2011 UKSim 13th International Conference on Modelling and Simulation

Performance Monitoring of Various Network Traffic Generators

Samad S. Kolahi, Shaneel Narayan, Du. D. T. Nguyen, Yonathan Sunarto
Unitec New Zealand
Carrington Road, Auckland, New Zealand

Abstract—**In this paper, in a laboratory environment, the performance of four network traffic generators (Iperf, Netperf, D-ITG and IP Traffic) are compared. Two computers with Windows operating systems were connected via a 100 Mbps link and for various payload sizes, ranging from 128 Bytes to 1408 Bytes, the TCP traffic on the link was measured using the various monitoring tools mentioned above. The results indicate that these tools can produce significantly different results. In the Windows environment, the bandwidth that the tools measure can vary as much as 16.5 Mbps for a TCP connection over a 100 Mbps link. For the same network set up, Iperf measured the highest bandwidth (93.1 Mbps) while IP traffic the lowest (76.7 Mbps). A comparison of capabilities of traffic generators is also provided.**

***Index Terms*—traffic generator, performance tool, performance analysis.**

I. Introduction

Performance monitoring tools are commonly used to generate traffic and analyse the performance of the networks. There is little work in the literature to compare the performance of these tools. Such a comparison is important as various researchers use different tools to study the same system. But the question is: do these tools produce the same results? The main contribution of this paper is to compare the results of some of the most commonly used network evaluation tools and investigate the performance of TCP in a Windows environment. The tools investigated are Iperf [1], Netperf [2], D-ITG [3], and IP-Traffic [4]. The paper also surveys features of these traffic generating tools.

One previous work in this area was done by Avallone et al.in [3], the authors of D-ITG performance analysis tool. Avallone et al. carried out several experiments to compare their product with some other traffic generators: Mtools [5], Rude & Crude [6], MGEN [7], Iperf [1] and UDP Generator[8]. The authors in [3] connected two Linux machine and monitored the bandwidth of the link using the UDP protocol. To the best of our knowledge, there is no work on comparison of the traffic generator tools on the commonly used TCP protocol in a Windows environment. In this study, the performance of TCP protocol in terms of bandwidth is compared for various traffic generation tools in a Windows platform. Two computers were connected via a 100 Mbps link and for various payload sizes, ranging from 128 Bytes to 1408 Bytes, the traffic on the link was measured using various tools mentioned above.

The rest of the paper is organised as follows: performance monitoring tools used are discussed in Section 2. In section 3, experimental setup is discussed. Section 4 compares the features of the analysis tools. Section 5 reports the experimental results and discusses. Conclusion is in Section 6 followed by the future work in Section 7 and then appendices and references.

II. Performance Monitoring Tools

This section discusses the features of the four traffic generation tools.

A. Iperf

Iperf [1] can be used for evaluation of parameters such as bandwidth, delay, window size, and packet loss. It is used in evaluation of both TCP and UDP traffic. Although, Iperf is a command line performance tool, some developers have used Java to develop a GUI interface for Iperf, call Jperf [9]. This tool is able to run both on Linux and Windows platform with the same command options. Newest version of Iperf, version 1.7, is designed to work with both IPv4 and IPv6.

Iperf has been used by researchers in [10] to study the impact of security protocol on wireless LAN performance; and in [11] to measure the network efficiency of an IPv6 related network.

B. Netperf

Netperf [2] was developed by Hewlett-Packard. This benchmark tool can be used to measure the performance of many different types of networks and it provides tests for throughput, and end-to-end latency. Similar to Iperf, Netperf can be used for both TCP and UDP in either IPv4 or IPv6 networks. This tool can be used for operating systems such as: UNIX (all the major variants), Linux, and Windows. It has two separate executable files: one for server side and other for client side. Netperf has been used in [12] to analyse the TCP performance over the Ethernet LAN in a Windows operating system environment; in [13] to study the wireless security protocol over the mobile IP network; and in [14] for comparison of end systems in IPv6 network.

