and has four functions. Those functions are:

1) write(doube data[], int length, string filename, int offset): Writes an array of type double to a file.

Game	Game Number
Q3	0
Q4	1
HLCS	2
HLDM	3
HL2CS	4
HL2DM	5
ET2PRO	6

TABLE III
Game numbers to use for function arma().

2) write(gsl_complex data[], int length, string filename): Writes an array of type gsl_complex to a file. Real in the first column and complex in the second.

*3) *convolveData(double *series1 , double *series2, double arrayresult[], bool same):* Convolves two series of the same size and place the result in a specified array. If both series are the same series, it must be stated.

4) arma(double N, double game, double debugSwitch): Calculates the mean UDP game packet length (in bytes) depending on the number of players specified. Value can be used to used in a distribution when synthesing game traffic.

B. *Omnet++ Implementation*

To use the C library, Omnet++ [3] and the INET framework [14] is required Assuming a working copy of Omnet++ and INET framework is installed, an Omnet example tarball called omnetArma.tgz can be downloaded from the BITSS project page, compiled and ran.

'ArmaPktlen' C library can be placed in the working directory of your Omnet++ project. The arma() function can be called from the NED files from the UDPApp module. UDPApp.messageLength is what the return value of arma() should be assigned to. At compilation, the compiler will compile all C files in the directoy and create a shared library. To compile the example, `opp_makemake` needs to be run first followed by GNU `make`.

For example:

```
 #opp_makemake -f --deep
-I/usr/local/include -L/usr/local/lib
-lgsl -lgslcblas
-L/pathtoINETsharedLibrary/inetmanet/src
-linet

 #make (or gmake under FreeBSD)
 #./omnetArma
-n "path to all omnet
```

```
module libraries separated by :" $*
```

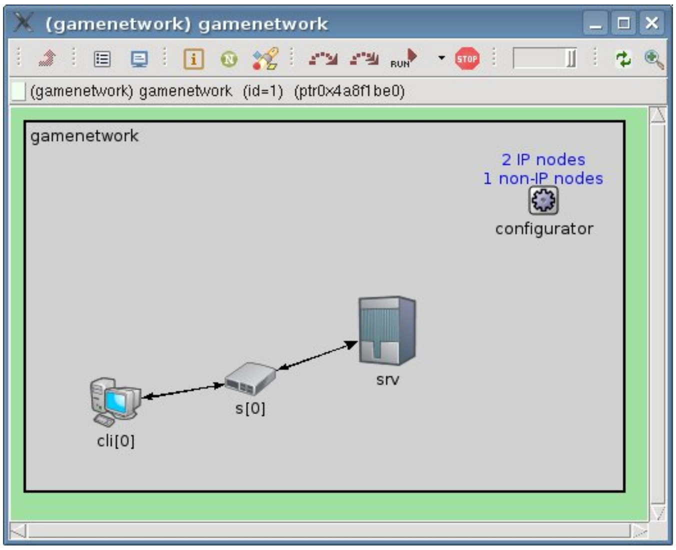

Fig. 8. Omnet++ simulation using the C library called 'omnetArma' to simulate first person shooter packet length using the ARMA and convolution technique.

C. Network Simulator Thoery

Network Simulator [4] uses Otcl for its scripts. 'ArmaPktlen' can be used if NS is recompiled and a new application and agent is created within the source code. Note this has not be attempted but there is documentation on how to do so at [15].

V. Future Work

Having only modeled only server to client game traffic the following also would need to be implemented to complete the packet generator and simulation library.

- Adding client module to simulate client to server traffic.
- Adding a server-client extension.
- Adding traffic models of other game genres.

VI. Conclusion

Live game packet generation and simulation tools have been created to give researchers, ISPs and the industry the ability generate first person shooter (FPS) game network traffic for any N player. This can be used is a variety of experiments and testing which can lead to improving how consumer broadband networks are designed and developed.

References

[1] P. Branch, A. Cricenti, and G. Armitage, "An ARMA(1,1) Prediction Model of First Person Shooter Game Traffic," in *10th IEEE Workshop on Multimedia Signal Processing (MMSP 2008)*, Cairns, Australia, 8-10 October 2008, pp. 736–741. [Online]. Available: http://dx.doi.org/10.1109/MMSP.2008.4665172

[2] A. Cricenti and P. Branch, "ARMA(1,1) Modeling of Quake4 Server to Client Game Traffic," in *6th ACM SIGCOMM workshop on Network and system support for games (Netgames 2007)*, Melbourne, Australia, 19-20 September 2007, pp. 70–74. [Online]. Available: http://doi.acm.org/10.1145/1326257.1326270

[3] "Omnet++ community site," December 2009, accessed 13 Sept 2009. [Online]. Available: http://omnetpp.org

[4] "The network simulator," December 2009, accessed 13 Sept 2009. [Online]. Available: http://www.isi.edu/nsnam/ns/

[5] T. Lang, P. Branch, and G. Armitage, "A Synthetic Traffic Model for Quake 3," in *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, vol. 74, Singapore, June 2004, pp. 233–238. [Online]. Available: http://dx.doi.org/10.1145/1067343.1067373

[6] A. Pavlicic and G. Armitage, "Quake 3 Packet Inter-Arrival and Length Over the Internet," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 030919B, 19 September 2003. [Online]. Available: http://caia.swin.edu.au/reports/030919B/

[7] A. Cricenti, P. Branch, and G. Armitage, "Time-series Modelling of Server to Client IP Packet Length in First Person Shooter Games," in *15th IEEE International Conference on Networks (ICON2007)*, Adelaide, Australia, 19-21 November 2007, pp. 507–512. [Online]. Available: http://dx.doi.org/10.1109/ICON.2007.4444138

[8] "Simulating online networked games database," September 2009, accessed 20 July 2009. [Online]. Available: http://caia.swin.edu.au/sitcrc/song

[9] "Ixia - leader in converged ip testing," 2010, accessed 30 October 2009. [Online]. Available: http://www.ixiacom.com/

[10] "Scipy," December 2009, accessed 13 November 2009. [Online]. Available: http://www.scipy.org/

[11] "Numpy," December 2009, accessed 13 November 2009. [Online]. Available: http://numpy.scipy.org/

[12] "Libpcap file format," December 2008, accessed 02 December 2009. [Online]. Available: http://wiki.wireshark.org/Development/LibpcapFileFormat

[13] "Gnu scientific library," January 2010, accessed October 10 2009. [Online]. Available: http://www.gnu.org/software/gsl/

[14] "Inet framework," June 2009, accessed 13 Sept 2009. [Online]. Available: http://inet.omnetpp.org/

[15] "Add new application and agent," January 2010, accessed October 10 2009. [Online]. Available: http://nile.wpi.edu/NS/new_app_agent.html