A Methodology for OSPF Routing Protocol Verification

Saif Ur Rehman Malik, Sudarshan K. Srinivasan, Samee U. Khan
Department of Electrical and Computer Engineering
North Dakota State University
Fargo, USA, 58108-6050
{saif.rehmanmalik, sudarshan.srinivasan}@ndsu.edu,
samee.khan@ndsu.edu

Lizhe Wang
Center for Earth Observation and Digital Earth,
Chinese Academy of Sciences,
P. R. China
lizhe.wang@gmail.com

***Abstract* - The emerging cloud computing paradigm has driven the creation of data centers that consist of hundreds of thousands of servers and that are capable of supporting a large number of distinct services. Data intensive systems have a real need for tens to hundreds of Gbps of bandwidth and deterministic Quality of Service (QoS), which is satisfied by thousands of servers interconnected together. Security of the data flowing in and out of the data center is of high concern. In this regard, verification of the protocols that are running into the data center is compulsory. Simulations and testing are the techniques, widely used for the verification purposes. But in case of data intensive systems the aforementioned techniques become infeasible because of large-scale datasets, flowing among physical storage devices and computational clusters. In this paper, we use verification tools and methods to verify the protocols running in the data center. As an example, we took an Open Shortest Path First (OSPF) routing protocol that is used for routing within the data center. We used (a) content verification and (b) route verification, to verify the correctness of OSPF on multi-access segments. We propose a novel method for route verification that is based on delay information, which can be further exploited to verify protocols in a scalable manner. Moreover, we use the Z3 solver as a tool for verification.**

Keywords: OSPF, Verification, Data Center

I. INTRODUCTION

Data intensive systems, such as data centers, have a real need for tens to hundreds of Gbps of bandwidth and deterministic Quality of Service (QoS), which is satisfied by thousands of servers interconnected together. Data Centers (DC) gained a great popularity for the provision of computing resources [3]. Amazon, Google, IBM, Facebook, and Microsoft have started to establish data centers that host cloud computing applications in geographically distributed locations [2]. DCs contains a pool of computing resources to host applications and store data, connected together using communication medium, such as fiber optics. The performance and stability of the network depends on the performance of the routing mechanisms implemented within the architecture [4]. Routing protocol plays an important role towards the performance realization of large scale networks. Therefore, it is compulsory to verify the working of the routing protocol to ensure reliable communication amongst the systems in the network.

Open Shortest Path First (OSPF) is an adaptive routing protocol that is used for Internet Protocol (IP) networks to distribute routing information within a single Autonomous System (AS) [5]. OSPF divides the network into areas, as shown in fig. 1 [1]. Each area consists of one or more segments. A segment constitutes the set of routers connected via a common communication channel, such as Ethernet. With the rapidly increasing and changing demands of QoS, modern routing domain, such as DCs need to maintain a very high level of service availability. Therefore, OSPF should attain fast convergence in response of topology changes, to meet the demands of modern systems. Moreover, to avoid loss of messages, the information flowing within the data center must be routed correctly by the OSPF. A slight misinformation can cause huge packets loss depending on the size of the network. In the aforementioned aspect, we have verified OSPF protocol with the help of the following:

A. SMT-LIB and Z3 Solver

Satisfiability Modulo Theories (SMT) is an area of automated deduction for checking the satisfiability of formulas with respect to some logical theory of interest [18]. SMT has been used in many fields including deductive software verification. Moreover, recent applications of computer science including planning, model checking, and automated test generation finding, also considers SMT as an important verification tool [19]. The solver can be distinguished amongst the features they provide, such as, underlying logic (example first order or temporal), background theories, input formulas, and interface. The details about the features can be found in [30]. Multiple solvers are available that supports SMT-LIB, such as Beaver, Boolector, CVC4, MathSAT5, Z3, and OpenSMT [19].

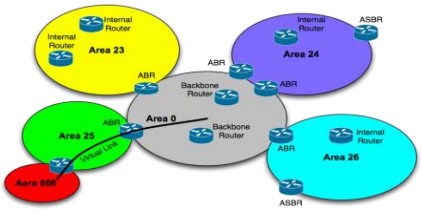

Figure 1. OSPF Areas and Routers.