978-0-7695-4376-5/11 $26.00 © 2011 IEEE
DOI 10.1109/UKSIM.2011.102

C. D-ITG

D-ITG or Distributed Internet Traffic Generator [15] is a platform capable of producing traffic with various packet sizes and a variety of probability distributions: Constant, Uniform, Exponential, Pareto, Cauchy, Normal, Poisson and Gamma. This feature is not available for Iperf and Netperf. D-ITG can monitor various protocols such as: TCP, UDP, ICMP, DNS, Telnet and VoIP. This tool measures throughput, jitters, one-way-delay (OWD), round-trip-time (RTT) and packet loss by using two different separate components called ITG-Send and ITG-Receive. D-ITG is designed to run on both Linux and Windows platform and the newest version is IPv6 compliant. Another GUI version of D-ITG is built by Volker Semken [16] is also available. D-ITG has been used by several researchers to evaluate networks. For example, in a study of IP traffic over interactive data casting systems in [17], voice performance on single radio multi hop IEEE 802.11b systems with chain topology in [18], and in analysing the timing of TCP servers for surviving denial-of-service attacks [19].

D. IP Traffic

IP-Traffic [4] is commercial software developed by ZTI-Telecom. It is a data generation/monitoring/testing tool for IP networks supporting TCP, UDP or ICMP protocols. It can use Microsoft Windows TCP/IP stack (Winsock2 interface) and is independent of any transmission link. IP-Traffic has graphic interface benchmark tool than run on Microsoft platforms such as Windows 98, Windows XP, Windows 2003 and Windows Vista. Like most other performance tools, IP-Traffic requires two separate parts: Traffic-Generator and Traffic-Answering. IP-Traffic has been used by Baghaei and Hunt [20] to study the impact of different wireless securities on the network performance by measuring TCP and UDP throughput with different security levels. Ezedin et al. [21] have used this tool to research the impact of encryption on the throughput of wireless LAN using IEEE 802.11g protocol.

III. NETWORK

A. Network setup

The test network comprises of two computers connected using crossover cable using TCP/IP protocol (IPv4) and Windows network operating systems (Figure 1). The computers come with Intel Pentium 4 with 3.0GHz CPU and 1GB of memory using Network Card Intel Pro/100 Adapter (100 Mbps). Hard drives were Seagate Barracuda 7200 series with 20 GB capacity. They are connected by a crossover cable to avoid any external influence factors such as router processing time. According to Killelea [22], throughput (the amount of data transferred) depends on several conditions over the network like the processor limitations and the hardware designs. To eliminate the effect of these conditions, the research team benchmarked the hardware and similar setup was used for all the tests.

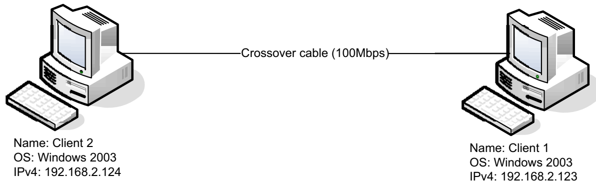

Fig1. Illustration of the laboratory setup

B. Network Parameters

Tables 1 to 4 describe the parameter settings used fot Iperf, Netperf, D-ITG (command line tools), and IP-Traffic (GUI tool). Each tool can have variety of settings by selecting different options. For simplicity, we run Iperf and Netperf tools with default settings and, to match these defaults, do some changes to default setting for D-ITG and IP-Traffic as explained below.

1) Iperf

TABLE 1. IPERF PARAMETERS

Window size of local machine	8K
Window size of remote machine	8K
Running period	10 second
Protocol	TCP
Payload size	From 128 to 1408

2) Netperf

Netperf uses buffer size term instead of window size, all Netperf settings are showed in table 2.

TABLE 2. NETPERF PARAMETERS

Buffer size of local machine	8K
Buffer size of remote machine	8K
Running period	10 second
Protocol	TCP
Payload size	From 128 to 1408

Both Iperf and Netperf are not able to change the payload size by program command lines. Changing payload size was done by a third party tool, Dr.TCP [23]. With Dr.TCP payload size was changed in multiple of 128 Bytes (ranging from 128 Bytes to 1408 Bytes) on both of the two testing machines.

3) D-ITG

The defaults used in D-ITG are: default payload size is 512 Bytes; default protocol is UDP (not TCP); default packet rate is 1,000 packets per second, the default delay parameter is one-way-delay (not round trip time). Unlike, Netperf, in D-ITG the payload can be changed using the command line. Some of the settings in D-ITG were changed to meet the previous settings on Iperf and Netperf tools. The following table describes the parameters used.

