IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 3 (Sep-Oct. 2012), PP 25-31
www.iosrjournals.org

Heuristic Route Discovery for Shared Firewall Network

Shahul Ali Khan P.Md[1], Shaik Jaffar[2], P.Saksha Ali[3], M.Narasimha Rao[4]
[1](Shahul Ali Khan P.Md currently doing his M.Tech in VLSI Technologies, Dept of ECE at MADINA Engineering College, Kadapa, A.P, India)
[2](Shaik Jaffar is currently working as Associate Professor and HOD, Dept of EIE, MADINA Engineering College, Kadapa, A.P, India)
[3](P .Saksha Ali, IT Manager, Andhra Pragathi Grameena Bank, Kadapa, A.P, India)
[4](M. Narasimha Rao is currently doing his M.Tech in VLSI Technologies, Dept of ECE at MADINA Engineering College, Kadapa, A.P, India)

Abstract: *In a connection-oriented system, shared firewall protection provides the identical stage of protection against on its own firewall failures as dedicated firewall protection, with potentially superior network exploitation. This document lists the Constraints of firewall placement strategies and proposes a heuristic routing algorithm for communal security provisioning. Simulations be conducted to confirm the algorithm and to balance network utilization of shared firewall protection to that of devoted fortification.*
Keywords: *Firewall, Network Security, intrusion, route discovery*

I. Introduction

A shared firewall protection system provides end-to-end firewalls in wired or wireless networks. To prevent from intruded traffic, a firewall may be shared by another firewall, or so-called relay firewall. The shared firewall has the same source and destination as the original or main firewall. When the firewall in sequence failed to detect intruded traffic, the shared firewall in the sequence is activated to go on with analyzing traffic. Statistically, the failure probabilities of the firewall should be rule specific; consequently if the failure probability of one firewall is p_f < 1, the probability of predicting intruded traffic failure is reduced to $p_f^2 < p_f$.

Based on whether the distribution of network resources is allowed, a guard scheme can be categorized as dedicated firewall protection or shared firewall protection. In dedicated safety, different protection firewalls do not share related set of rules, which may be dedicated to a substantial transmission line. Here we immediately refer to these different positions as "connections" between two nodes. The failure and activation of one firewall don't affect any other shared firewall. The provisioning of this type of defence is easy, and its performance is deterministic. On the other hand, in shared defence, multiple firewalls share the related set of rules. When a shared firewall is activated, other shared firewalls to facilitate sharing common connections with it will have to be rerouted. When a firewall in the sequence fails, all shared firewalls to facilitate to analyze the traffic, consequently shared firewall protection is more multifaceted to stipulate and maintain.

However shared firewall protection does propose one benefit over dedicated firewall protection, i.e., it may offer superior network utilization. Assume that each part of the traffic needs to verify, in the dedicated case, the most excellent network operation would be 50%. On the additional hand, for shared firewall protection, while multiple firewalls share common connections, the total number of connections required for all the protection firewalls can potentially be a lot lower. If the failure probabilities of main firewalls are statistically self-sufficient, we wouldn't wait for multiple firewalls to fail at the same time, in which case shared firewall protection provides the identical protection as dedicated firewall protection. This idea can be illustrated in the subsequent example.

Suppose that there are 10 most important sets of firewalls in a network, every with a failure probability of 0.01. At several moments the probability of firewall breakdown is $1- \ (1-0.01)^{10} = \ 0.9562$, of which on its own firewall failure probability is $10*0.01*(1-0.01)^9 = 0.9135$. Thus a single firewall failure counts for 95.534% of total firewall failures, for which shared firewall protection performs as well as dedicated firewall protection. For the remaining 4.466% failures, i.e., various firewall failures, the presentation of shared firewall protection would depend on how in effect the other protection firewalls are rerouted when a single protective firewall is activated.

The shared firewall protection provides decent protection with a lot lower system resources; thus, the system can realize higher consumption. Shared firewall protection and enthusiastic protection schemes

complement each other to offer more flexible solutions. Only the firewalls with the most severe protection obligation need to be protected in a dedicated manner. The left behind firewalls can be protected under the shared firewall protection and complimentary up network property, to moreover support more firewalls, or to guard firewalls that had no fortification before.