We used Z3 solver in our study, which is a high performance theorem prover developed at Microsoft Research. Z3 is an automated satisfiability checker. Moreover, Z3 also checks whether the set of formulas are satisfiable in the built-in theories of SMT-LIB. Readers are encouraged to see [20], for the detailed information about the working and commands of Z3 solver.

In this paper, we propose a novel method to verify the OSPF routing protocol that incorporate Designated Routers (DRs). The proposed method uses delay information of the router as a property to verify the protocol. We used the delay information to identify the occurrence of events as an update is received by the corresponding DRs. The proposed method can scale up the verification process by reducing the size of state space and limiting it to a single parameter. We used Satisfiability Modulo Theory (SMT-LIB) library and Z3 solver as a tool for the verification purpose. Moreover, BRITE [9] topology generator is used to generate the topologies that represents characteristics similar to those of Internet. There are four steps involved in our verification process: **(a)** we have simulated the detailed implementation of OSPF protocol based on the specifications available in [12] on a small scale network, **(b)** we modeled the system and specified the properties, **(c)** the model and properties in SMT-LIB are given to Z3-Solver for model checking, and **(d)** execution and generation of results.

II. OSPF Routing Protocol

OSPF is a link-state routing protocol [6]. The link state is the description of the interface of the router (IP address of the interface, mask, type of network, routers connected to) and the relationship to other routers. OSPF constructs a topological map of the entire network by gathering the link state information from available routers [1]. Unlike other routing protocols, such as Routing Information Protocol (RIP) that uses Bellman-ford vector based algorithms, OSPF introduces new concepts, such as areas, Variable Length Subnet Mask (VLSM), and route summarization [14].

To decrease the intra-area convergence time, a router amongst the routers is selected as a DR in OSPF. All other routers on a segment communicate only to the DR, which cuts the information flow cost from $O(n^2)$ to $O(n)$ (instead of sending update to every router on a segment the update is sent to a DR and then that DR will flood the update to all of the other routers) [6]. Table 1 illustrates the types of routers used in OSPF. The type of router is identified based on the router interface and link states. Do not confuse DR with OSPF router types. A router can have some interfaces that are designated, which makes a router DR. Moreover, different types of routers generate different Link State Advertisements (LSA), which is a way to communicate the routing topology to other router in and outside an area. Table 2 depicts some of the basic LSAs supported by the routers. Note, that there are other LSA types 6-11, whose information can be found in [15].

TABLE I OSPF ROUTERS

Router Type	Description
Internal	Router that has all the interfaces in single area
Backbone	Router that has at least one interface in backbone area
Area Border	Router having at least one interface in backbone area and another in non-backbone area
Autonomous System Boundary	Router performing route injection from other source (RIP, EIGRP) into OSPF domain.

TABLE II OSPF LINK STATES AND ASSOCIATED ROUTERS

LSA Type	Description	Associated Router	Scope
1	Describes directly attached link to a router within an area.	All routers	Intra area
2	Describes the number of routers attached in a segment. Gives information about the subnet mask of a segment	DR	Intra area
3	Describes destinations outside an area to flood information from one area to another.	ABR	Inter area
4	Describes a route and information to an ASBR outside the area.	ABR	Inter area
5	Defines routes to destinations external to OSPF domain.	ASBR	Inter area

III. Problem Formulation

The problem formulation is taken from our previous work in [31]. However, the formulation is modeled accordingly to accommodate the verification aspect of the OSPF protocol. Consider a network composed of N routers. Let R_i be the i^{th} router, where $1 \leq i \leq N$. A link between two routers R_i and R_j (if it exists) has a communication cost *(del)* that represents the minimum time for transferring message from R_i to R_j, which can be represented by the following expression [29, 31]:

$$del(R_i, R_j) = \frac{D(R_i, R_j)}{v} + \frac{s}{\beta_{ij}}, \qquad (1)$$

where, $D(R_i, R_j)$ is the physical distance between R_i and R_j, v is the propagation delay of the medium (optical fiber in our case), s is the size of the message in kilobytes, and β_{ij} is the available bandwidth between R_i and R_j. If the routers are not directly connected, then the communication cost is the sum of the cost of all links in the shortest path from R_i to R_j. Without the loss of generality, we assume that $del(R_i, R_j) = del(R_j, R_i)$, which is a common assumption in literature [29]. Let M be the number of segments within an area and S^k be the k^{th} segment in that area, where $1 \leq k \leq M$. Let DS be the set of DRs within an area and ς^k is the convergence time (time a router takes to discover the area topology) of S^k. If a failure occurs (could

be a link or a router), the routers connected to the failed link or failed router will initiate the updates. Let R_o^k be a router that initiates an update in response to a failure. Let Rr be the set of all other routers in the area defined as $Rr = \left(\cup_{i=1}^{N}\{R_i\}\right)\backslash\{\{R_o^k\} \cup DS\}$.