TABLE 3. D-ITGPARAMETERS

Packet inter-departure-time (IDT)	Constants IDT
Number of packet sent per second	100,000
Protocol	TCP
Payload size (bytes)	From 128 to 1408

For each of the above tools (Iperf, Netperf, D-ITG) and for each data point in Figure 2, 25 runs were done for each payload size.

4) *IP Traffic*

IP Traffic, a graphic tool, uses the following default settings: payload size is 512 Bytes; the number of packets generated is unlimited; and it has no round trip time option. To match the settings of other tools, we changed some settings in IP Traffic as shows in table 4 below.

TABLE 4. IP TRAFFIC PARAMETERS

Number of packet generate	1,000,000
Packet contents	Fix contents
Payload size (bytes)	From 128 to 1408
Inter packet delay	0 ms

Similar to D-ITG, IP Traffic can generate packets with different size payloads. For IP-Traffic, each data point was for a total of 25 runs.

IV. TOOLS COMPARISONS

In this section, we evaluate the four tools mentioned above and include the experiences we obtained during the experiments. Each tool has it own advantage and disadvantage. Table 5 in appendix A displays the summary of all tools. In general, four traffic generators can work with various protocols. All four traffic generators considered are able to evaluate TCP and UDP and all can use IPv4 or IPv6. However, we had some difficulty running D-ITG with IPv6. In addition to these protocols, D-ITG can measure and test many other protocols such as Telnet, VoIP, DNS and ICMP. Netperf also supports SCTP (Stream Control Transmission Protocol) and DLPI (Data Link Provider Interface). IP-Traffic can evaluate IGMP (Internet Group Multicast Protocol). Iperf does not measure any other protocol apart from TCP and UDP protocols.

Throughput, packet loss, jitters and round trip time are common metrics used in all tools considered. Some of the tools can measure additional parameters, for examples: Netperf can measure CPU utilization; and D-ITG can measures one-way-delay.

Unfortunately, not all tools support generating traffic with various probability distribution functions. D-ITG and IP-Traffic can support probability distribution functions for various packet sizes and packet inter-arrival times. Both of these can support distributions like Pareto, Exponential, Poisson and Gamma distributions. Iperf and Netperf do not support various traffic probability distributions.

Another important factor of a traffic generator is platforms supported. Most performance tools are designed to run on UNIX/Linux platform. Three of the tools evaluated in this paper (Iperf, Netperf and D-ITG) can work on both Windows and Linux platforms while IP-Traffic mainly works on Windows platform only.

Among the four evaluated tools, D-ITG and IP-Traffic support packet delay while Iperf and Netperf do not and no related information is provided in their manuals.

To record the results of the measurements for analysis purpose, a log file needed to be kept. Log file can be in a text file format which can be read by any text editor application or in other formats such as spreadsheet. Only D-ITG, Iperf and IP-Traffic tool provide the log mechanism. Log file of D-ITG is a text file and log file can be stored on local machine (both sending traffic machine and receiving machine) or on another third machine (D-ITG call this machine a log server). IP-Traffic log file format is in CSV format which can be opened with Microsoft Excel or some others spreadsheet application. Iperf, like D-ITG, supplies a log system in a text file and stores locally on the machine running Iperf. Netperf does not provide any log file. In our experiments, we used a DOS function command to re-direct the Netperf results on screen to a text file for later analysis.

We found that three of tools are IPv6 compliant. Iperf and Netperf and IP-Traffic were tested and show it fully compatible with IPv6 in Windows environment.

To produce the data points, the experiments needed to run many times and the results averaged. For repetitive runs, we developed a batch file to repeat the commands for the command line tools (Iperf and Netperf). Both Iperf and Netperf work well with DOS batch file while D-ITG and IP-Traffic do not. In the experiments, we run D-ITG manually for all of the tests. IP-Traffic, a graphic user interface cannot be used in conjunction with a batch file, therefore IP-Traffic was run manually for all of the tests. However, we generally found that all the traffic generators involved in this paper were user friendly and easy to use; and all tools have useful online manuals that instruct the users how to use the software. Moreover, Iperf and Netperf have online community which include many global users. D-ITG and IP-Traffic, on the other hands, do not have an online community.

