Congestion Hot Spots identification using HITS and Improving Quality of Service

Dr. E. Ramaraj† and **Mrs. A. Padmapriya††**,

Madurai Kamaraj University, Madurai, Tamil Nadu, INDIA.† Alagappa University, Karaikudi, Tamil Nadu, INDIA.††

Summary
Traffic Engineering (TE) [1] broadly relates to optimization of the performance of a network. The Overlay approach [2] has been widely used by many service providers for Traffic Engineering (TE) in large Internet backbones. In the Overlay approach, logical connections are set up between edge nodes to form a full mesh virtual network on top of the physical topology. IP routing is then run over the virtual network, the Integrated approach [2] runs shortest path IP routing natively over the physical topology. Traffic engineering needs to determine the optimal routing of traffic over the existing network infrastructure by efficiently allocating resource in order to optimize traffic performance on an IP network. Traffic engineering objectives are achieved through carefully routing logical connections over the physical links.
Common objectives of traffic engineering include balancing traffic distribution across the network and avoiding congestion hot spots. The Bayesian Networks [3] are used to identify the congestion hot spots in the MPLS. Based on the Conditional Probability Distribution (CPD) the congestion hot spots are identified. Although it is implemented in operational networks, is has a number of well known scaling issues. We propose HITS method to identify the congestion hot spots in MPLS. Once the congestion hot spots are identified then the traffic can be distributed so that no link in the network is either over utilized or under utilized. With this method we can improve the quality of the routing there by avoiding congestion.
Key words:
Traffic Engineering, Congestion hot spots, Conditional Probability Distribution, HITS.

1. Introduction

IP routing typically uses shortest-path computation with some simple metrices such as hop count or delay. Although simplicity of this approach allows IP routing to scale very large networks, it does not make the best use of network resources [4]. In large Internet backbones, Service Providers typically have to explicitly manage the traffic flows in order to optimize the use of network resources. This process is often referred to as Traffic Engineering [1].

Traffic Engineering (TE) is defined as…. that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks….[1]. The goal of performance optimization of operational networks [4] is accomplished by routing traffic in a way to utilize network resources efficiently and reliably. A routing specifies how to route the traffic between each origin-destination pair across a network. The traffic sharing is applied in routing and allocating process to enhance the survivability of network [5].

1.1 Overlay Approach

Currently, most large Internet backbones employ the so-called Overlay approach [2] for traffic engineering. With this approach service providers establish logical connections between the edge nodes of backbones and then mapping these logical connections onto the physical topology [2]. Service providers can control the distribution of traffic over physical topology through carefully routing these logical connections over physical links. The optimal mapping between the logical connections and the physical links can be computed using a linear programming formulation.

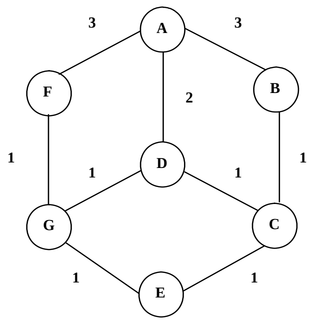
While the overlay approach has been widely implemented on current Internet backbones, it suffers the so called "N-Square" problem. As the size of the backbone network increases, the number of logical connections to be established will rise drastically, adding considerable management complexity and messaging overheads. Second, while IP routing runs over such a fully meshed virtual network, each edge node has to establish routing peering with (N-1) other nodes. This poses a significant problem to current IP routers as most of them can not support a large number of peers. Note that multiple logical connections may go over the same physical link. Thus, the breakdown of a single physical link may cause multiple logical connections to fail, and this will exaggerate the routing update load.

Manuscript received August 5, 2010
Manuscript revised August 20, 2010

1.2 Integrated Approach

Yujei Wang, Zheng Wang and Leah Zhang proposed a new approach called Integrated approach [2] that accomplishes traffic engineering objectives without full mesh overlaying. Instead of overlaying IP routing over the logical virtual network, the new approach runs shortest-path IP routing natively over the physical topology. It is theoretically proved that for any given traffic demands it is possible to select a set of link weights such that the shortest paths based on the selected link weights produce the same traffic distribution as that of the overlay approach with the assumption that traffic between the same source – destination pair can be split across multiple equal cost shortest paths, if exists.

Let us first illustrate with a simple example how the integrated approach works. Figure 1 shows a simple network topology, optimal link weights, and traffic demands. Each link has a capacity of 5 units and each demand needs bandwidth of 4 units. Although link capacities, link weights and traffic demands are unidirectional in IP networks, we assume they are bidirectional here for simplicity.