Suppose R_o^k gets an update, such as node failure. R_o^k will update its link state and forward the updated link state to the DR of segment *k* (represented as d^k). The link state is the description of the interface of the router (IP address of the interface, mask, type of network, routers connected to) and the relationship to other routers. The DR will then flood the information to every other router in the segment after receiving the update. The verification of the routing protocol can be done in two aspects: **(a)** content verification (if the link state is being calculated correctly) and **(b)** routing verification (if the information is propagated correctly in a same order). For **(a)**, the Link State Database (LSDB) should be same for all the routers after convergence is achieved, such that $LSDB(R_i^k) = LSDB(R_{i+1}^k) \ldots = LSDB(R_n^k) \forall k \in S$. For **(b)**, let $\mathcal{K}^k$contains the list of router that belongs to segment *k* in ascending order of $del(d^k, R_i^k) \forall k \in S: R_i \in k$. All the routers in a segment must receive the updates in a same order as listed in $\mathcal{K}^k$.

Let R_i^k represent a router R_i that belongs to S^k. The time for R_i^k to receive the update $\left(\psi(R_i^k)\right)$ can be calculated as follows [31]:

$$\psi(R_i^k) = \begin{cases} DI, & if\ R_i^k = R_o^k \\ \psi_{\forall k \in S: R_i \in k}(d^k) + del(d^k, R_i^k), & if\ R_i^k \in Rr \end{cases} \quad (2)$$

where,

$$\psi(d^k) = \psi_{\forall j \in k: k \in S}(R_j^k) + del(R_j^k, d^k). \quad (3)$$

Other updates, such as change in bandwidth ($\Delta\beta_{ij}$) are assumed to be local and incur zero update time. Therefore in (2), the value of $\psi(R_i^k)$ for $R_i^k = R_o^k$, is DI. The DI of routers is usually four times the "Hello" interval, which is the time between consecutive transmissions of "Hello" packets that are used to indicate the liveliness of nodes. The "Hello" interval is 10 seconds for broadcast and P2P networks, and 30 seconds for all other media [2]. The value of $\psi(R_i^k)$ for $R_i^k \in Rr$ is the sum of the time required for d^k to receive updates and the time d^k takes to deliver updates to R_i^k. The value of $\psi(d^k)$ is calculated in (3), which is the sum of $\psi(R_i^k)$ (the node sending the update to d^k) and the communication cost between them, which is given as $del(R_j^k, d^k)$. Moreover, (2) and (3) are used to calculate ς^k based on the following expression [31]:

$$\varsigma^k = \max\left(\psi(d^k) + del_{\forall j \in k}(d^k, R_j^k)\right). \quad (4)$$

The last router (maximum time taken to receive an update from the corresponding d^k) in S^k that receives the update, determines ς^k. Now, using (2), (3), and (4) the convergence time of an area τ can be calculated as follows:

$$\tau = \max_{\forall k \in S}\left(\left(\varsigma^k\right)\right) + \psi\left(\left(R_o^k\right) + del(R_o^k, d^k)\right). \quad (5)$$

The maximum ς^k amongst all of the segments plus the time when the update is initiated and reaches to the respective DR determines the value of τ.

IV. Verification of OSPF Using Proposed Method

Verification is the process to demonstrate the correctness of the underlying system [16]. We verify the correctness of OSPF through **(a)** content verification and **(b)** route verification as discussed in problem formulation. Note that our goal is to verify the correctness and not to measure the performance of the protocol. Two parameters are required for the verification of the system, namely specification and properties. We achieved verification through model checking [17], using SMT-Lib and Z3 Solver. The detailed description for the possible behavior of the protocol (specification) along with the desirable behavior (properties) of the protocol are modeled in SMT and provided to Z3. Given the aforementioned parameters Z3 can perform a verification of the model. Z3 generates a counter example in case of an error that represents the state or values for which the model is incorrect. If there are no errors, then the model specifications can be fine-tuned until converged to the real system. The proposed method can scale up the verification process by reducing the size of the state space and narrowing it down to a single parameter. In the following section we will discuss content verification and route verification in detail.

