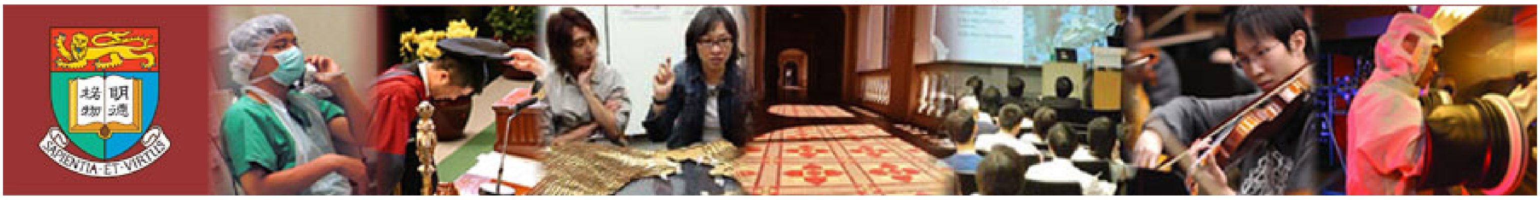


Title	**Inter-domain router placement and traffic engineering**
Author(s)	**Lam, F; Lau, WC; Li, VOK**
Citation	**Ieee International Conference On Communications, 2002, v. 4, p. 2443-2448**
Issued Date	**2002**
URL	**http://hdl.handle.net/10722/46300**
Rights	**Creative Commons: Attribution 3.0 Hong Kong License**

Inter-Domain Router Placement and Traffic Engineering

Fung Lam, Wing Cheong Lau and Victor O. K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
e-mail: {flam,wclau,vli}@eee.hku.hk

Abstract— **The Internet is organized as an interconnection of separate administrative domains called Autonomous Systems (AS). The Border Gateway Protocol (BGP) is the de facto standard for controlling the routing of traffic across different ASs. It supports scalable distribution of reachability and routing policy information among different ASs. In this paper, we study a network design problem which determines (1) the optimal placement of border router(s) within a domain and (2) the corresponding inter- and intra-domain traffic patterns within an AS. Practical constraints imposed by BGP and other standard shortest-path-based intra-domain routing protocols are considered. The problem is formulated as a variant of the uncapacitated network design problem (UNDP). While it is feasible to use a brute-force, integer-programming-based approach for tackling small instances of this problem, we have resorted to a dual-ascent approximation approach for mid/large-scale instances. The quality of the approximation approach is evaluated in terms of its computational efficiency and network cost sub-optimality. Sensitivity analysis w.r.t. various network/traffic parameters are also conducted. We then describe how one can apply our optimization results to better configure BGP as well as other intra-domain routing protocols. This serves as a first-step towards the auto-configuration of Internet routing protocols, BGP in particular, which is "well-known" for its tedious and error-prone configuration needs.**

I. INTRODUCTION

The explosive growth of the Internet has been accompanied by a wide range of internetworking problems related with routing. As the Internet is segregated into different autonomous systems (AS) with clear administrative and technical autonomy, the inter-domain routing protocols, such as the Border Gateway Protocol (BGP) [1], play an important role in exchanging reachability information among different ASs efficiently. BGP4 is the de facto standard for the dynamic routing protocol on the Internet [2]. Due to the increasing demand of the Internet, routing problems related to BGP become more significant, and have attracted considerable attention from routing researchers and practitioners. Routing behavior in the Internet is mainly due to routing protocol configurations, data traffic behavior, and network design. Most of the BGP research to date has been focused on stability and scalability issues of BGP [3], [4]. However, given the tedious and error-prone nature of BGP configurations, automation of such process is a critical and practical issue that has largely been under-stated. This paper considers an optimum network design problem with inter-domain routing constraints. Our results can be applied to auto-configure border routers to realize desirable traffic routing patterns.

Network design problems have found wide applications in various areas, such as the design of transportation networks and topological design of computer communication networks. Despite the extensive efforts in network design [5], situations with additional practical requirements continue to pose significant challenges to the optimization community [6], [7]. Most existing work on network design does not consider practical constraints imposed by the Internet routing protocols, e.g. the different treatments of intra- vs. inter-domain routing and the dominant requirement of shortest-path-based routing within a domain. In this paper, the optimized design can be realized via setting various routing protocol configuration parameters.

