Resilient Internet Access Using Tunnel-Based Solution for Malicious ISP Blocking

Marwan Abu-Amara, Mohammed A.K. Asif, Mohammed H. Sqalli, Ashraf Mahmoud, Farag Azzedin*
Computer Engineering Department, *Information and Computer Science Department
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
{marwan, khadir, sqalli, ashraf, fazzedin}@kfupm.edu.sa

***Abstract*—A tunnel-based solution is presented in this paper to provide a resilient Internet access in face of a malicious act of denial of Internet access by higher-tier Internet service providers (ISPs). The proposed solution describes the different types of tunneling protocols that can be used, and the needed configurations to establish the tunnels. The validity of the proposed solution is demonstrated by means of network simulations using OPNET. Furthermore, the proposed solution is evaluated using different types of traffic generated by well known applications such as file transfer protocol (FTP) and video conferencing. We also considered different tunneling protocols such as IP-in-IP, generic routing encapsulation (GRE), and GRE with checksum under high traffic load. Based on the simulation results, the IP-in-IP tunneling protocol performs the best among all protocols considered in this work.**

Keywords-malicious ISP, intentional Internet access denial, resilient Internet, tunneling protocol, OPNET

I. INTRODUCTION

The Internet has become one of the most important means for reachability and communication with others. The number of people using the Internet in June 2010 exceeded 1.96 billion users [1]. The Internet is a huge network of hundreds of millions of nodes that exchange information with each other. Because of the dynamic nature and very large size of the network, routing of data takes place on two hierarchical levels. At a lower level, a group of devices are controlled by a single administrative autonomy that has a complete view of its own network and is responsible for the routing of Internet traffic within its boundaries (intra-domain routing) [2]. The network controlled by a single administrative autonomy is referred to as an autonomous system (AS). At a higher level, routing between different ASes (inter-domain routing) takes place by means of the Border Gateway Protocol (BGP). Moreover, most ASes are operated by Internet Service Providers (ISPs). ISPs are loosely classified into 3 tiers, with higher-tier ISPs forming the Internet core and lower-tier ISPs providing Internet service to end-users.

Access to the Internet can be disrupted by either non-malicious or malicious causes including hardware and/or software failures, denial of service attacks, and deliberate Internet access denial by higher-tier ISPs. Also, the Internet access disruption can occur at three levels: application, routing, and physical [3].

The intentional Internet access denial can occur due to the fact that BGP does not permit an AS to fully control how its traffic is routed to or from distant ASes [4]. Thus, if some traffic traverses a path that passes through a malicious ISP, then this traffic can be intentionally dropped by the malicious ISP. Accordingly, a higher-tier ISP can utilize BGP to maliciously block routing information, and to filter out traffic from and to a particular AS.

The term Malicious Service Provider (MSP) has been used in the literature to describe a malicious activity by a service provider to hijack Internet prefixes [5]. For example, 106,089 prefixes were hijacked in December 2004 by AS 9121 [6]. Similarly, prefix 64.233.161.0/24 which includes IP addresses for Google was hijacked in May 2005 by AS 174 [7]. Such incidents can expose sensitive traffic sent by users to the hijacker, allowing the hijacker to drop, record, and/or modify the contents of the intercepted traffic.

The concept of a malicious ISP is realistic even though ISPs are supposed to provide the promised service to their customers for fear of losing them, and, ultimately, jeopardizing their reputations. However, there are many reasons that may force a higher-tier ISP to become malicious and perform an Internet access denial to a specific region. For example, Internet access denial can be driven by political motivations, as governments may attempt to establish an Internet embargo on a targeted region by forcing ISPs to block Internet access to that specific region. Many large services and networks have been attacked recently for political motivations. As an example, on December 2009, Gmail had many attacks targeting email accounts of Chinese human rights activists [8]. Similarly, Twitter has also been attacked during 2009 by hackers from Iran [9]. Another prime example of political motivations of a service provider to deny Internet access to an organization are the recent attempts by many governments to pressure service providers to deny access to WikiLeaks [10]. Likewise, ISPs' routers may be hacked by attackers and reconfigured to drop traffic, causing Internet access denial.