A. Content Verification

OSPF is a link state protocol and all routers in an area must have the same LSDB in order for the protocol to work correctly [1]. We assume that the DR is already being elected and initial LSDB synchronization is already being achieved. In content verification, we analyze the state of LSDB for all routers in an area as an update is generated and propagated by the corresponding DR. For content verification, we have simulated the detailed implementation of OSPF on multi-access segments having multiple DRs in one area. The system model and the property to verify are generated in SMT and are provided to Z3. The property to verify for content verification is that LSDB should be same for all the routers after convergence is achieved, such that $LSDB(R_i^k) = LSDB(R_{i+1}^k) = \cdots = LSDB(R_n^k) \forall k \in S$. Whenever an update occurs, the router initiating an update generates a LSA. The LSA must be propagated to all the routers in an area to have the same view of the topology and to reach the stable state. The aforementioned is necessary to avoid message loss and for the protocol to work properly.

B. Route Verification

The LSA generated by the routers in case of updates must be propagated to corresponding DR, and then from DR to all other routers in a segment. We propose the use of delay information of routers for route verification. The delay of routers is calculated using (1) as discussed in problem formulation. The delay information is further

utilized to order the events as an update occurs. Maintaining the order of the routers reduces the size of state space while verifying the protocol. If no such information is available, then all scenarios have to be considered during the verification process. Suppose we have a topology as shown in fig. 2 below, with arbitrary delays. The topology has two segments. Segment 1 (S^1) has six routers (R_0, *R1*, *R2*, *R4*, *R5*, *R7*) and *R2* is the DR (d^1) of S^1. Segment 2 (S^2) has three routers (R_3, R_4, R_6) and R_3 is the DR (d^2) of S^2. Suppose d^1 receives an update and $\psi(d^1) = 0$. Then using (2) we can calculate the $\psi(R_i^1))\forall i \in S^1$. R_4 is the connecting router between the two segments. Therefore, the update will be propagated from S^1 to S^2 through R_4. The $\psi(d^2) = 0.28$ (using (3)).If we want to verify the routing, then we can compare the difference of update time of routers and DR $\left((\psi(R_i^k) - \psi(d^k))\forall k \in S: R_i \in k\right)$ with $Ж^k$ to verify the routing.

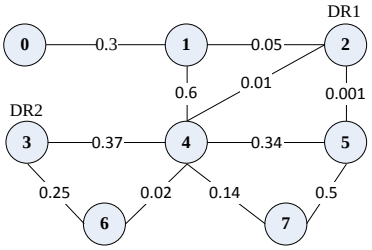

Figure 2. Example Topology and Associated Delays.

TABLE III COMPARISON OF UPDATE TIME AND ORDERED LIST OF ROUTER FOR EXAMPLE TOPOLOGY

$Ж^1$	$(\psi(R_i^1) - \psi(d^1))$	$Ж^2$	$(\psi(R_i^2) - \psi(d^2))$
0.001 (R5)	0.001-0=0.001	0.25 (R6)	0.53-0.28=0.25
0.01 (R4)	0.01-0=0.01	0.27 (R4)	0.55-0.28=0.27
0.05 (R1)	0.05-0=0.5		
0.15 (R7)	0.15-0=0.15		
0.35 (R0)	0.35-0=0.35		

We can analyze from Table that the values of $\left((\psi(R_i^k) - \psi(d^k))\forall k \in S: R_i \in k\right)$ and $Ж^k$ are identical, which indicate that the routing is done correctly and all the routers are receiving the updates in a correct order and time. If the values are not identical, then the protocol may have a problem.

V. RESULT AND DISCUSSION

Performance realization of large scale networks depends highly on the routing protocols. Therefore, it is compulsory to verify the working of the routing protocol to ensure reliable communication amongst the systems in the network. To this end, we have simulated the detailed implementation of OSPF, based on the specifications reported in [12] for **(a)** content verification and **(b)** route verification. In OSPF, the routers are usually the Level3 (L3) routers. Therefore, we used optical fiber as a communication medium having propagation delay $v = 300 \times 10^6 \text{miles/sec}$. Ethernet channels have the Maximum Transmit Unit (MTU) of 1500 bytes and in OSPF the fragmentation is usually avoided [1]. Therefore, the message size *s* is kept as 1KB, which is neither low nor high and which is typically used in the literature for experimentation (modeling, simulation, and testing). (Readers are encouraged to see the work reported in [7] and [8] to get and insight into the typical modeling and simulation parameters pertaining to the OSPF modules). Moreover, the bandwidth value of β_{ij} is kept at 100Mbps, as advocated in [9, 10, 11] for the evaluation purposes. The values of $D(R_i, R_j)$ are assigned from within the range of [1-100] km.