The balance of the paper is organized as follows. In Section II, we formulate our problem. Based on this formulation, optimal solutions for small instances of this problem can be found by using conventional branch-and-bound integer programming techniques. This exact solution approach is used to gain insight of the nature of the problem and serves as theoretical benchmarks for approximated solutions. However, scalability remains one of the major challenges of this problem. Section III introduces the methodology used to transform our formulation into the framework of a general uncapacitated network design problem so that it can be solved efficiently by applying the dual-ascent approximation approach [8], which is reviewed in Section IV. In Section V, the quality and the efficiency of the approximation approach is assessed. Finally, Section VI outlines how one can use our optimization results for BGP configuration purposes. The paper is concluded in Section VII.

II. PROBLEM DEFINITION AND FORMULATION

A. Motivation From A Practical Problem

Consider an enterprise that wants to set up its own network (as an AS). Connection of this AS to the outside world is provided via one or more Internet Service Providers (ISPs), where each ISP is a separate AS. Given (1) the set of internal locations, i.e. nodes in the enterprise network, (2) the list of internal nodes which can be used as border routers to the outside world via various ISPs, (3) the point-to-point traffic demands among the internal nodes as well as the nodal demands to/from destinations/sources outside the enterprise, the objective is to decide the required link placements, as well as their corresponding capacities, among internal network nodes and between border routers and ISPs, so that the traffic demands can be satisfied at a minimum cost. In accordance with common practice in IP networks, shortest-path-based routing is assumed within the AS[1].

Note that this design problem includes the optimal selection of border routers to support inter-domain routing via the ISPs. The design outcome will also specify the required routing pattern of various demands within the network. In general, based on the outcome of our network design, customized configuration of BGP and intra-domain routing protocols are required in order to realize the required optimal routing pattern.

This research is supported in part by the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-01/99).

[1] Via careful problem formulation, the link-weights used for shortest-path computation in intra-domain routing will be obtained as a by-product of our network design optimization. See Section V of this paper for details.

0-7803-7400-2/02/$17.00 © 2002 IEEE

B. Formulation

The aforementioned network design problem can be formulated as follows: Consider a network represented by a directed graph $G(V, E)$ where V is the set of nodes and E is the set of directed uncapacitated links. By uncapacitated, we mean one can put as much traffic as desired on a link at some finite cost. The network consists of three domains, namely source, transit and destination domains. The source node set, denoted by *VS*, is defined as the set of nodes in the source domain. These nodes correspond to the nodes in the aforementioned enterprise internal network. The transit node set, denoted by *VT*, is defined as the set of nodes in the transit domain. Here they represent the point-of-presence (POP) of the ISP candidates available to the enterprise. The destination node set, denoted by *VD*, contains nodes that correspond to external Internet sources or destinations outside the enterprise. We divide the link set E into three sets: the source link set, denoted by *SE*, is defined as the links within the source domain; the source-transit link set, denoted by *XE*, is defined as the links between the source domain and the transit domain; the transit-destination link set, denoted by *DE*, is defined as the links between the transit domain and the destination domain.

Let D be the set of point-to-point demands of the nodes in *VS*, where $d_{ij} \in D$ represents the demands from node i to node j. We permit multiple commodities that might represent different classes of data traffic, or the same class of data traffic but with different origins and destinations. Let H be the set of commodities and for each commodity $h \in H$, the amount of flow to be sent from the origin $O(h)$ to the destination $D(h)$ is denoted by d^h. Since all links are assumed to be uncapacitated, all the d^h units of commodity h could be sent along the same path without loss of generality. If each commodity has a single origin and single destination, $d^h = d_{ij}$ where $i = O(h)$ and $j = D(h)$.