This paper addresses the problem of intentional Internet access denial by higher-tier ISPs that occurs at the routing level. Section II presents the related work, while section III includes the problem description and the proposed tunnel-based solution. Section IV discusses the validity of the solution, while in section V the effect of the proposed solution on the network performance is evaluated through simulations. The paper is concluded in section VI.

978-1-61284-486-2/11/$26.00 ©2011 IEEE

II. RELATED WORK

The major concern about BGP security is that malicious BGP routers can arbitrarily falsify BGP routing messages and spread incorrect routing information [15]. BGP routers exchange routing information via UPDATE messages. Serious problems to routing in the Internet may occur when incorrect UPDATE messages are exchanged, maliciously or non-maliciously, between routers. For example, the May 2005 prefix hijacking that caused Google to go down was a direct result of an exchange of incorrect UPDATE messages between routers of AS 174 and routers of other ASes [7]. Similarly, in June 2006, several prefixes were hijacked by AS 23520 [11]. As stated earlier, another serious concern about BGP is that BGP does not allow an AS to control exactly how its traffic is routed to other destinations [4]. Thus, Internet access denial can take place if the packets are routed by BGP through a malicious ISP that drops these packets. Hence, the Internet access denial can be prevented by controlling the traffic path so that the traffic does not pass through the malicious ISP, or by preventing the traffic from being dropped at the malicious ISP by concealing the traffic identity.

A modification of BGP is needed in all routers in the Internet in order to control the traffic path so that traffic is not sent through the malicious ISP. Accordingly, Quoitin et al. [12][13] proposed BGP tuning techniques to influence the path selection process of remote ASes. Alrefai [14] enhanced the BGP tuning techniques proposed by Quoitin et al. to achieve better scalability results.

On the other hand, the occurrence of an Internet access denial can be avoided by using techniques to conceal the traffic's origin or destination from the malicious ISP. Such techniques use IP addresses that are different from the blocked ones. Therefore, the malicious ISP will be mislead into routing the traffic without filtering it.

Accordingly, tunneling protocols can be used to hide the identity of the traffic by encapsulating the traffic, and carrying the encapsulated traffic through a tunnel created between the two tunnel endpoints. Hence, the intermediate routers will only see the two tunnel ends as the source and destination addresses. Examples of tunneling protocols include IP-in-IP [15], Internet Protocol Security (IPSec) [16] and generic routing encapsulation (GRE) [17].

Such usage of tunneling protocols is common to achieve better Internet resiliency. For example, Kini et al. [18] used IP-in-IP to enhance the robustness of a network to dual link failures. Similarly, Wu et al. [19] considered a failure scenario that breaks an AS into two or more isolated parts and disrupt the connectivity among these AS partitions. Wu et al. proposed the use of tunneling techniques by the neighbors of the affected AS partitions to provide extra connectivity to bypass the failure. Thus, through the use of tunneling techniques, the AS partitions can communicate with each other.

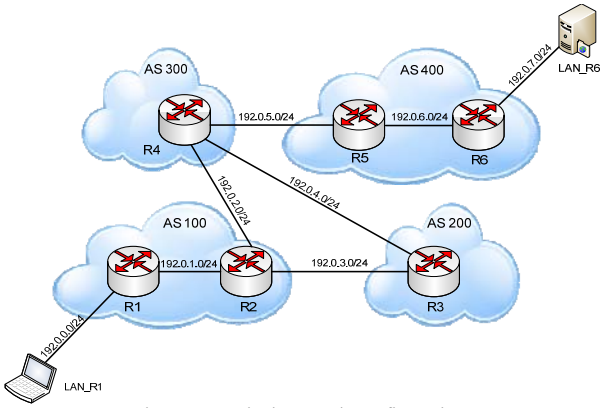

Figure 1. Typical network configuration.