Fig. 3 depicts the execution time for the content and route verification. For content verification the link state for all the routers in an area must be same. To verify the aforementioned property, we have modeled the simulated system in SMT and generated link states for all the routers. When the convergence is achieved, then the link states of all the routers are compared with each other to verify the similarities. For route verification, as discussed in above section, the values of $\left((\psi(R_i^k) - \psi(d^k))\forall k \in S: R_i \in k\right)$ and $Ж^k$ must be identical in order for the protocol to work properly. The system model is verified to check if the aforementioned property is satisfied using SMT and Z3 solver. For our implementation using SMT-LIB, we used QF_AUFLIA logic [19], which is used for closed quantifier-free linear formulas over the theory of integer arrays extended with free sort and function symbols.

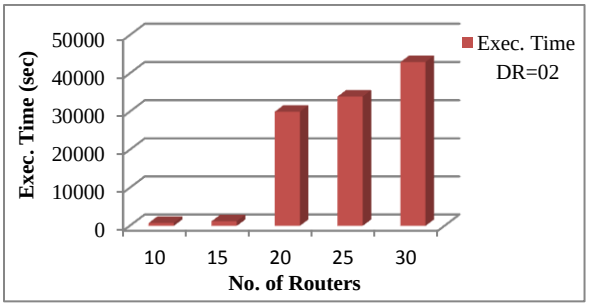

Figure 3. Execution Time for verification.

VI. RELATED WORK

A formal verification of ad-hoc routing protocols using SPIN model checker is performed in [21]. The authors of [21] used Wireless Adaptive Routing Protocol (WARP) to formally verify the real time aspects of the protocol. In [22], the authors studied different implementations of Ad-hoc On-demand Distance Vector (AODV) routing protocol. Moreover, to checks C and C++ implementations directly, the authors used their own model checker. A topology approximation algorithm is proposed in [23], to tackle the problem of mobility by modeling AODV using colored petri nets. In the paper [24], the authors performed specification and verification of LambdaRAM, which is a wide area

distributed cache for high performance computing. The authors in [24] used TLV for model checking, which uses SMV as an input language. Xiong *et* al. [23] have modeled AODV using colored Petri nets (CPN). Some other work towards the verification of routing protocols can be found in [27], [26], and [25].

VII. CONCLUSIONS

Formal analysis of routing protocols is compulsory for a secure and efficient performance of modern large scale networks. In this aspect, we have proposed a novel method to verify the properties of OSPF protocol that uses delay information of the routers. We have verified the protocol in two parts: **(a)** content verification and **(b)** route verification. For **(a)**, we verify the property that the LSDB for all the routers in an area must be identical. For **(b)**, we uses delay information to order the events and then verify if the events are occurring in the same order. The aforementioned properties are verified using SMT-LIB and Z3 solver. We have simulated the detailed implementation of OSPF and BRITE topology generator is used for the generation of realistic topological interconnections. The proposed method can scale up the verification by reducing the state space and narrowing it down to a single parameter.

REFERENCES

[1] J. T. Moy, 'OSPF; Anatomy of an Internet Routing Protocol', Addison-Wesley, 1998.

[2] D. Abadi, "Data management in the cloud: Limitations and opportunities", *IEEE Data Engineering, Bulletin*, Vol. 32, No. 1, 2009, pp.3-12.

[3] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: Data Center Energy-Efficient Network-Aware Scheduling," *ACM/IEEE International Conference on Green Computing and Communications (GreenCom)*, Dec. 2010, pp. 69-75.

[4] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb, "A case study of OSPF behavior in a large enterprise network", *ACM SIGCOMM Internet Measurement Workshop*, France, 2002.

[5] A. Caslow, Cisco Certification: Bridges, Routers & Switches for CCIEs. Upper Saddle River, NJ: Prentice Hall PTR, 1998, pp. 373-410.