The profit of shared firewall protection is paying attention some research comfort, especially for the emerging all visual networks. ([1], [2]). This document tries to find a nonspecific framework for shared firewall based defence routing that is appropriate in connection inclination network together with the all optical densed distributed network systems. The next segment describes a heuristic routing algorithm for setting up shared firewalls that followed by Section III, which discusses other issues related to shared guard, and section 4 presents' reproduction results for the shared firewall proposal. Section 5 concludes the document.

I. Routing Algorithm

Initially, we catalog the Constraints for the shared firewall protection routing algorithm.

1.1 Constraint 1

A direction-finding algorithm to firewall defence is subjected to the danger disjointing constraint, i.e., a main firewall and its protection firewall must not undertake the same hard(s); otherwise the same failure may cause both firewalls to fail. This Constraint applies to both dedicated and shared firewall protections.

For each hard, we can assign a unique number, a Peril ID. If a connection in the network is subject to various complexities, the group of the peril IDs describes all the complexities for the connection, and the gathering of the peril IDs of all of a firewall's connections describes the firewall's entirety complexities. This collection of peril IDs is called the Hard Vector. For occurrence, in a dense distributed network system, a light firewall consists of two connections, l_1 and $_c$. l_1 runs crossways two bridges, A and B. l_2 crosses one bridge, C. The failure of some bridge is able to reason the light firewall to fail. If we allocate peril ID 2 to A, peril ID 5 to B and peril ID 3 to C, then {2, 5} is l_1 's hard vector and {3} is l_2's peril vector. The light firewall's peril vector is then {2, 5, 3}. With the notion of peril vector, the hard taking apart constraint requires that the present must not be any ordinary peril IDs in the peril vectors of a main firewall and its guard firewall.

Figure 1 gives another example. There are two visual light firewalls in a dense distributed network system, $l_1 =$ abc and $l_2 =$ abdc. Connection ab of l_1 and connection ab of l_2 are on top of the similar fiber between swelling a and b, so both connections are subjected to the same fiber failure. We are able to assign an ordinary peril ID 2 to the thread ab.

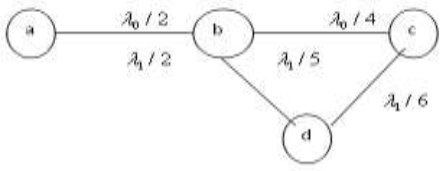

Fig. 1 Peril IDs in a dense distributed network

Connection bc of l_1 and connection bd, dc of l_2 are on dissimilar fibers in disjoint terrain. The complexities of fiber breakdown are different. We allocate a peril ID of 4 to bc, 5 to bd and 6 to dc. Combining all connections, we have l_1's peril vector {2, 4} and connection l_2's peril vector {2, 5, 6}. It is clear that l_1 and l_2 do not satisfy the hard disjointing restriction because of their common hard ID 2.

1.2 Constraint 2

Together the main firewall and guard firewall must be routed in order to claim success. This Constraint applies to dedicated firewall protection as well. It is up to the network operator to handle the failure. For instance, the operator may decide that the request is infertile, or might make the main firewall unprotected. Due to the hard disjointing constraint, if the peril IDs of the main firewall's peril vector appears on many unexploited connections, those connections will have to be disqualified from the guard firewall routing. Clearly, our routing algorithm should have a partiality for the connections with rare peril IDs. This applies to the routing of both the main firewall and protection firewall.

1.3 Constraint 3

If a connection is previously taken by a guard firewall, that connection should be communal as much as probable by subsequent protection firewalls, up to the highest number allowable on that connection. The point is to reduce the number of entirety connections taken by protection firewalls in the system. Therefore communal connections should be particular higher partiality for direction-finding the protection firewall.

1.4 Constraint 4

Stipulation various protection firewalls share common connections, those protection firewalls should not activate concurrently. In arranging to achieve this, the direction finding algorithm must prohibit protection firewalls from allocation common connections if their main firewalls include common elements in their peril vectors.