III. PROPLEM DESCRIPTION AND PROPOSED SOLUTION

A. *Problem Description*

The paper addresses the problem of the Internet access denial due to the malicious act by higher-tier ISPs. Furthermore, in the problem considered we only address the routing level of this problem.

To illustrate the problem further, Figure 1 provides a typical network configuration that consists of four ASes. AS100 represents the local AS that is targeted by the malicious ISP, AS200 represents a neighboring AS, AS300 is the malicious ISP and connects both AS100 and AS200 to other ASes in the Internet, and AS400 is a possible destination AS for AS100. The normal behavior of the Internet when the higher-tier ISP (i.e., AS300) is non-malicious is for AS100 to communicate normally with AS400 through AS300.

The problem considered by this paper assumes that the higher-tier ISP (i.e., AS300) starts to act maliciously towards the local AS (i.e., AS100) by denying Internet access to it. In the meantime, the higher-tier ISP continues to provide Internet access to other ASes such as AS200 and AS400. The problem is then how to provide Internet access to the local AS, even after the higher-tier ISP has denied Internet access to it.

B. *Proposed Tunnel-Based Solution*

The proposed solution to the Internet access denial by higher-tier ISPs is based on the use of tunneling protocols. To implement this solution, available tunneling protocols like IP-in-IP and GRE can be utilized. A tunnel is created from the local AS to a destination AS only if the normal path to the destination AS passes through the malicious ISP. For the proper establishment of the tunnel, the solution assumes the presence of at least one cooperating AS that precedes the malicious ISP on the tunnel path, and at least another cooperating AS that follows the malicious ISP on the tunnel path. As a result of creating the tunnel, the malicious ISP can be bypassed.

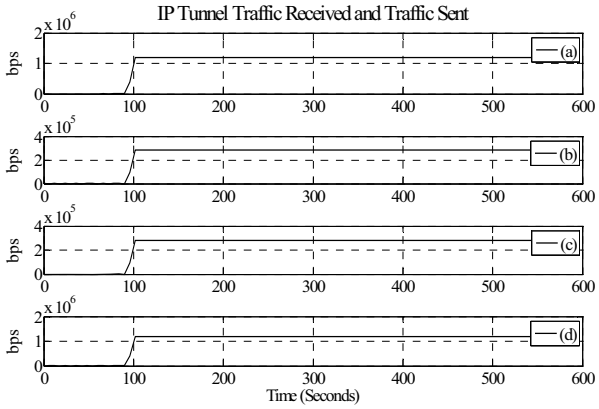

Figure 2. IP tunnel traffic received and sent by routers R2 and R5.

IV. VALIDITY OF THE SOLUTION

The proposed solution is validated through simulations using the OPNET network simulator [20]. In the simulation setup, the baseline network shown in Figure 1 was used. In addition, a tunnel between R2 (from the blocked AS) and R5 (from the distant AS) was created. The created tunnel passes through R3 and R4. The non-blocked IP address provided by the neighboring AS (i.e., AS200) was used to create the tunnel. Thus, with the help of a neighboring AS, a tunnel that passes through the malicious ISP (i.e., AS300) was created. The use of a non-blocked IP address will prevent the malicious router (i.e., router R4) from dropping incoming and outgoing traffic to and from the affected AS.

To create a tunnel, we need a prefix to be used for the tunnel interface. In the simulation, the chosen prefix belongs to subnet 200.0.0.0/24 (i.e., AS200). The tunnel starting point IP address is 200.0.0.1, and the tunnel ending point IP address is 200.0.0.2. The routing protocol used for the tunnel interface is OSPF.

To validate that the proposed solution is setup to forward the traffic properly through the tunnel, we can first examine the IP forwarding table on both routers R2 and R5. From the tables we can determine that the incoming and the outgoing traffic on router R2 and router R5 use the created tunnel. Furthermore, Figure 2(a) and Figure 2(b) show the IP tunnel traffic received, in bits per second (bps), at router R2, and the IP tunnel traffic sent, in bps, by router R2, respectively. In contrast, Figure 2(c) and Figure 2(d) the IP tunnel traffic received, in bps, at router R5, and the IP tunnel traffic sent, in bps, by router R5, respectively. This validates the proper setup and operation of the tunnel, and the proposed solution.