V. RESULTS

The tools were run with the same settings as discussed in section 3. Traffic was generated by tools for payload sizes in multiples of 128 Bytes (ranging from 128 Bytes to 1408 Bytes) and throughput was measured. For each payload, 25 runs were performed and the results averaged and variance of the results calculated (Appendix B). The comparison result for four different performance monitoring is in Figure 2.

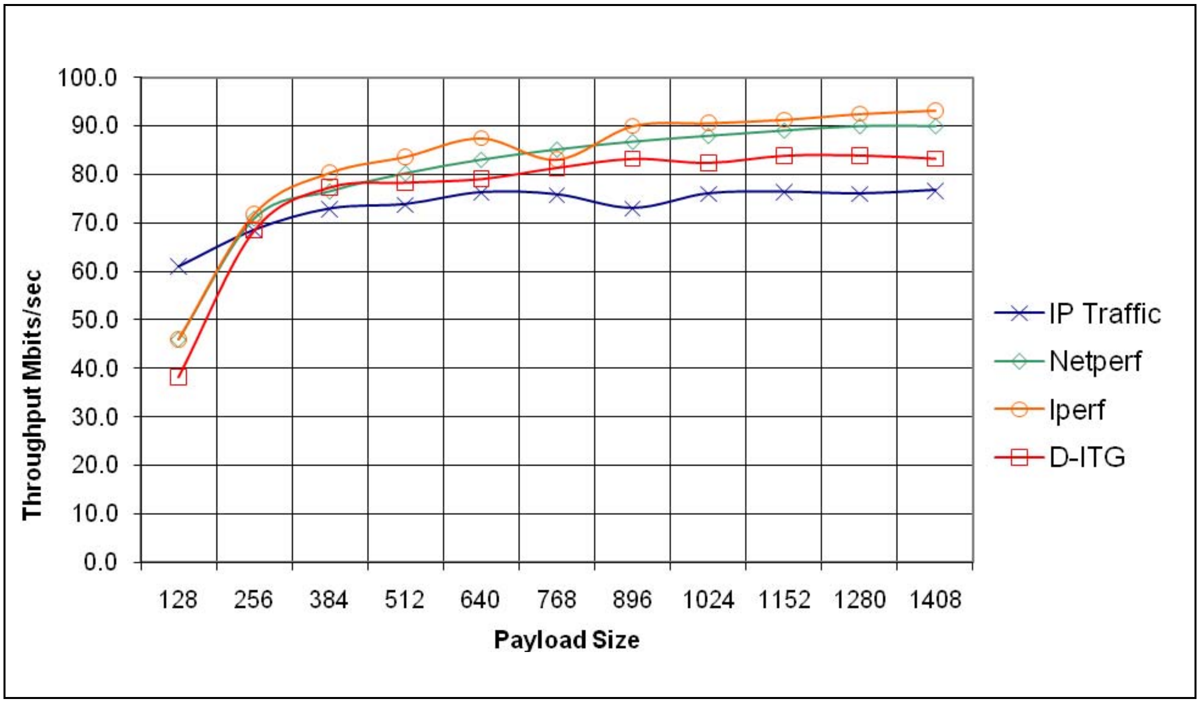

Fig2. TCP Throughput comparison in different tools

The comparison of results in Figure 2 indicates that throughput reduces while changing the software from Iperf (highest throughput), to Netperf, D-ITG, and IP-traffic (lowest). At payload size of 1408 Bytes, the bandwidths are 93.1 Mbps (Iperf), 89.9 Mbps (Netperf), 83.1 Mbps (D-ITG), and 76.6 Mbps (IP-Traffic). This means that changing the performance evaluation tool, can significantly affect the bandwidth as much as 16.5 Mbps for the system studied. Iperf indicates an unusual point when it drops the throughput below Netperf at 768 Bytes point while at 128 Bytes point IP-Traffic has an unusual point where it gives higher throughput than all other software (15 Mbps more that Iperf and Netperf and 22.9 Mbps more than D-ITG). At this payload size (128 Bytes), both Netperf and Iperf give the same result of 46 Mbps.

For all performance tools considered, the throughput increases when changing the payload size form 128 Bytes to 384 Bytes after which it appears that increasing the payload size (from 384 to 1408 Bytes) does not have the same significant effect. Low throughput for small payload sizes is because in small packet sizes, the percentage of overhead to payload size is very high that causes bandwidth to be wasted. In addition small payload sizes will cause the ACK flood (and more delay) on network because TCP requires an ACK.