1.5 Constraint 5

For various routing Constraints, we can process the requests either one at a time, or all at once. The latter has a higher chance of obtaining additional optimal routes, but in a distributed system, routing Constraints often get there at different nodes of the system. It is more realistic and simpler to route Constraints one at a time. Once we enhance the algorithm for a single application, we can grip the multi-request case by organization iterations of the algorithm and choosing the majority optimal routes.

1.6 Constraint 6

Specific networks may inflict additional Constraints. For occurrence, a dense distributed network system has the continuity constraint, in which case we might need to run iterations of the algorithm, one for each type of traffic.

The heuristic direction-finding algorithm we are proposing is a customized OSPF routing algorithm. Each node has a global system topology and complete information about every connection in the network. In addition to OSPF common connection state information, all nodes have information about every connection regarding,

1. The connection's peril IDs.
2. Whether the connection is already taken by a main firewall.
3. Whether the connection is running a protection firewall. If so, the peril IDs of the main firewalls are also known. If a lot of protection firewalls share this connection, the total data on this item is potentially huge.
4. The highest number of shared firewall protection firewalls the connection supports. By lowering this number, we are able to decrease the quantity of data for item 3.

Based on the on top of information, we will adapt the connection costs such that the OSPF algorithm generates the routes that get together all of the Constraints. The algorithm is run at the source node and proceeds explicit routes.

For direction-finding a main firewall, we modify the cost of every connection as follows:

1. Set the cost to time without end if a connection is previously taken with a main firewall or defend firewall.
2. Enlarge the cost if the peril IDs of the connection have high occurrence in the system. For instance, for each occurrence of the peril ID, we increase the connection cost by a certain percentage, or by a fixed amount.

Commonly, the consequential cost, c/, of the i-th connection, is a function of the innovative cost c and the number of occurrences of its peril ID, n_{ri}, $i.e., c' = f(c_{;}, n_{ri})$, and $c' > c_{;}$. This purpose may be either linear or non-linear.

3. Enlarge the cost if the connection has the same peril IDs of existing main firewalls.

Both 2 and 3 potentially increase the extent of sharing when routing the fortification firewall, but item 3 requires the basis node to be conscious of the peril IDs of all existing main firewalls, which may be a hard task.

Subsequent to routing the main firewall, we modify the connection cost to route its fortification firewall:

1. Set the cost to infinity if a connection is already taken by a main firewall. Efficiency this connection is unconcerned.
2. Set the cost to infinity if a connection is running the maximum number of protection firewalls. Effectively this connection is removed.
3. Set the cost to infinity if a connection has a ordinary peril ID with the hard vector of the most important firewall. If *Constraint 1*, the hard disjointing constraint, is tolerated, then the connection cost might be set to a large positive numeral instead of time without end.
4. Set the cost to time without end if a connection is management a protective firewall whose main firewall has common peril IDs with the current main firewall. If *Constraint 4* is liberal, the connection cost might be set to a huge positive numeral instead of time without end.
5. Enlarge the cost if the peril IDs of the connection have a high amount in the system as in item 2 of higher than a main firewall routing algorithm.
6. For the outstanding connections, reduce the cost if a connection is running at smallest amount one but less than the utmost number of defence firewalls. The lower connection cost makes the connection more preferable and increases the extent of sharing. For case in point, for each shared firewall protection firewall immobile allowed on this connection, we reduce the connection cost by a certain percentage, or by an unchanging amount, pending it reaches least amount. The smallest amount cost should be a constructive number.

Generically, the consequential cost, c/, of the i^{th} connection, is a function of the innovative cost c and

the number of protection firewall unmoving allowed on this connection, n_{pi}, i.e., $c' = g(c, n_{pi})$, and $c > c' > 0$. This function may be either linear or non-linear.

Now we demonstrate the algorithm in an illustration with the system shown in Figure 2.