Traffic demands:
A → F : 4
A → B : 4
B → F : 4
A → E : 4

Figure. 1. Network Topology, Optimal link weights and Traffic Demands

To meet the traffic engineering objectives, we need to place the demands over the links in a way that the traffic distribution is balanced and there is no congestion or hot spots in the network. The optimal routes can be calculated using a linear programming formulation [6].

Demand A to B uses path AB, and demand A to F uses AF. Demand B to F has two paths. Half of the demand goes over BCEGF and the other half over BCEGF. Similarly for A to E. Half of the demand traverses path ADCE and the other half traverses through ADGE. BCDGH and BCFGH appear as equal-cost paths, so routing protocols such as OSPF will perform load sharing over them. Optimal routes and traffic distribution are given below.

A → B : 4 A-B
A → F : 4 A-F
B → F : 2 B-C-D-G-F
B → F : 2 B-C-E-G-F
A → E : 2 A-D-G-E
A → E : 2 A-D-E-E

With the integrated approach, there is no need to establish the logical connections. We simply calculate and set the appropriate link weights on the links, and the shortest-path routing will calculate the paths by it.

This Integrated approach has a number of advantages. First, it retains the simplicity of IP routing and requires little changes to the basic Internet architecture. Once the weights are calculated and set, the shortest-path routing protocol such as OSPF [7] can calculate the paths in the normal way, and packets are forwarded along the shortest paths. Second it eliminates the "N-Square" problem all together and reduces managing overheads in setting up logical connections. Common objectives of traffic engineering include balancing traffic distribution across the network and avoiding congestion hot spots. To meet the objectives, demands have to be placed over the links in order to achieve balanced traffic distribution and to avoid congestion hot spots in the network

This paper is organized as follows. In Section 2, we discuss about the background of the study and propose the HITS method for congestion hot spot identification. In Section 3, we apply this HITS method to integrated approach and describe the LP formulation for this method. Finally, we make the conclusion in section 4.

2. Background

2.1 Motivation – Bayesian Network Approach

A routing specifies how to route the traffic between each Origin-Destination pair across a network. The objective in designing a routing is to provide good quality of service and to optimize the utilization of network resources. Awduche et al [8] note that a distinctive function performed by Internet Traffic Engineering is the control and optimization of the routing function, to steer traffic through the network in the most effective way. Load balancing is an important approach to address network congestion problems resulting from inefficient resource allocation [4]. Currently the so-called Bayesian network approach [3] is used to congestion hot spots in MPLS. With this approach, in addition to the network topology, it is necessary to specify the parameters of the nodes in the network. For each node in the network the Conditional Probability Distribution (CPD) [9] is specified.

Optimal routes having minimum congestion likelihood can then be calculated using the following formula.

$$P(R = r \mid e) = \frac{P(e \mid R = r)P(R = r)}{P(e)} \quad (1)$$

where P(R = r | e) denotes the probability that random variable R has value r given evidence e. Instead of routing the demands over the congested routes, routes that suit the current traffic demand and capacity of the network are selected.

While the Bayesian approach has been implemented, it does have some scalability limitations. First, it suffers the so-called “N-Square” problem. To find the congestion hot spots among N nodes in a network, the CPD for all the N nodes must be specified. Only based on the CPD, the congestion likelihood can be calculated. Suppose if the network uses full mesh topology, each Conditional Probability Distribution Table will have (N-1) entries. Thus N*(N-1) CPD values must be specified to identify the congestion hot spots of the network with a size of N nodes. As the size of the backbone network increases, the number of CPD specifications will rise drastically, adding considerable management complexity and measuring overheads. Second, besides a major objective, like identifying and eliminating congestion hot spots, there are other factors to consider, such as path failure is not considered in Bayesian Network approach.

2.2 Link Mining : HITS Algorithm

Link mining is a newly emerging research area that is at the intersection of the work in link analysis [14; 10], hypertext and web mining [11], relational learning and inductive logic programming [13] and graph mining [12]. Link mining is an instance of multi-relational data mining in its broadest sense. Link mining encompasses a range of tasks including descriptive and predictive modeling. Both classification and clustering in linked relational domains require new data mining algorithms. Both the well known page rank measure [16] and hubs and authority scores [15] are based on the link structure of the web. These algorithms are based on the citation relation between web pages.

The collection *V* of hyperlinked pages can be viewed as a directed graph *G* = (*V*, *E*): the nodes correspond to the pages, and a directed edge $(p,q) \in E$ indicates the presence of a link from *p* to *q*. Then the out-degree of a node *p* is the number of nodes it has links to, and the in-degree of *p* is the number of nodes that have links to it.

