International Journal of Research
ISSN: 2348-6848 Vol-3, Special Issue-1
National Conference on Advanced Computing Technologies
Held on 21st July 2015 at Hyderabad city organized by **Global Research Academy,** Hyderabad, Telangana, India.

Development of a Nontrivial Approximation Algorithm to resolve Optimization Problem of Overlay Routing Resource Allocation

G.Vinod Kumar[1]; A.Shoba Rani[2]& Prof.Dr.G.Manoj Someswar[3]

[1]M.Tech.(CSE) from Narasimha Reddy Engineering College, Affiliated to JNTUH, Hyderabad, Telangana, India

[2]M.Tech. (CSE), Associate Professor, Department of CSE, Narasimha Reddy Engineering College, Affiliated to JNTUH, Hyderabad, Telangana, India

[3]B.Tech., M.S.(USA), M.C.A., Ph.D., Principal & Professor, Department Of CSE, Anwar-ul-uloom College of Engineering & Technology, Affiliated to JNTUH, Vikarabad, Telangana, India

Abstract:

In this research paper, we concentrate on this point and study the minimum number of infrastructure nodes that need to be added in order to maintain a specific property in the overlay routing. In the shortest-path routing over the Internet BGP-based routing example, this question is mapped to: What is the minimum number of relay nodes that are needed in order to make the routing between a group of autonomous systems (ASs) use the underlying shortest path between them? In the TCP performance example, this may translate to: What is the minimal number of relay nodes needed in order to make sure that for each TCP connection, there is a path between the connection endpoints for which every predefined round-trip time (RTT), there is an overlay node capable of TCP Piping. Regardless of the specific implication in mind, we define a general optimization problem called the Overlay Routing Resource Allocation (ORRA) problem and study its complexity. It turns out that the problem is NP-hard, and we present a nontrivial approximation algorithm for it.

Keywords: Overlay Routing Resource Allocation; Round trip time; Autonomous Systems; TCP Piping; Voice over IP; Resilient Overlay Network

INTRODUCTION

OVERLAY routing has been proposed in recent years as an effective way to achieve certain routing properties, without going into the long and tedious process of standardization and global deployment of a new routing protocol. For example, in [1], overlay routing was used to improve TCP performance over the Internet, where the main idea is to break the end-to-end feedback loop into smaller loops. This requires that nodes capable of performing TCP Piping would be present along the route at relatively small distances. Other examples for the use of overlay routing are projects like RON [2] and Detour [3], where overlay routing is used to improve reliability. Yet another example is the concept of the "Global-ISP" paradigm introduced in [4], where an overlay node is used to reduce latency in BGP routing.

In order to deploy overlay routing over the actual physical infrastructure, one needs to deploy and manage overlay nodes that will have the new extra functionality. This comes with a non negligible cost both in terms of capital and operating costs. Thus, it is important to study the benefit one gets from improving the routing metric against this cost.

Note that if we are only interested in improving routing properties between a single source node and a single destination, then the problem is not complicated, and finding the optimal number of nodes becomes trivial since the potential candidate for overlay placement is small, and in general any assignment would be good.

However, when we consider one-to-many or many-to-many scenarios, then a single overlay node may affect the path property of many paths, and thus choosing the best locations becomes much less trivial.

We test our general algorithm in three specific such cases, where we have a large set of source–destination pairs, and the goal is to find a minimal set of locations, such that using overlay nodes in these locations allows to create routes (routes are either underlay routes or routes that use these new relay nodes) such that a certain routing property is satisfied.

The first scenario we consider is AS-level BGP routing, where the goal is to find a minimal number of relay node locations that can allow shortest-path routing between the source–destination pairs. Recall that routing in BGP is policy-based and depends on the business relationship between peering ASs, and as a result, a considerable fraction of the paths in the Internet do not go along a shortest path (see [5]). This phenomenon, called path inflation, is the motivation for this scenario. We consider a one-to-many setting where we want to improve routing between a single source and many destinations. This is the case where the algorithm power is most significant since, in the many-to-many setting, there is very little overlap between shortest paths, and thus not much improvement can be made over a basic greedy approach.1

We demonstrate, using real up-to-date Internet data, that the algorithm can suggest a relatively small set of relay nodes that can significantly reduce latency in current BGP routing.

The second scenario we consider is the TPC improvement example discussed above. In this case, we test the algorithm on a synthetic random graph, and we show that the general framework can be applied also to this case, resulting in very close-to-optimal results.