Increasing the payload size from 128 Bytes to 384 Bytes increases the throughput by 11.9 Mbps for IP-Traffic, 30.5 Mbps for Netperf, 34.3 Mbps for Iperf and 39.1 Mbps for D-ITG. It seems at low payload sizes of between 128 Bytes and 384 Bytes; changing payload size does not have much impact on throughput in IP-Traffic while other tools show good sensitivity. However, the statistics are different for large payload sizes. Increasing the payload size from 512 Bytes to 1408 Bytes changes the throughput by 2.9 Mbps for IP-Traffic, 9.7 Mbps for Netperf, 9.5 Mbps for Iperf, and 4.9 Mbps for D-ITG. The data shows that IP-Traffic followed by D-ITG results is not sensitive to increasing payload sizes between 512 Bytes and 1408 Bytes, while Netperf and Iperf are more sensitive to payload sizes variations.

The variance of data points (Appendix B) for IP-Traffic tool was between 0.25 and 4.0 for various runs with the average of 1.25, Netperf was between 0.00 to 0.09 with average of 0.03, Iperf was between 0.16 to 1.51 with average of 0.56, and D-ITG was between 0.64 and 1.96 with average of 1.24. The results indicate that IP-Traffic has higher variance and therefore higher variation of results between different runs while Netperf has the lowest.

Results in [3] indicate that in a Linux environment and when measuring UDP traffic for payload size of 1024 Bytes the difference between D-ITG and Iperf measurements is 5% (D-ITG producing the higher bandwidth.) Our experiment, using TCP traffic in Windows environment, the percentage difference between D-ITG and Iperf is 9% (D-ITG producing the lower bandwidth.) Therefore, we observed that D-ITG generate higher UDP traffic than Iperf under Linux 5 environment while in Windows platform and TCP, Iperf measures higher traffic than D-ITG. Note that UDP measurements used the first version (version 0.1) of D-ITG while TCP protocol measurements were added in later version of D-ITG (version 0.2).

VI. CONCLUSIONS

In this paper, we studied the performance comparisons of four different monitoring tools in the Windows operating system. For packet size of 1408, Iperf showed the highest throughput (93.1 Mbps) while IP Traffic was the lowest throughput (76.6 Mbps) of the four tools. However, this was not the case for packet sizes of 128 Bytes and 256 Bytes. At 128 Bytes, IP-Traffic measured highest bandwidth (61 Mbps) while D-ITG measured the lowest (38.1 Mbps). At 256 Bytes the tools provided the same results. This paper also compared different features of the monitoring tools compared.

VII. FUTURE WORK

An extension of this work can comparing using more performance monitoring tools using both TCP and UDP with IPv6. Further work could include deciding which traffic generator actually provides the best results.

APPENDICES

A. Comparison tools

TABLE 5. COMPARISON TOOLS

	Iperf	Netperf	D-ITG	IP Traffic
Interface	Command line	Command line	Command line	GUI interface
Multi-platform	Yes	Yes	Yes	Windows only
User guide	Yes	Yes	Yes	Yes
Protocols	TCP and UDP	TCP, UDP SCTP, DLPI	TCP, UDP, ICMP, DNS, Telnet, VoIP	TCP, UDP IGMP
Packet departure Packet delay	No	No	Yes	Yes
Probability distributions	No	No	Yes	Yes
Log file	Yes	No	Yes	Yes
Internet Protocol	IPv6 and IPv4	IPv6 and IPv4	IPv6 and IPv4	IPv6 and IPv4
Measurements metrics	Jitter Packet loss Throughput	Packet loss Throughput Response time CPU usage	One-way-delay Round-trip-time Packet loss Jitter Throughput	Throughput Round trip time Packet loss Jitters

B. Summary results table

TABLE 6. SUMMARY THROUGHPUT RESULTS TABLE WITH VARIANCE

Packet Size	IP Traffic	Variance	Netperf	Variance	Iperf	Variance	D-ITG	Variance
128	61.0	4.0	46.0	0.04	46.0	1.51	38.1	1.44
256	68.5	1.69	70.9	0.00	71.8	0.25	68.3	1.69
384	72.9	1.44	76.5	0.00	80.3	0.36	77.2	1.21
512	73.8	0.49	80.2	0.00	83.6	0.16	78.2	1.44
640	76.2	0.64	83.0	0.04	87.4	0.16	78.9	1.21
768	75.8	0.64	85.1	0.00	83.0	0.64	81.2	0.81
896	73.0	1.96	86.7	0.00	89.9	0.81	83.1	0.81
1024	76.0	0.25	87.9	0.01	90.5	0.49	82.2	0.64
1152	76.3	0.25	89.0	0.09	91.2	0.36	83.7	1.00
1280	76.0	1.96	89.9	0.09	92.4	0.64	83.8	1.96
1408	76.7	0.49	89.9	0.09	93.1	0.81	83.1	1.44