A query to a search engine such as Alta Vista, retrieves a sub graph of the web whose nodes (pages) match the query. For any given query in a search engine, the pages can be categorized either as an authority or hub. Page *i* is called an ***authority*** for the query, if it contains valuable information on the subject. However, there is a second category of pages relevant to the process of finding the authoritative pages, called ***hubs***. Their role is to advertise the authoritative pages. They contain useful links towards the authoritative pages. In other words, hubs point the search engine in the "right direction".

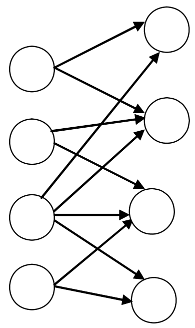


Figure. 2. Densely linked set of Hubs and Authorities

Jon Kleinberg's algorithm called HITS (Hyperlink Induced Topic Search) [15] identifies good authorities and hubs for a topic by assigning two numbers to a page *i*: an authority weight a_i, and a hub weight h_i. These weights are defined recursively. Pages with a higher a_i number are considered as being better authorities, and pages with a higher h_i number as being better hubs.

$$a_v = \sum_{u \to v \in E} h_u \quad (2)$$

$$h_u = \sum_{u \to v \in E} a_v \quad (3)$$

Hubs and authorities exhibit a mutually reinforcing relationship: a good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs. The idea is then to apply the two operations above alternatively until equilibrium values for the hub and authority weights are reached.

Let A be the adjacency matrix of the graph G, A^t be the transpose of the matrix and v be the authority weight vector and u be the hub weight vector. Then,

$$v = A^t.u \quad (4)$$

$$u = A.v \quad (5)$$

where,

$$u = \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_n \end{bmatrix} \quad \text{and} \quad v = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix}$$

If we consider the initial hub and authority weights of the nodes are,

$$u_0 = \begin{bmatrix} 1 \\ 1 \\ .. \\ 1 \end{bmatrix} \quad (6)$$

$$v_0 = A^t.\begin{bmatrix} 1 \\ 1 \\ .. \\ 1 \end{bmatrix} \quad (7)$$

then, after k steps we get the system

$$v_k = (A^t.A).v_{k-1} \quad (8)$$

$$u_k = (A.A^t).u_{k-1} \quad (9)$$

Let us consider the simple graph as shown in Figure 2. Since the graph here is rather small, the calculations can be done manually.

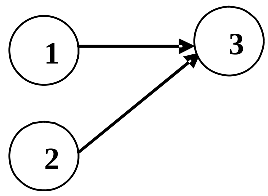


Figure 2. A simple example graph for HITS algorithm

The adjacency matrix of the graph is

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \text{ with transpose } A^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Assume the initial hub weight vector is:

$$u_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Then the authority weight vector can be computed by:

$$v_0 = A^t.u_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} . \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Then, the updated hub weight is:

$$u_0 = A.v_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

The node 3 is the most authoritative, since it is the only one with incoming edges, and that nodes 1 and 2 are equally important hubs. If we repeat the process further, we will only obtain scalar multiples of the vectors v and u computed at step 1. So the relative weights of the nodes remain the same. To meet the traffic engineering objectives, demands have to be placed over the links in order to achieve balanced traffic distribution and to avoid congestion hot spots in the network. The HITS algorithm [15] can be applied for identifying the congestion hot spots in a network. In the next section, the HITS method is proposed to measure the congestion hot spots in the Integrated approach [2].

3. Congestion hot spots identification using HITS and QoS

Enhancing the performance of an operational network at both traffic and the resource levels are major objectives of Traffic Engineering [1]. The goal of performance optimization of operational networks [4] is accomplished by routing traffic in a way to utilize network resources efficiently and reliably. To meet the objectives, demands have to be placed over the links in order to achieve balanced traffic distribution and to avoid congestion hot spots in the network. The optimal routes can be calculated using a linear programming formulation [7].

3.1 Network Representation

Given a network topology and existing traffic flows, the Hits algorithm can be used to identify the congested nodes in the network. When compared to the Bayesian network approach it requires only minimum efforts.