V. PERFORMANCE EVALUATION AND RESULTS

When tunneling protocols are used, extra overhead bytes are added to the packets entering the tunnel as compared to normal packets. The amount of extra overhead bytes that will be added depends on the type of the tunneling protocol used. The tunneling protocols IP-in-IP, GRE, and GRE with check sum add 20, 24, and 28 of extra overhead bytes, respectively, to each packet entering the tunnel. As such, an impact on the network performance is expected as a result of using tunneling protocols in the proposed solution. Thus, it is important to evaluate the effect of the proposed tunnel-based solution on the network performance. Hence, in this section we compare the network performance when the tunnel-based solution is used, to the normal operation (i.e., no malicious activity by the ISP) in terms of end-to-end delay, and traffic throughput overhead under different types of traffic and with 75% network load. The performance evaluation is conducted by means of OPNET simulations.

A. Simulation Setup

The network model shown in Figure 1 is used for the performance evaluation. The local and remote local area networks (i.e., LAN_R1 and LAN_R6) are set to 100 Mbps Fast Ethernet networks. The gateway routers are based on the generic router model in OPNET that supports BGP, OSPF, and tunneling. All routers are interconnected to each other as shown in Figure 1 using DS-1 links, providing a data rate of 1.544 Mbps. The total network simulation duration is set to 600 seconds. Moreover, several simulation scenarios are considered by varying the type of tunneling protocol, the type of traffic, and with 75% network load. The tunneling protocols considered in the simulations are IP-in-IP, GRE, and GRE with checksum. Furthermore, each simulation scenario is repeated for 5 different seeds/runs, and the average of the 5 results is reported. The performance statistics collected are the end-to-end delay and the traffic throughput overhead.

The solution is evaluated for different types of traffic generated by well known applications such as file transfer protocol (FTP) and video conferencing. FTP represents a network application that runs over TCP, whereas video conferencing represents a network application that runs over UDP. Each simulation is run with 75% of the available link bandwidth (i.e., 1,158 kbps).

B. Performance Metrics and Results

Two performance metrics are investigated in the evaluation. The first metric is the end-to-end delay, measured in seconds, at LAN_R1. The second metric is the throughput, measured in bits per second, at the link between R5 to R4. The R5 to R4 link is selected because the tunnel overhead can be examined at this link with respect to each tunneling protocol.

C. Simulation for End-to-End Delay

The end-to-end delay refers to the duration of time that a packet takes to travel from the client to the server. The end-to-end delay includes the transmission time, the propagation time, and the queuing delay. For the purpose of the simulation, the FTP application is simulated with a file size of 50KB, and the video conferencing application uses the OPNET built in file size of 1172 bytes.

To achieve the desired traffic load of 75% for FTP and video conferencing on the links, the number of users used is set to 30 users. In the case of FTP, both the file size in bytes and the inter-request time in seconds are constant. Similarly, for video conferencing both the frame size in bytes and the frame inter-arrival time in seconds are constant.

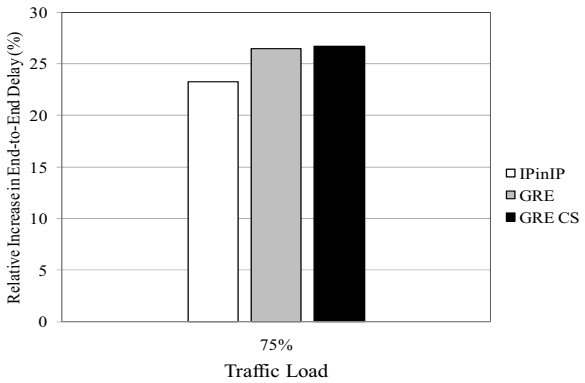

Figure 3. FTP relative increase in end-to-end delay.