The third scenario addresses overlay Voice-over-IP (VoIP) applications such as Skype (http://www.skype.com), Google Talk (http://www.google.com/talk/), and others. Such applications are Becoming more and more popular offering IP telephone services for free, but they need abounded end-to-end delay (or latency) between any pair of users to maintain a reasonable service quality. We show that our scheme can be very useful also in this case, allowing applications to choose a smaller number of hubs, yet improving performance for many users.[6]

Note that the algorithmic model we use assumes a full knowledge of the underlying topology, the desired routing scheme, and the locations of the required endpoints. In general, the algorithm is used by the entity that needs the routing improvement and carries the cost of establishing and maintaining overlay nodes, using the best available topology information. For example, in the VoIP case, the VoIP application is establishing the overlay nodes, and thus the application can gain by using our approach.

The main contributions of this paper are as follows.

• We develop a general algorithmic framework that can be used in order to deal with efficient resource allocation in overlay routing.[7]

• We develop a nontrivial approximation algorithm and prove its properties. We demonstrate the actual benefit one can gain from using our scheme in three practical scenarios, namely BPG routing, TCP improvement, and VoIP applications.

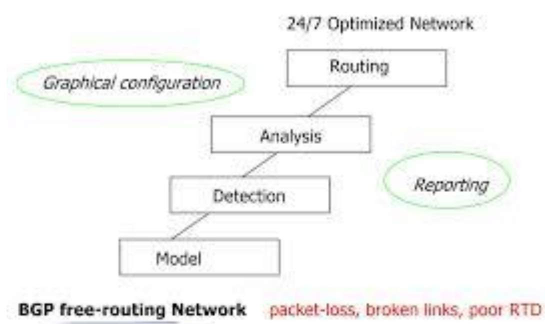

Figure 1: BGP Routing Network

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.[8] This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.[9]

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.
- What kind of output will be generating by the system?
  1. The encrypted file from the service provider
  2. The encrypted file from router after load balancing to the destination
  3. The encrypted file and decrypting the file in Destination, storing into destination

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer- based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

International Journal of Research
ISSN: 2348-6848 Vol-3, Special Issue-1
National Conference on Advanced Computing Technologies
Held on 21st July 2015 at Hyderabad city organized by **Global Research Academy,** Hyderabad, Telangana, India.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

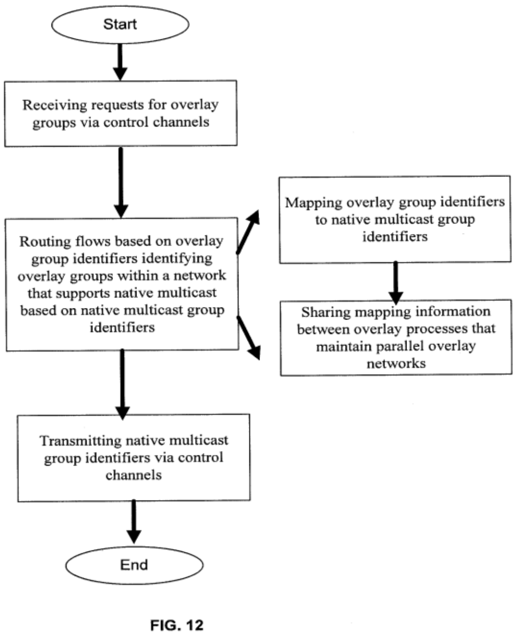

Figure 2: Data Flow Diagram for Overlay Routing Resource Allocation

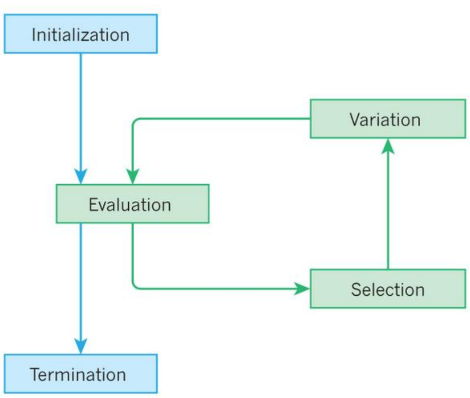

Figure 3: Process of Input Design

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.
- System Analysis
- In the existing system, the system is using overlay routing to improve network performance is motivated by many works that studied the inefficiency of varieties of networking architectures and applications. Analyzing a large set of data, Savage *et al.* explore the question: How "good" is Internet routing from a user's perspective considering round-trip time, packet loss rate, and bandwidth? They showed that in 30%–80% of the cases, there is an alternate routing path with better quality compared to the default routing path. In the current system and later in existing system, the authors show that TCP performance is strictly affected by the RTT. Thus, breaking a TCP connection into low-latency sub connections

improves the overall connection performance. In the existing system, the authors show that in many cases, routing paths in the Internet are inflated, and the actual length (in hops) of routing paths between clients is longer than the minimum hop distance between them. Using overlay routing to improve routing and network performance has been studied before in several works.[10]