The nodes in the IP network can be considered as the pages of HITS algorithm. The Source nodes S are considered as Hubs and the Destination nodes D are considered as Authorities. The Intermediate nodes I between any source and destination may act either as a hub or as an authority. Before we present the theoretic results, let us first illustrate with a simple example how the HITS algorithm can be implemented in a network. Let us consider the network shown in Figure 1,

$$G = (V, E)$$
$$V = (A, B, C, D, E, G, F)$$
$$E = (\langle AB \rangle, \langle AF \rangle, \langle AD \rangle, \langle BC \rangle, \langle CD \rangle, \langle CE \rangle, \langle DC \rangle, \langle DG \rangle, \langle EG \rangle, \langle GE \rangle, \langle GF \rangle)$$

Here S={A, B,F}, D={B,E,F} and I={B,C,D,E,G}

$$A = \begin{bmatrix} 0\,1\;0\,1\;0\,0\,1 \\ 0\,0\,1\;0\,0\,0\,0 \\ 0\,0\,0\,1\;1\;0\,0 \\ 1\;0\,1\;0\,0\,1\;1 \\ 0\,0\,0\,0\,0\,1\;0 \\ 0\,0\,0\,0\,1\;0\,1 \\ 0\,0\,0\,0\,0\,0\,0 \end{bmatrix}$$

Then, using (6), (7), (8) and (9), we get,

$$a_0 = \begin{bmatrix} 8 \\ 6 \\ 10 \\ 10 \\ 9 \\ 10 \\ 19 \end{bmatrix} \qquad h_0 = \begin{bmatrix} 35 \\ 10 \\ 19 \\ 47 \\ 10 \\ 28 \\ 0 \end{bmatrix}$$

3.2 LP Formulation and Algorithm

Let digraph G = (V, E) represent the IP network, where V is the set of nodes and E is the set of links. Please note that the links and their capacities are directional, i.e. link i → j is considered different from link j → i, each with its own capacity. Let A represents the Adjacency matrix of the IP network. The hub and authority weights (h and a respectively) are represented by single column matrixes. For any given traffic flow S be the source of the flow, D be the destination of the flow and intermediate nodes are denoted by Ij. Then the congestion hot spots identification using HITS algorithm can be formulated as follows

$$\text{Min } \forall I_j \in I : \sum_j a_{Ij} + h_{Ij} \qquad (10)$$

Subject to

$$a_{Ij} >= 0 \qquad (11)$$

$$h_{Ij} >= 0 \qquad (12)$$

$$0 \le j \le n \qquad (13)$$

The objective function is to minimize congestion by selecting a route with nodes having least hub and authority weights. Here in equation (10) only intermediate nodes are taken into consideration. Equations (11) and (12) say that the authority and hub weights may be either zero or some positive value. It never takes negative values. Equation (13) says that the number of intermediate nodes in a path may be zero or less than the number of intermediate nodes in the path. The above LP formulation is not restricted for the given example network but they can be generalized to any network topology.

3.3 Algorithm for QoS using HITS

The process of finding congestion hot spots in a network using HITS can be done with the help of three functions.

Initialize_Network (V,E)

1. Intialize the adjacency matrix A of the graph G=(V,E)
   (a) For all node i varies from v_1 to v_n where $v_i \in V$
   (b) For all nodes j varies from v_1 to v_n where $v_i \in V$
   (c) Set A[i,j]=0
2. Update the adjacency matrix based on the links between nodes
   (a) For all edge $e \in E$
   (b) If $\exists edge : v_i \rightarrow v_j$, Set A[$v_i$,$v_j$]=1
   (c) Calculate the transpose of the adjacency matrix A^t
   (d) Initialize the elements of hub weights matrix h_0 to 1 for the nodes in the network and Calculate authority weights matrix $a_0 = A^t.h_0$

HITS_Calculation(G,A,A^t,a_0,h_0)

1. Calculate the new hub and authority weights using the formula

(i) $a_1 = (A^t.A).a_0$

(ii) $h_1 = (A.A^t).h_0$

Optimal_Routing_Using_HITS(A,a_1,h_1,S,D)

1. Find all the possible paths between the Source S and Destination D
2. Path $p_i = SI_1I_2I_3\ldots I_nD$, here $I_j \in I$ are the intermediate nodes in the path.
3. For all paths, Calculate

   HITS(PathP_i)= $\forall I_j \in I : \sum_{j=1}^{n} a_{Ij} + h_{Ij}$,

   where n represents the number of intermediate nodes in the path.
4. Select the path which minimizes the function HITS(Path P_i)

If there exists more than one path with same HITS(Path P_i) value, then traffic can be distributed evenly across the paths.

3.3 Results and Analysis

For example, let us consider in figure 1, there comes a new flow from E→A. Multiple paths are available from E→A. They are E→C→D→A, E→C→B→A, E→G→D→A and E→G→F→A etc. The objective function (10) will select the route E→C→B→A. Because, it is the one having least hub and authority weight when compared with the remaining routes. So the path E→C→B→A, will be selected for data transmission between node E and A.