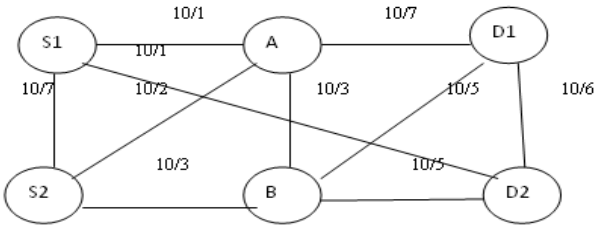

Figure 2: Example shared firewall network with connection costs and peril IDs

Here are two directions-finding requests. The first request asks for a firewall from swelling S1 to swelling D1. The second request asks for a firewall from S2 to D2. Each connection has a cost of 10 and peril IDs as marked in the figure. All connections and firewalls are bi-directional. With dedicated firewall protection, one of the firewalls would have to be unprotected.

With shared firewall protection, S1D1 is routed first. For the main firewall, since it is the initial firewall in the network, we only need to modify the connection costs based on the peril ID occurrence before running OSPF. For each extra occurrence of a peril ID, we increase the connection cost by 10%. The resulting network is shown in Figure 3. Running OSPF yields a main firewall S1-A- D1. Its hard vector is {1,7}. The total cost is 20.

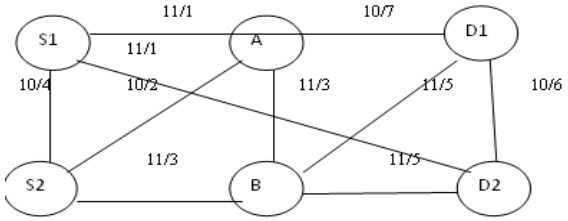

Figure 3: Firewalls with modified connection costs

Next we route S1D1's protection firewall. Since it is the first protection firewall in the network, we only need to remove the main firewall and the connections with common peril IDs with the main firewall from Figure 3. We then obtain the protection firewall S1-S2-B-D1, as shown in Figure 4. Its hard vector is {3, 4, 5}. This firewall satisfies the hard disjointing constraint. The total cost is 30.

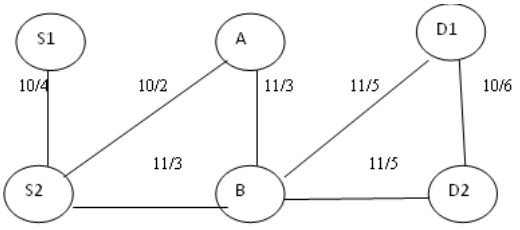

Figure 4: Network after routing the first request

We now process the second request. To route the main firewall from S2 to D2, we need to remove the connections on the main firewall S1-A-D1 and its protection firewall S1-S2-B-D1 from Figure 3. We also need to increase the cost of connections that have a common peril ID with the hard vector of main firewall S1-A-D1, {1,7}. The resulting network topology is shown in Figure 5. OSPF yield the second main firewall S2-A-B-D2. Its hard vector is {2, 3, 5}, and its total cost is 30.

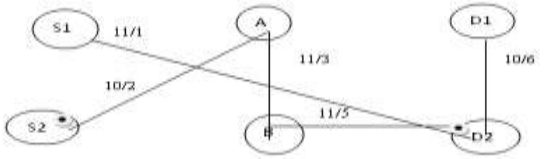

Figure 5: Network with modified connection costs

To route the protection firewall, we need to remove all connections of the two main firewalls from Figure 3 and connections with peril ID 2 or 3 or 5. Then we decrease by 10% the cost on connections that are

running the first protection firewall. The resulting topology is shown in Figure 6. The protection firewall becomes S2-S1-D2. Its hard vector is {1, 4}, and its total cost is 20.

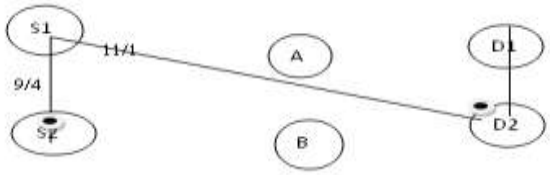

Figure 6: Network after routing the second request

Finally we obtain the network with all firewalls routed as shown in Figure 7. Solid lines indicate the main firewall; dash lines indicate the protection firewall. Connection S1S2 is shared by two protection firewalls. With shared firewall protection, both main firewalls are protected, which is infeasible through dedicated defense.