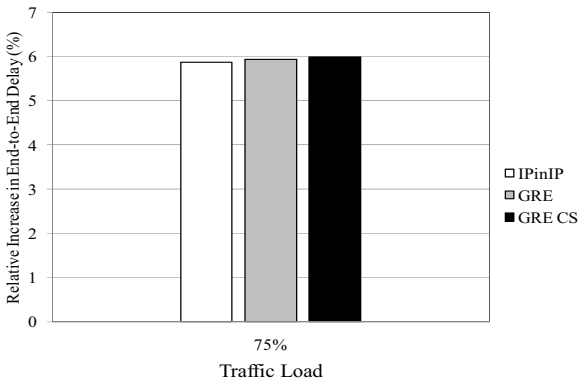

Figure 5. Video conferencing relative increase in end-to-end delay.

FTP is simulated as requests to download a file from the server in LAN_R6. The results for the relative increase in the end-to-end delay, which is computed as $(Delay_{Tunnel} - Delay_{NoTunnel})/Delay_{NoTunnel}$, are shown in Figure 3.

The relative increase in the end-to-end delay is caused by the introduction of the tunnel. Furthermore, we note that the IP-in-IP tunneling protocol has the least relative increase in the end-to-end delay. The observation is justified by noting that IP-in-IP tunneling protocol adds the least amount of overhead among the tunneling protocols considered, and therefore, produces the least amount of fragmentation. Although the increase in the relative end-to-end delay is around 25%, the absolute increase in the end-to-end delay shown in Figure 4, and computed as $(Delay_{Tunnel}) - (Delay_{NoTunnel})$, is less than 1.5 ms, which is considered to be negligible for FTP.

For the video conferencing scenario, the end-to-end delay results show similar behavior to the results obtained for FTP. The relative increase in the end-to-end delay is shown in Figure 5, and as evident it shows a lower relative increase in the end-to-end delay than the results for FTP. This is mainly due to the fact that video conferencing runs over UDP which has a considerably smaller header than the TCP header, over which FTP runs. Also, UDP is a connectionless protocol that does not wait for acknowledgements as in the case of TCP.

Figure 6 shows the absolute amount of increase in the end-to-end delay caused by the introduction of the tunnel. Similar to FTP, we see that the IP-in-IP tunneling protocol has the least amount of increase in the end-to-end delay. Although the increase in the relative end-to-end delay is around 6%, the absolute increase in the end-to-end delay shown in Figure 6 is less than 7 ms, which is considered to be insignificant for video conferencing.

D. Simulation of Traffic Throughput Overhead

The traffic throughput overhead, in bits per seconds, measures the amount of overhead bits added to each packet entering the tunnel. The simulation is set to measure the traffic throughput overhead at the link R5 to R4 where the tunnel starts.

The results for both the FTP and the video conferencing applications are shown in Figure 7 and Figure 8. The relative increase in the throughput overhead is shown in Figure 7, whereas the absolute overhead of the tunnel is shown in Figure 8.

It is obvious from Figure 7 that the relative increase in the traffic throughput overhead for video conferencing is lower than the relative increase in the throughput for FTP for the same reason stated earlier for the end-to-end delay. Likewise, It can be observed from Figure 7 that the IP-in-IP tunneling protocol has the least amount of relative increase in traffic throughput overhead as it adds the smallest header size among the other tunneling protocols.

Furthermore, we have shown in Figure 8 the amount of absolute traffic throughput overhead that is caused by the introduction of the tunnel. In both the FTP and the video conferencing scenarios, the amount of absolute traffic

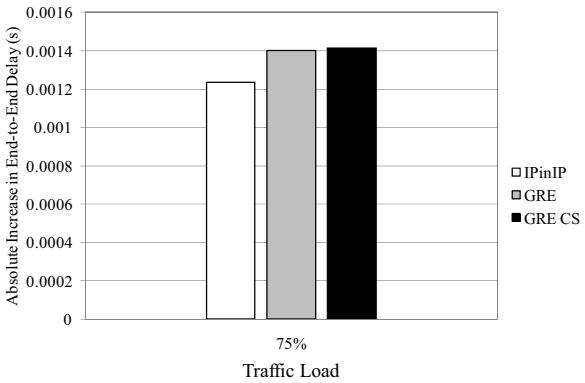

Figure 4. FTP absolute increase in end-to-end delay.