We assume each ISP can provide full connectivity to all external sources/destinations in the Internet. Also, border routers are the only entry and/or exit points for inter-domain traffic. In line with common practice, routing through inter-domain links for intra-domain traffic is prohibited. The cost of a link is modeled by an affine function, which accounts for the fixed and variable cost components. The decision variables include two modeling discrete design choices and one modeling continuous flow decision, so as to minimize the total cost. A binary variable b_i is associated with every node $i \in VS$ to indicate whether (b_i=1) or not (b_i=0) node i is to serve as a border router. Every link $(i, j) \in E$ is associated with another binary variable x_{ij} to indicate whether (x_{ij}=1) or not (x_{ij}=0) the link (i, j) is established. For every commodity $h \in H$, we use a continuous variable $y_{ij}^h = [0, 1]$ to denote the normalized (w.r.t. d^h) traffic flows along the link $(i, j) \in E$.

We introduce a constant P_B to denote the cost of installing one border router. We assume that the ISPs will charge the customer for a fixed cost f_{ij} in using the link (i, j) connecting to them. Besides, the ISPs will charge a per-unit flow cost on each link, i.e. let ρ_{ij} denote this unit flow cost of using the link (i, j)[2]. Thus link cost function C_{ij} can be expressed as follows:

$$C_{ij} = f_{ij} x_{ij} + \rho_{ij} \sum_{h \in H} d^h y_{ij}^h \quad \forall (i, j) \in SE \cup XE$$

The general formulation of this problem is written as follows:

Minimize

$$\sum_{i \in VS} P_B b_i + \sum_{(i,j) \in SE \cup XE} f_{ij} x_{ij} + \sum_{(i,j) \in SE \cup XE} \sum_{h \in H} \rho_{ij} d^h y_{ij}^h$$

Subject to:

$$\sum_{(i,j) \in E} y_{ij}^h - \sum_{(k,i) \in E} y_{ki}^h = \begin{cases} 1, & \text{if } i = O(h) \\ -1, & \text{if } i = D(h) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, \forall h \in H \tag{1}$$

$$y_{ij}^h \leq x_{ij} \quad \forall (i, j) \in E, \forall h \in H \tag{2}$$

$$y_{ij}^h \geq 0 \quad \forall (i, j) \in E, \forall h \in H \tag{3}$$

$$y_{ij}^h = 0 \quad \forall (i, j) \in XE, O(h) \in VS, D(h) \in VS \tag{4}$$

$$b_i \geq x_{ij} \quad \forall (i, j) \in XE, i \in VS \tag{5}$$

$$b_j \geq x_{ij} \quad \forall (i, j) \in XE, j \in VS \tag{6}$$

$$b_i \text{ binary} \quad \forall i \in VS \tag{7}$$

$$x_{ij} \text{ binary} \quad \forall (i, j) \in E \tag{8}$$

The summations in the objective function represent the cost of installed border routers, the link setup cost and the flow cost respectively; constraint set *(1)* contains the network flow conservation equation; constraint set *(2)* is the flow constraints that restrict the flow of traffic on link (i, j) unless the link has been set up; constraint set *(3)* states the non-negativity of the continuous variables y_{ij}^h; constraint set *(4)* restricts the routing patterns of intra-domain traffic, in which the routing of intra-domain traffic through inter-domain links is forbidden; constraint sets *(5)* and *(6)* state that a border router can only be installed if the inter-domain link connecting it exists; constraint sets *(7)* and *(8)* ensure the discrete variables x_{ij} and b_i to assume binary values.

According to the inter-domain routing environment, our problem can be enhanced with additional forcing constraints. For example, we may remove any return paths for both the outbound and inbound traffic by adding the constraint sets *(9)* to *(12)*:

$$y_{ij}^h = 0 \quad \forall (i, j) \in XE, i \in VT, O(h) \in VS, D(h) \in VD \tag{9}$$

$$y_{ij}^h = 0 \quad \forall (i, j) \in DE, i \in VT, O(h) \in VD, D(h) \in VS \tag{10}$$

$$y_{ij}^h = 0 \quad \forall (i, j) \in XE, j \in VT, O(h) \in VD, D(h) \in VS \tag{11}$$

$$y_{ij}^h = 0 \quad \forall (i, j) \in DE, i \in VT, O(h) \in VD, D(h) \in VS \tag{12}$$