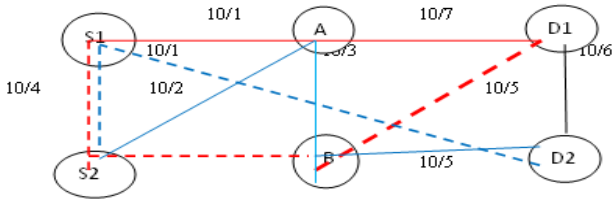

Figure 7: Network with both requests routed

II. Issues And Improvements

2.1 Additional Data and Computation Constraints

Compared with the algorithm for dedicated firewall protection, the algorithm for the shared firewall protection requires connections by protection firewalls to have the extra knowledge of the peril IDs of the related main firewalls. If the maximum number of shared firewall protection firewall allowed is M, and the number of nodes in the network is N, then on each connection, the number of peril IDs is on the order of Oc(M*N). To arrange to find out whether the firewalls have any common peril IDs with the target main firewall, the algorithm needs a spare $O\left(NlogN + \left(M*N\right) \log \left(M*N\right)\right)$computation on every connection.

2.2 Firewall Removal

When a main firewall and its guard firewall are torn down, the resources that were once occupied are freed up. If we redirect the remaining firewalls, we may get most favourable routes ([3]). This applies to both enthusiastic protection and shared firewall protection.

With either type of protection, rerouting most important routes may cause traffic hits. It may be more practical to reroute only the protection firewalls from time to time.

2.3 Protection Activation

When various protection firewalls share common connection(s), only one can be activated at a time. In order to allow various beginning, we can do one of the following after a protective firewall is activated:

- Reroute the failed main firewall. Once the fresh main firewall is recognized, move traffic into it from the protection firewall, then deactivate the protection firewall. This move toward requiring one reroute, plus signalling for firewall deactivation. The prospect of traffic hit is high when traffic is motivated to the new main firewall.
- Disappear the traffic on the activated protection firewall and create it the new main firewall. Establish a new protective firewall for it, as well as reroute all other protection firewalls that shared a common connection(s) with it. This approach doesn't introduce a traffic hit, but it requires rerouting multiple protection firewalls as well as signalling connected with the rerouting.

The network operator should decide which option to take. If the end user has a high tolerance for traffic hits, the first approach is clearly more suitable, since, when the system utilization is relatively high; rerouting multiple protection firewalls has higher failure probability than rerouting only the main firewall. On the other hand, if rerouting various protection firewalls is not an issue, then the subsequent approach may be considered.

2.4 Signalling

Shared firewall protection requires additional signalling than dedicated firewall protection. In addition to firewall establishment and removal, protection activation needs further signalling as described above. Signalling can be completed either in-band or out-of band, depending on the system type.

2.5 Maximizing Connection Sharing

Constraint 4 prohibits connection sharing amongst protection firewalls whose main firewalls have common peril IDs. Nevertheless, if we break each main firewall into various segments, we may find hard disjointing segments of the main firewalls. We can then establish segment-based protection instead of firewall based, and achieve higher sharing this way ([4]).

III. Simulations

We ran our simulations using the 16-swelling, 25-connection spinal column topology as exposed in Figure 8. The cost of every connection is assumed to be 100, and the peril IDs are as noticeable in Figure 8.

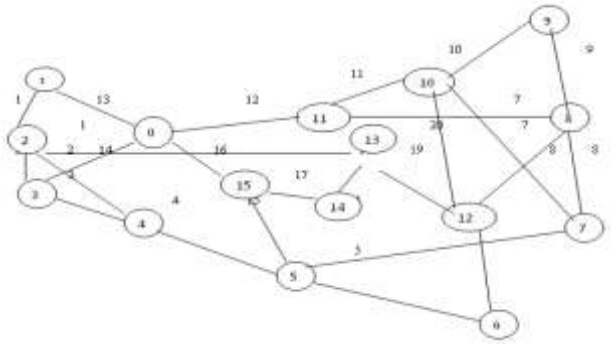

Figure8: The spinal column topology

Let the bandwidth of every one connection be BW. A main firewall takes one component of bandwidth, as does a dedicated firewall protection firewall. When various protection firewalls share a ordinary connection, they take one component of bandwidth. We ran simulations with various principles of BW.