In the existing system, the authors studied the routing inefficiency in the Internet and used an overlay routing in order to evaluate and study experimental techniques improving the network over the real environment. While the concept of using overlay routing to improve routing scheme was presented in this work, it did not deal with the deployment aspects and the optimization aspect of such infrastructure. A resilient overlay network (RON), which is architecture for application-layer overlay routing to be used on top of the existing Internet routing infrastructure, has been presented in the current system. Similar to our work, the main goal of this architecture is to replace the existing routing scheme, if necessary, using the overlay infrastructure. This work mainly focuses on the overlay infrastructure (monitoring and detecting routing problems, and maintaining the overlay system), and it does not consider the cost associated with the deployment of such system.

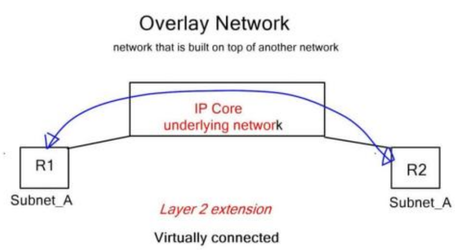

Figure 4: Overlay network

PROPOSED SYSTEM:

In the proposed system, the system concentrates on this point and study the minimum number of infrastructure nodes that need to be added in order to maintain a specific property in the overlay routing. In the shortest-path routing over the Internet BGP-based routing example, this question is mapped to: What is the minimum number of relay nodes that are needed in order to make the routing between a groups of autonomous systems (ASs) use the underlying shortest path between them? In the TCP performance example, this may translate to: What is the minimal number of relay nodes needed in order to make sure that for each TCP connection, there is a path between the connection endpoints for which every predefined round-trip time (RTT), there is an overlay node capable of TCP Piping?

Regardless of the specific implication in mind, we define a general optimization

problem called the Overlay Routing Resource Allocation (ORRA) problem and study its complexity. It turns out that the problem is NP-hard, and we present a nontrivial approximation algorithm for it.

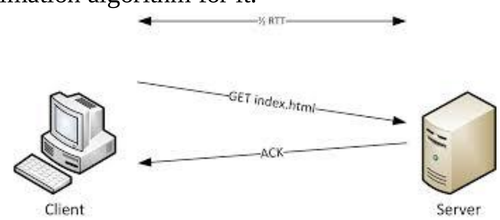

Figure 5: Round Trip Time

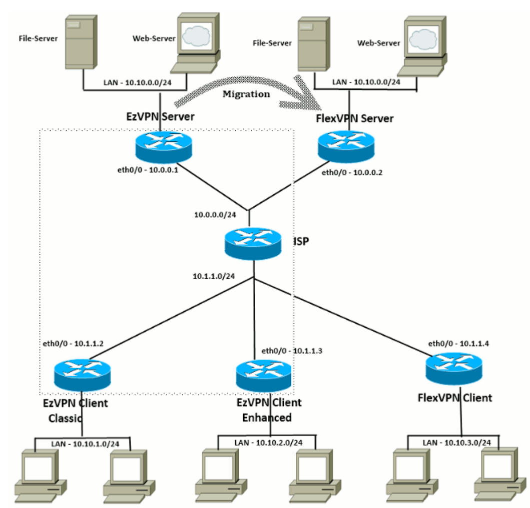

Figure 6: Proposed Overlay Network

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a

finished product It is the process of exercising software with the intent of ensuring that the
Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program.[11] Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.[12]

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. [13] Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.[14]

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.[15]
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IMPLEMENTATION

- **Service Provider**

  In this module, the Service Provider calculates the shortest path to Destination, The shortest-path routing over the Internet BGP-based router. The Service provider browses the required file and uploads their data files to the Specified End User (A, B, C, D) and with their DIP (Destination IP) of End User.

- **Overlay Router**

  The Overlay Router is responsible to route the file to the specified destination, the overlay routing scheme is the set of the shortest physical paths simplifies the execution of this system, and finding a minimal path to the destination using overlay routing, one can perform routing via shortest paths, the router is also responsible for Assigning the cost and also can view the cost of nodes with their tags From the node (from), To the node (to) and the cost.