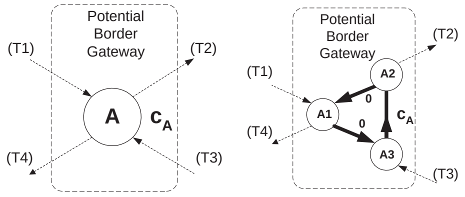
In order to design the source domain *VS* with known potential BGP router locations, the routing of traffic between the transit and destination domain will not be considered in the design, and thus we may further assume the traffic behind the transit domain is transparent to the source domain. This can be enforced by incorporating constraint *(13)* in the formulation.

$$y_{ij}^h = 0 \quad \forall (i, j) \in DE, i \in VT, D(h) \neq j \tag{13}$$

In order to enhance the efficiency of solving the problem, it is also possible to aggregate commodities by destination (or origin). In such an aggregate formulation, y_{ij}^h denotes the total flow on link (i, j) whose destination is node h, which corresponds to the aggregation over all source nodes of the commodities with destination h. y_{ij}^h in constraints (2) will be modified to the demand at a node i , which has been normalized by the total demand that belongs to the aggregated destination $D(h)$. Although

---

[2] To model IP networks that support Type-of-Service (TOS) routing, we can simply make the unit flow cost at each link (i, j) a function of the type-of-service of each commodity, i.e. ρ_{ij}^T where T is the TOS index of the demand of concern. Our formulation remains valid.



(a) (b)

Fig. 1. Potential border router: (a)Before node splitting ; (b)After node splitting

segregating the formulation by commodities and O-D pairs will produce more commodities and constraints, it is preferred computationally since it provides a tighter lower bound [8].

We observe that our formulation is similar to the well-studied uncapacitated network design problem (*UNDP*). Exceptions are: (1) the additional binary variables b_i's for the border router location selections and (2) the inter-domain routing constraints. Note that *UNDP* is known to be NP-hard [9], [10]. In the next section, we will show the equivalence of our problem with *UNDP* by introducing artificial nodes together with node-splitting techniques. As a result, our problem is also NP-hard.

III. PROBLEM TRANSFORMATION

Based on the above formulation, an optimal solution can be found by using conventional branch-and-bound approaches with standard integer programming solvers such as *CPLEX*. However, such brute-force approach has serious scalability problems. We have therefore resorted to approximation approaches for problems of practical interest. Fortunately, due to the close resemblance of our problem to *UNDP*, it is possible, via additional transformation, to use existing approximation techniques for *UNDP* to tackle our problem. In what follows, we describe how to transform our problem to *UNDP*.

A. Transformation of Border Router Costs

Our proposed transformation process includes the transformation of border router cost and elimination of inter-domain routing forcing constraints. Inter-domain traffic includes both inbound and outbound traffic, at least one of which will require the setting up of a border router. Since each border router incurs an extra setup cost, we can transform the node cost to link cost by applying the node splitting technique on every potential border router. Fig. 1(a) illustrates the flows of different types of traffic. Node A denotes the potential border router and c_A is its cost. In general, there are three different traffic flows associated with one potential border router, namely intra-domain traffic *(T1→T4)*, the traffic with both the origin and destination in the source domain; inter-domain outbound traffic *(T1→T2)*, the traffic that leaves from the source domain; and inter-domain inbound traffic *(T3→T4)*, the traffic that comes in the source domain from other domains.

We solve this problem by splitting one potential border router into three nodes, as shown in Fig. 1(b). The border router A is divided into $A1$, $A2$, and $A3$, respectively. Three extra directed links are added, namely $(A2, A1)$, $(A1, A3)$, and $(A3, A2)$. The node cost c_A is completely transferred to the link cost of link $(A3, A2)$, while the other two links are assigned zero link cost. Therefore, the intra-domain traffic will come in from link $(T1)$ and leave through link $(T4)$ without incurring any cost on the link $(A3, A2)$. The inter-domain traffic requires the setup of a border router to allow exchanges of traffic. The only path that can leave from the source domain is *A1→A3→A2*, and that for the inbound traffic is *A3→A2→A1*. Both of these traffic will pass through the link $(A3, A2)$ once and incurs the cost c_A.