We used typical Dijkstra's Shortest Firewall algorithm for the dedicated firewall protection and the heuristic algorithm described earlier for shared firewall protection.

For shared firewall protection, we enlarged a connection's cost by 100% for each occurrence of its peril ID when routed main and protection firewall. We also decrease a connection's cost by 50% if it ran a protection firewall when the routed protection firewalls.

We randomly generated 500 source-destination pairs as the routing requests. Then we compared the number of successes for dedicated firewall protection and shared firewall protection. For shared firewall protection, we also changed the maximum number of shared firewall protection firewalls, M, allowed on each connection.

We list the simulation results in Table 1. The first row contains the numbers of firewall pairs being successfully routed with dedicated firewall protection, with various connection bandwidths, BW. The remaining rows contain the results for shared firewall protection, with different connection bandwidths, and sharing degrees, M.

Two observations can be made from the results. First, shared firewall protection routes additional requests than dedicated defence. It confirms our earlier psychoanalysis that shared firewall protection offers higher network utilization. It is also significance noting that network utilization can increase fairly considerably even with the smallest amount of sharing. For instance, when the connection bandwidth is 10, 36 main-protection firewall pairs are routed successfully beneath dedicated firewall protection. But with only a sharing of two, 53 pairs are routed beneath shared firewall protection, an increase of nearly 50% in network utilization. Secondly, in shared firewall protection, higher degree of sharing further than 8 do not provide considerable additional gains of system utilization. This is a region deserves further study.

Protection Type		Number of Firewall Pairs Routed			
		BW=2	BW=5	*BW* = 10	BW = 20
Dedicated		8	18	36	67
Shared	M=2	10	26	53	98
	M=4	12	34	69	128
	M=8	13	37	74	134
	M=16	13	37	74	134
	M=32	13	37	74	134

Table 1: simulation results for path discovery for firewall placement

IV. Conclusion

Shared firewall defence provides a decent stage of protection with less system resource than dedicated firewall protection. It complements the in progress dual level protections of no protection or dedicated firewall protection, and tender three-level defence. It does so at the expense of additional signalling, data, computation, and firewall rerouting. Practical implementations can use a little degree of sharing to decrease the extra disbursement while still achieving higher system utilization.