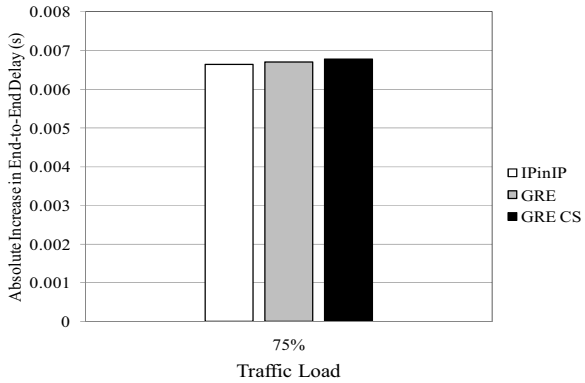

Figure 6. Video conferencing absolute increase in end-to-end delay.

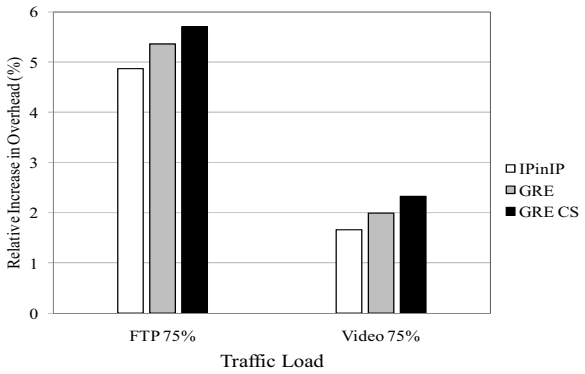

Figure 7. Relative increase in throughput overhead.

throughput overhead is considered to be negligible relative to the overall throughput of about 1,158 kbps.

In conclusion, UDP-based traffic has the least amount of relative increase in the tunnel overhead when compared to the TCP-based traffic. To further make the comparison fair, an experiment was conducted with a small file size for the FTP application. The results of the experiment confirmed that the UDP-based traffic incurs less amount of tunnel overhead than the TCP-based traffic. As stated earlier, this is mostly attributed to the differences between UDP and TCP in header size and connection type.

VI. CONCLUSION

In this paper, we proposed a tunnel-based solution to overcome the malicious act of Internet access denial by a higher-tier ISP. The proposed solution was validated using the OPNET network simulator. Moreover, the proposed solution was then evaluated by means of OPNET simulations to quantify the effect of the tunneling protocols used by the proposed solution on the network performance. The results showed that the effect is negligible, although performance degradation is more evident in TCP-based applications than in UDP-based applications. Furthermore, and as expected, it was found that IP-in-IP tunneling protocol outperformed the other tunneling protocols considered in this study.

ACKNOWLEDGMENT

The authors acknowledge the support provided by King Fahd University of Petroleum and Minerals (KFUPM). This project is funded by King Abdulaziz City for Science and Technology (KACST) under the National Science, Technology, and Innovation Plan (project No. 08-INF97-4).

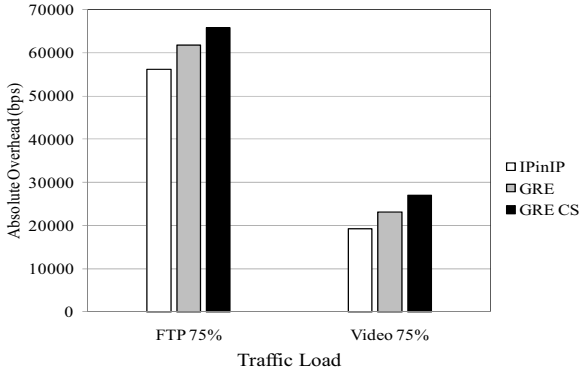

Figure 8. Absolute throughput overhead.