As a result, this splitting mechanism can successfully transform the border router cost to a link cost and provide a dedicated path for the inter-domain traffic to pass through, such that the link cost (or the border router cost) will be included in the objective function whenever a border router is set up. This is analogous to the functions imposed by constraint sets *(5)* and *(6)*. Besides, this transformation allows the replacement of the binary variable b_A for border router A by another binary variable x_{A1A2} for an additional link $(A1, A2)$, which therefore further eliminates the constraint set *(7)* from the formulation. Although this transformation will increase the total number of nodes and links in the network by $2N_B$ and $3N_B$ respectively, where N_B is the number of potential border routers. The border router binary variable can be eliminated so as to fit the problem in the general *UNDP* framework.

B. Elimination of Inter-Domain Constraints

After the potential border router cost is transformed, the only difference between our model and the general *UNDP* is the constraint sets *(4)*, *(9)*-*(12)* , which are related with inter-domain routing. The purpose of incorporating these constraints is to restrict the inter-domain traffic patterns, and therefore we may make use of the flow cost $\rho_{ij} d^h$ in the network. These constraints can be realized in the network by adjusting the cost of the related inter-domain links. We propose to increase the costs of these links to a very large value M (e.g. $M \geq \sum_{(i,j) \in SE} \sum_{h \in H} \rho_{ij} d^h$), such that the overall cost of building up the network becomes very high if the design includes any of those links. Since our formulation is a minimization problem, these links will automatically be dropped.

IV. APPROXIMATED SOLUTION VIA DUAL-ASCENT

In this section, we briefly review a prominent approximation approach, namely, the dual-ascent approach, which has been successfully applied in finding good solutions for large-scale instances of *UNDP*. Several highly successful dual-ascent procedures can be found in the literature [8], [9], [11]. The quality of this approximation approach on our particular problem will be assessed by comparing the results with that obtained via the exact solution approach.

The dual-ascent is an integer program solution methodology that extends the primal-dual method for linear programs. It takes advantage of the special dual structure that a problem has, especially an *UNDP*, and provides a structural way to solve it rapidly. Several dual-ascent procedures were invented, one common property among them is the solving process often starts from some dual feasible solutions, and always retains the dual feasibility through the algorithm. Such method can generate a tight lower bound and terminate with a primal feasible solution.

After applying the transformations in Section III, our problem (primal problem) is reduced to a general *UNDP* problem.

By relaxing the binary constraint set *(8)*, the dual version for our problem is formed as follows:

Maximize

$$\sum_{h \in H} v^h_{D(h)}$$

Subject to:

$$v^h_j - v^h_i \leq \rho_{ij} d^h + w^h_{ij} \qquad \forall (i,j) \in E, \forall h \in H \tag{14}$$

$$\sum_{h \in H} w^h_{ij} \leq f_{ij} \qquad \forall (i,j) \in E \tag{15}$$

$$w^h_{ij} \geq 0 \qquad \forall (i,j) \in E, \forall h \in H \tag{16}$$

The resulting dual variable v^h_i corresponds to the flow conservation equation *(1)*. This can be interpreted as the node potential of node i in commodity h, which represents the shortest-path distance from the origin $O(h)$ to node i. Another dual variable, w^h_{ij}, corresponds to flow constraint *(2)*. The dual objective should be composed of two terms, $v^h_{O(h)}$ and $v^h_{D(h)}$. We arbitrary set $v^h_{O(h)} = 0$ since one of the flow conservation equations *(2)* is redundant for each commodity *h*∈ *H*. Via careful observations, given any vector $w = \{w^h_{ij}\}$ that satisfies the dual constraint set *(15)*, our dual problem can be decomposed into subproblems by commodity, each of which is the dual form of a shortest-path problem from origin $O(h)$ to the destination $D(h)$ w.r.t. the modified link length $\rho_{ij} d^h + w^h_{ij}$.