References

[1] A. Rubin, D. Geer, and M. Ranum, Web Security Sourcebook. Wiley Computer Publishing, 1997.
[2] S. Hinrichs and S. Chen, "Network Management Based on Policies," Proc. SPIE Multimedia Computing and Networking Conf., Jan. 2000.
[3] J. Wack, K. Cutler, and J. Pole, Guidelines on Firewalls and Firewall Policy. Nat'l Inst. of Standards and Technology, Jan. 2002.
[4] Y. Bartal, A. Mayer, K. Nissim, and A. Wool, "Firmato: A Novel Firewall Management Toolkit," ACM Trans. Computer Systems, vol. 22, no. 4, pp. 381-420, Nov. 2004.
[5] A. Wool, "A Quantitative Study of Firewall Configuration Errors," Computer, vol. 37, no. 6, pp. 62-67, June 2004.
[6] H. Court, Knutsford, and Cheshire, "High-Availability: Technology Brief Firewall Load Balancing," High-Availability.Com, http://www.High-Availability.Com, 2003.
[7] "Firewall Load Balancing," Nortel Networks, www.nortel.com, 2009.
[8] "Check Point Firewall-1 Guide," Check Point, www.checkpoint. com, 2009.
[9] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms. MIT Press, 2003.
[10] M.G. Gouda and A.X. Liu, "Firewall Design: Consistency, Completeness and Compactness," Proc. Int'l Conf. Distributed Computing Systems (ICDCS '04), pp. 320-327, Mar. 2004.
[11] A.X. Liu and M.G. Gouda, "Diverse Firewall Design," Proc. IEEE Int'l Conf. Dependable Systems and Networks (DSN '04), pp. 595-604, June 2004.
[12] M.G. Gouda and A.X. Liu, "A Model of Stateful Firewalls and Its Properties," Proc. IEEE Int'l Conf. Dependable Systems and Networks (DSN), June 2005.
[13] A.X. Liu, M.G. Gouda, H.H. Ma, and A.H.H. Ngu, "Firewall Queries," Proc. Eighth Int'l Conf. Principles of Distributed Systems (OPODIS), Dec. 2004.
[14] A.X. Liu, "Change Impact Analysis of Firewall Policies," Proc. 12th European Symp. Research Computer Security (ESORICS), Sept. 2007.
[15] A.X. Liu, "Formal Verification of Firewall Policies," Proc. IEEE Int'l Conf. Comm. (ICC), May 2008.
[16] A.X. Liu, E. Torng, and C. Meiners, "Firewall Compressor: An Algorithm for Minimizing Firewall Policies," Proc. IEEE INFOCOM '08, Apr. 2008.
[17] A. Wool, "The Use and Usability of Direction-Based Filtering in Firewalls," Computers and Security, vol. 23, no. 6, pp. 459-468, 2004.
[18] E.W. Fulp, "Optimization of Network Firewall Policies Using Ordered Sets and Directed Acyclical Graphs," Proc. IEEE Internet Management Conf., 2005.
[19] E.S. Al-Shaer and H.H. Hamed, "Discovery of Policy Anomalies in Distributed Firewalls," Proc. IEEE INFOCOM '04, Mar. 2004.
[20] R.N. Smith, Y. Chen, and S. Bhattacharya, "Cascade of Distributed and Cooperating Firewalls in a Secure Data Network," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 5, pp. 1307-1315, Sept./Oct. 2003.
[21] R.N. Smith and S. Bhattacharya, "Firewall Placement in a Large Network Topology," Proc. IEEE CS Workshop Future Trends Distributed Computing Systems (FTDCS '97), 1997.
[22] H. Hamed, A. El-Atawy, and E. Al-Shaer, "On Dynamic Optimization of Packet Matching in High Speed Firewalls," IEEE J. Selected Areas in Comm., vol. 24, no. 10, pp. 1817-1830, Oct. 2006.
[23] A. El-Atawy, T. Samak, E. Al-Shaer, and H. Li, "On Using Online Traffic Statistical Matching for Optimizing Packet Filtering Performance," Proc. IEEE INFOCOM '07, May 2007.
[24] P. Gupta and N. McKeown, "Algorithms for Packet Classification," IEEE Network, vol. 15, no. 2, pp. 24-32, Mar. 2001.
[25] P. Gupta and N. McKeown, "Packet Classification on Multiple Fields," Proc. ACM SIGCOMM '99, 1999.
[26] T. Lakshman and D. Stiliadis, "High-Speed Policy-Based Packet Forwarding Using Efficient Multi-Dimensional Range Matching," Proc. ACM SIGCOMM '98, 1998.
[27] A. Hari, S. Suri, and G. Parulkar, "Detecting and Resolving Packet Filter Conflicts," Proc. IEEE INFOCOM '00, Mar. 2000.
[28] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, "Fast and Scalable Layer Four Switching," Proc. ACM SIGCOMM '98, 1998.
[29] P. Gupta, "Algorithms for Routing Lookups and Packet Classification," PhD thesis, Stanford Univ., 2000.
[30] A.X. Liu and M.G. Gouda, "Removing Redundancy from Packet Classifiers," Proc. Ann. IFIP Conf. Data and Applications Security, Aug. 2005.
[31] C.R. Meiners, A.X. Liu, and E. Torng, "TCAM Razor: A Systematic Approach towards Minimizing Packet Classifiers in TCAMs," Proc. IEEE Int'l Conf. Network Protocols (ICNP), Oct. 2007.
[32] A.X. Liu, C.R. Meiners, and Y. Zhou, "All-Match Based Complete Redundancy Removal for Packet Classifiers in TCAMs," Proc. IEEE INFOCOM, Apr. 2008.
[33] MyungKeun Yoon, Shigang Chen, and Zhan Zhang; Minimizing the Maximum Firewall Rule Set in a Network; IEEE TRANSACTIONS ON COMPUTERS, VOL. 59, NO. 2, FEBRUARY 2010.