REFERENCES

[1] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf," http://dast.nlanr.net/Projects/Iperf/.

[2] R. Jones, "Netperf 2.4.3," http://www.netperf.org/netperf/.

[3] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, "D-ITG distributed Internet traffic generator", First International Conference on the Quantitative Evaluation of Systems, 2004.

[4] ZTI Telecom, "IP Traffic - test & measure," http://www.zti-telecom.com.

[5] S. Avallone, "Mtools 1.1," http://www.grid.unina.it/grid/mtools/, 2002.

[6] J. Laine, S. Saaristo, and R. Prior, "Rude & crude," http://rude.sourceforge.net/.

[9] T. Lattner, D. Cook, and K. Gibbs, "Jperf,"http://dast.nlanr.net/projects/jperf/.

[10] A. K. Agarwal and W. Wang, "Measuring performance impact of security protocols in wireless local area networks," 2nd International Conference on Broadband Networks, 2005.

[11] T.-Y. Wu, H.-C. Chao, T.-G. Tsuei, and Y.-F. Li, "A measurement study of network efficiency for TWAREN IPv6 backbone," International Journal of Network Management, vol. 15, pp. 411-419, 2005.

[12] K. A. Gotsis, S. K. Goudos, and J. N. Sahalos, "A test lab for the performance analysis of TCP over Ethernet LAN on windows operating system," *IEEE* Transactions on Education, vol. 48, pp. 318-328, 2005.

[13] A. K. Agarwal, J. S. Gill, and W. Wenye, "An experimental study on wireless security protocols over mobile IP networks," IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall.

[14] S. Zeadally, R. Wasseem, and I. Raicu, "Comparison of end-system IPv6 protocol stacks*," IEE Proceedings Communications*, vol. 151, pp. 238-242, 2004.

[15] A. Botta, A. Dainotti, and A. Pescapè, "Multiprotocol and multi-platform traffic generation and measurement," INFOCOM 2007 DEMO Session., Anchorage, Alaska, 2007.

[16] V. Semken, "Graphical user interface for D-ITG 2.4," http://www.semken.com/projekte/index.html.

[7] Naval Research Laboratory, "Multi-Generator (MGEN)," http://cs.itd.nrl.navy.mil/work/mgen/index.php.

[8] The University of Michigan, "gen_send, gen_recv: a simple UDP traffic generator application," http://www.citi.umich.edu/projects/qbone/generator.html.

[17] L. Wei, L. Hong, and G. Gagnon, "Performance assessment of IP traffic over ATSC interactive datacasting systems," IEEE Transactions on Consumer Electronics, vol. 51, pp. 54-62, 2005.

[18] T. Kee Ngoh, K. Yin Fern, and S. Moh Lim, "Voice performance study on single radio multihop IEEE 802.11b systems with chain topology," 13th IEEE International Conference on Networks, 2005.

[19] V. Krishna Nandivada and J. Palsberg, "Timing analysis of TCP servers for surviving denial-of service attacks," 11th IEEE Real Time and Embedded Technology and Applications Symposium, 2005.

[20] N. Baghaei and R. Hunt, "IEEE 802.11 wireless LAN security performance using multiple clients", The 12th IEEE International Conference on Networks, 2004. (ICON 2004).

[21] B. Ezedin, B. Mohammed, A. Amal, S. Hanadi Al, K. Huda, and M. Meera Al, "Impact of Security on the Performance of Wireless-Local Area Networks," Innovations in Information Technology, 2006.

[22] P. Killelea, "Web Performance Tuning," http://www.amazon.ca/Web-Performance-Tuning-Patrick-Killelea/dp/product-description/059600172X.

[23] T. Orkun, "Dr.TCP 0.21," http://www.dslreports.com/drtcp.