- **BGP Router**

  The BGP Router is responsible to route the nodes using BGP routing, where the goal is to find a minimal number of relay node locations that can allow shortest-path routing between the source–destination pairs, BGP Router consider a one-to-many destination where we want to improve routing between a single source and many destinations. BGP routing table contains valid paths from its source to the entire set of nodes. BGP is also responsible for storing the possible path to destination, can view the recent routing path to destination with their tags Filename, Recent Path, Destination, DIP, Delay and date and time.

- **End User(Destination)**

  In this module, the End user (Node A, Node B, Node C, Node D) is responsible to receive the file from the Service Provider In the shortest-path routing between the source–destination nodes, the system consists of a one-to-many relationship. Where end User receives file from a single source to destination (Node A, Node B, Node C, Node D).

CONCLUSION

While using overlay routing to improve network performance was studied in the past by many works both practical and theoretical, very few of them consider the cost associated with the deployment of overlay infrastructure. In this paper, we addressed this fundamental problem developing an approximation algorithm to the problem. Rather than considering a customized Algorithm for a specific application or scenario, we suggested a general framework that fits a large set of overlay applications. Considering three different practical scenarios, we evaluated the Performance of the algorithm, showing that in practice the algorithm provides close-to-optimal results.

Many issues are left for further research. One interesting direction is an analytical study of the vertex cut used in the algorithm. It would be interesting to find properties of the underlay and overlay routing that assure a bound on the size of the cut. It would be also interesting to study the performance of our framework for other routing scenarios and to study issues related to actual implementation of the scheme.

In particular, the connection between the cost in terms of establishing overlay nodes and the benefit in terms of performance gain achieved due to the improved routing is not trivial, and it is interesting to investigate it. The business relationship between the different players in the various use cases is complex, and thus it is important to study the economical aspects of the scheme as well. For example, the one-to-many BGP routing scheme can be used by a large content provider in order to improve the user experience of its customers. The VoIP scheme can be used by VoIP services (such as Skype) to improve call quality of their customers. In both these cases, the exact translation of the service performance gain into actual revenue is not clear and can benefit from further research.

REFERENCES

[1] H. Pucha and Y. C. Hu, "Overlay TCP: Multi-hop overlay transport for high throughput transfers in the Internet," Purdue University, West Lafayette, IN, USA, Tech. Rep., 2005.

[2] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM SOSP*, 2001, pp. 131–145.

[3] S. Savage, T. A. A. Aggarawl, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan, "Detour: A case for informed internet routing and transport," *IEEE Micro*, vol. 19, no. 1, pp. 50–59, Jan.–Feb. 1999.

[4] R. Cohen and A. Shochot, "The "global-ISP" paradigm," Comput. *Netw.*, vol. 51, no. 8, pp. 1908–1921, 2007.

[5] L. Gao and F. Wang, "The extent of as path inflation by routing policies," in *Proc. IEEE GLOBECOM*, 2002, vol. 3, pp. 2180–2184.

[6] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of Internet path selection," in *Proc. ACM SIGCOMM*, 1999, pp. 289–299.

[7] R. Cohen and S. Ramanathan, "Using proxies to enhance TCP performance over hybrid fiber coaxial networks," *Comput. Commun*, vol. 20, no. 16, pp. 1502–1518, Jan. 1998.

[8] N. Spring, R. Mahajan, and T. Anderson, "The causes of path inflation," in *Proc. ACM SIGCOMM*, 2003, pp. 113–124.

[9] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The impact of routing policy on Internet paths," in *Proc. IEEE INFOCOM*, 2001, pp. 736–742.

[10] M. Cha, S. Moon, C.-D. Park, and A. Shaikh, "Placing relay nodes for intra-domain path diversity," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[11] M. Kodialam, T. V. Lakshman, and S. Sengupta,"Efficient and robust routing of highly variable traffic," in *Proc. HotNets III*, 2004.

[12] S. Roy, H. Pucha, Z. Zhang, Y. C. Hu, and L. Qiu, "On the placement of infrastructure overlay nodes," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1298–1311, Aug. 2009.

[13] L. Qiu, V. N. Padmanabhan, and G. M. Voelker "On the placement of Web server replicas," in *Proc. IEEE INFOCOM*, 2001, vol. 3, pp. 1587–1596.

[14] E. Cronin, S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, "Constrained mirror placement on the Internet," in *Proc. IEEE INFOCOM*, 2001, vol. 1, pp. 31–40.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.