REFERENCES

[1] Neilsen, Internet World Stats, http://www.internetworldstats.com, World Wide Web electronic publication, 2010.

[2] Matthew Caesar and Jennifer Rexford. "BGP routing policies in ISP networks," IEEE Communications Society, Volume 19, Issue 6, pp. 5-11, November-December 2005.

[3] M. Abu-Amara, A. Mahmoud, F. Azzedin, and M. Sqalli, "Internet access denial by international Internet service providers: analysis and counter measures," Research Proposal, April 2008.

[4] K. Butler, T. Farley, P. McDaniel, and J. Rexford, "A survey of BGP security issues and solutions," Proceedings of the IEEE, vol. 98, pp. 100-122, Jan. 2010.

[5] N. Wang, Y. Zhi, and B. Wang, "AT: an origin verification mechanism based on assignment track for securing BGP," Proceedings of the 2008 IEEE International Conference on Communications, Beijing, pp. 5739-5745, 19-23 May 2008.

[6] A. Popescu, B. Premore, and T. Underwood, "Anatomy of a leak: AS9121," 34th North American Network Operators' Group meeting, Seattle, 15-17 May 2005.

[7] J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: improving BGP by cautiously adopting routes," Proceedings of the 2006 14th IEEE International Conference on Network Protocols, Santa Barbara, pp.290-299, 12-15 Nov. 2006.

[8] D. Drummond, "A new approach to china," The Official Google Blog, http://googleblog.blogspot.com/2010/01/new-approach-to-china.html, Jan. 2010.

[9] J. Finkle and D. Bartz, "Twitter hacked, attacker claims Iran link," Reuters, http://www.reuters.com/article/idUSTRE5BH2A620091218, December, 2009.

[10] "WikiLeaks," Wikipedia, http://en.wikipedia.org/wiki/WikiLeaks#cite_note-197, 2011.

[11] J. Karlin, "A fun hijack: 1/8, 2/8, 3/8, 4/8, 5/8, 7/8, 8/8, 12/8 briefly announced by AS 23520," http://www.merit.edu/mail.archives/nanog/2006-06/msg00082.html.

[12] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain traffic engineering with BGP," IEEE Communications Magazine, vol. 41, no. 5, pp. 122-128, May 2003.

[13] B. Quoitin and O. Bonaventure, "A cooperative approach to interdomain traffic engineering," Proceedings of Next Generation Internet Networks, Rome, Italy, pp. 450- 457, 18-20 April 2005.

[14] A. Mahmoud, A. Alrefai, M. Abu-Amara, M. Sqalli, and F. Azzedin, "Qualitative analysis of methods for circumventing malicious ISP blocking," Arabian Journal for Science and Engineering, in press.

[15] C. Perkins, "IP encapsulation within IP," RFC 2003, Internet Engineering Task Force, Oct. 1996.

[16] R. Atkinson, "Security architecture for the internet protocol," RFC 1825, Internet Engineering Task Force", http://www.rfc-editor.org/rfc/rfc1825.txt, Aug. 1995.

[17] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic routing encapsulation (GRE)," RFC 2784, Internet Engineering Task Force, Mar. 2000.

[18] Shrinivasa Kini, Srinivasan Ramasubramanian, Amund Kvalbein, and Audun F. Hansen. "Fast recovery from dual link failures in IP networks," Proceedings of 2009 IEEE INFOCOM, Rio de Janeiro, pp.1368-1376, 19-25 April 2009.

[19] Jian Wu, Ying Zhang, Z. Morley Mao, and Kang G. Shin. "Internet routing resilience to failures: analysis and implications," Proceedings of the 2007 ACM CoNext, New York, US, 2007.

[20] "OPNET Modeler," http://www.opnet.com/.