TABLE I
HUB AND AUTHORITY CALCULATION

Feasible Paths	Summation of hub and authority values of Intermediate nodes
E-C-B-A	29+16=45
E-C-D-A	29+57=86
E-G-D-A	38+57=95
E-G-F-A	38+19=57

Path breakage or failure is concerned with the breakage of links between two nodes. This method holds good to any network. Since the hub and authority weights are obtained by iterations over time, we can easily identify the path failure. Whenever there is path breakage the adjacency matrix entry will become zero, whose impact is immediately known by the hub and authority weights of that particular node. Suppose if the path E→C→B→A fails, the path E→G→F→A can be used as alternate path. Among all the other paths, it is the one having least hub and authority weight summation.

4. Conclusion

We identified the congestion hot spots in a network with no knowledge of the link weights and shortest paths. We observe that the HITS algorithm effectively identifies the congestion hot spots in operational IP networks. It can identify the congestion hot spots with the help of the network topology alone.

Our theoretical results show that is strikes good result against even path breakage and loopy paths. Furthermore, it overcomes the draw back on “N-square” problem of Bayesian approach.

Similar approaches have been tried by some service providers in the past. When a link is experiencing congestion, service providers typically increase the weight for that link in the hope that traffic will be moved away from it. These experiments, however, were done based on simple heuristics. The lack of systematic strategy and comprehensive studies of link weight change impact has prevented it from being widely adopted in operational backbone networks. By using HITS method, for any network, the congestion hot spots can be calculated with minimum effort and requirements. With the HITS algorithm, we improve the quality of routing to almost perfection there by avoiding congestion hot spots in the network.

References

[1] D. Awduche et al "Overviews and principles of Internet Traffic Engineering", IETF, Internet draft, draft-ietf-tewg-principles-02.txt, Jan 2002.

[2] Yufei Wang, Zheng Wang and Leah Zhang, "Internet traffic engineering without full mesh overlaying", Infocom 2001.

[3] Raj P.H and SVK Raja, "Identifying Congestion hot spots in MPLS using Bayesian network", Asian J. Inform. Technol., 6 : 854-858, 2007.

[4] D. Awduche, J. Maledm, J. Agogbua, M. O'Dell and J. McManus, "Requirements for traffic engineering over MPLS", Internet draft, IETF, RFC 2702, Sep 1999, draft-awduche-mpls-traffic-eng-00.txt.

[5] Yao Cheng, Yue Wang, Yueming Lu, Yuefeng Ji, Yongmei Sun,, "Traffic Engineering-based Load Balancing Algorithm in GMPLS Networks", Key Lab. Of Opt. Commun. & Lightwave Technol., Beijiing Univ. of Posts & Telecommunication, Beijing, Dec 2008.

[6] Antonio Nucci and Konstantina Papagiannki, "Design, Measurement and Management of Large-Scale IP Networks – Bridging gap between theory and practice", Cambridge University Press, UK, 2009.

[7] Yufei Wang and Zheng Wang, "Explicit routing algorithms form internet traffic engineering", Proceedings of ICCCN'99, Sept 1999.

[8] ATM Forums, Private-Network-Network-Interface Specification Version 1.0 (PNNI 1.0), International Standart-of-pnni-oo55.000, March 1996.

[9] F. Jensen, "An Introduction to Bayesian Networks", UCL Press, 1996.

[10] R. Feldman. Link analysis: Current state of the art. In KDD-02 Tutorial, 2002.

[11] S. Chakrabarti. *Mining theWeb*. Morgan Kaufman, 2002.

[12] D. Cook and L. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[13] S. Dzeroski and N. Lavrac, editors. Relational Data Mining. Kluwer, Berlin, 2001.

[14] D. Jensen and H. Goldberg. *AAAI Fall Symposium on AI and Link Analysis*. AAAI Press, 1998.

[15] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bring order to the web. Technical report, Stanford University, 1998.

Dr. E. Ramaraj is with the Madurai Kamaraj University, Madurai, Tamil Nadu, INDIA as Technology Advisor. He has 25 years teaching experience and 6 years research experience. He has presented research papers in more than 40 national and international conferences and published more than 30 papers in national and international journals. His research areas include Data mining and Network security.

Mrs. A.Padmapriya is with Department of Computer Science & Engineering, Alagappa University, Karaikudi – 630 003, Tamil Nadu, India as Assistant Professor. She is currently pursuing Ph.D in Computer Science at Alagappa University. She has been a member of IACSIT. She published 5 papers in National and International conferences.