[6] Cisco, 'OSPF Design Guide', http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094e9e.shtml, accessed 04 May, 2012.

[7] I. Krinpayorm, and S. Pattaramalai, "Link Recovery Comparison Between OSPF & EIGRP", ICICN, 2012, pp. 192-197.

[8] B. Wang, J. Zhang, Y. Guo, and W. Chen, "Fast-Converging Distance Vector Routing Mechanism for IP Networks", *Journal of Networks*, 2010, Vol. 05, No. 9, pp. 1069-1075.

[9] HP, "Building Virtualization-Optimized Data Center Networks", Technical Report. 4AA3-3346ENW, HP, 2011.

[10] R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools", *IEEE Network*, 2003, Vol. 17, No. 06, pp. 27-35.

[11] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. S. E. Ng, M. Kozuch , and M. Ryan, "c-Through: Part-time Optics in Data Centers", *SIGCOMM*, 2010.

[12] Moy, J. (April 1998). RFC 2328 "OSPF Version 2," The Internect Society OSPFv2.

[13] G. Retvari, F. Nemeth, R. Chaparadza, and R. Szabo, "OSPF for Implementing Self-adaptive Routing in Autonomic Networks: A Case Study", *Mid-American Association for Computers in Education (MACE)*, 2009, pp. 72-85.

[14] D. Katz, K. Kompella, D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", http://tools.ietf.org/rfc/rfc3630.txt, accessed on 02 May, 2012.

[15] RFC 2370, the OSPF Opaque LSA Option, IETF Network Working Group - Coltun - 1998

[16] Verification, http://en.wikipedia.org/wiki/Verification

[17] P. Wolper. An Introduction to Model Checking, 1995. http://www.montefiore.ulg.ac.be/~pw/papers/papers.html,

[18] C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. Satisfiability Modulo Theories. Handbook of Satisfiability, Vol. 185, chapter 26, pp. 825-885. IOS Press, February 2009.

[19] SMT-LIB, http://www.smtlib.org/

[20] Z3, http://research.microsoft.com/en-us/um/redmond/projects/z3/

[21] R. de Renesse and A. Aghvami, "Formal verification of ad-hoc routing protocols using SPIN model checker", *12th IEEE Mediterranean Electro technical Conference*, 2004, pp. 1177–1182.

[22] D. Engler and M. Musuvathi, "Static analysis versus software model checking for bug finding", *Verification, Model Checking, and Abstract Interpretation, 5th International Conference*, Lecture Notes in Computer Science, 2004, pp. 191–210.

[23] C. Xiong, T. Murata, and J. Tsai, "Modelling and simulation of routing protocol for mobile ad hoc networks using coloured Petri nets", *Workshop on Formal Methods Applied to Defence Systems in Formal Methods in Software Engineering and Defence Systems*, 2002.

[24] V. Vishwanath, L. Zuck, J. Leigh, "Specification and verification of LambdaRAM – a wide-area distributed cache for high performance computing" *6th IEEE/ACM Conference on Formal Methods and Models for Codesign (MEMOCODE) 2008*, USA, June 2008.

[25]. S. Chiyangwa, M. Kwiatkowska, "A timing analysis of AODV", *Formal Methods for Open Object-Based Distributed Systems: 7th IFIP WG 6.1 International Conference (FMOODS)*, (2005).

[26] D. Obradovic, Formal Analysis of Routing Protocols. PhD Thesis, University of Pennsylvania (2002).

[27] S. Das, D. L. Dill, "Counter-example based predicate discovery in predicate abstraction", *Formal Methods in Computer-Aided Design, Springer-Verlag*, (2002).

[28] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation", *MASCOTS*, Cincinnati, Ohio, 2001.

[29] Khan, S. U., and Ahmad, I., "A Pure Nash Equilibrium based Game Theoretical Method for Data Replication across Multiple Servers", *IEEE TKDE*, 2009, Vol. 21, No. 4, pp. 537-553.

[30] C. Barrett, A. Stump, and C. Tinelli, "The SMT-LIB Standard: Version 2.0", *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories*, 2010.

[31] S. U. R. Malik, S. K. Srinivasan, and S. U. Khan, "Convergence Time Analysis of Open Shortest Path First Routing Protocol in Internet Scale Networks," *IET Electronics Letters*, vol. 48, no. 19, pp. 1188-1190, 2012.