Since the dual objective is to maximize the destination node potentials, the procedure should then be motivated by maximizing the shortest-path distance between $O(h)$ and $D(h)$ in all commodities. This can be done if each of the aforementioned sub-problems is solved to its optimal. Aiming to increase the shortest-path distance in each commodity, we may increase one or more w^h_{ij} iteratively, and at the same time retain the dual feasibility, i.e. satisfying the dual constraints set *(15)*. This is called the dual-ascent procedure. Different allocating schemes for increasing w^h_{ij} gives rise to different implementations of the dual-ascent method. The labeling method of Balakrishnan et al. [8] suggests to increase the shortest-path distance between $O(h)$ and $D(h)$ by introducing a slack variable s_{ij} (the unabsorbed fixed charge) for each of the links *(i,j)*∈ *E*, and the suggested iterative procedure seeks to allocate the slacks selectively in each commodity, in order to increase the overall dual objective.

V. IMPLEMENTATION AND RESULTS

A. Problem Generation Procedures

We are going to construct a network topology, which consists of source, transit and destination domains, by using *GT-ITM*. The source domain *(VS)* is generated either by using the random flat approach or transit-stub hierarchical method. The transit and destination domains *(VT and VD)* are generated by the transit-stub hierarchical method, in which the transit-stub relationship represents exactly the same as that between the transit and destination domains. The links are unidirectional and the average node degree is set in the range of 4 to 5.

In order to connect these domains together, we first select one-third of nodes from *VS* as potential border routers, and then connect them fully with the transit nodes in *VT*. The variable cost for unit flow, ρ_{ij}, is set equal to the Euclidean length of the link (i,j). The fixed charge f_{ij} is a user specified multiple of the variable cost, we name it as *FC/VC* ratio. The border router cost P_B is another user specified multiple of fixed cost, we refer it as *BC/FC* ratio. In the process of traffic generation, we select one third of the nodes to generate both the intra- and inter-domain traffic; intra-domain traffic was generated by randomly selecting an O-D pair in *VS* and assigning a random positive integer less than 6; Similarly, O-D pairs were randomly selected from two different domains (*VS* and *VD*) and a random positive integer less than 16 was assigned for the inter-domain traffic. All traffic assignments are unidirectional.

For our experiments, we generated test problems in 7 sizes, ranging from 94 nodes, 712 links and 465 commodities to 804 nodes, 5423 links and 1778 commodities. Table I shows the dimensions of the networks.

TABLE I

NETWORK DIMENSIONS OF TEST PROBLEMS.

Problem ID	**Node**			**Edge**			**Commodity**
	VS	VD	VT	SE	XE	DE	
1	10	4	80	48	24	640	465
2	20	4	80	106	48	640	1004
3	50	4	80	262	120	640	2833
4	100	4	80	490	240	640	6870
5	150	4	80	752	360	640	12163
6	100	4	500	332	80	4000	503
7	300	4	500	1183	240	4000	1778

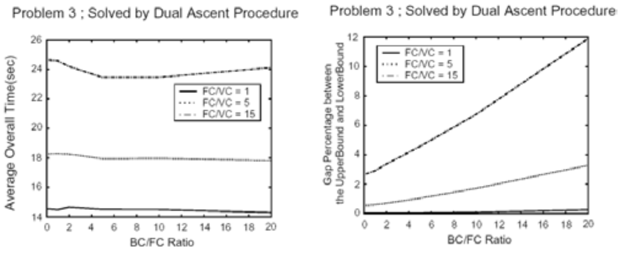
We used two approaches to generate the testing topologies: problems 1 to 5 used flat graphs as the source domain VS, while problems 6 and 7 used a multi-tier structure. The major difference between the two approaches are (1) the number of nodes in the network and (2) the portion of nodes generating the traffic, which determines the total commodities in the network. Usually a network that connects a large number of locations is modeled by a hierarchical structure. The results for problems 6 and 7 can, therefore, provide insight of the performance of applying our solution method in realistic networks.

B. Implementation

By assuming every ISP (transit node) can provide full connectivity to any external nodes, we further reduce the number of commodities and thus the size of the problem by aggregating all external destination nodes into one virtual node. We have implemented the general formulation *(1)-(8)* with additional forcing constraints *(9)-(13)*, and solved the problems by two approaches. For the exact solution approach, we solved our problem using *AMPL* modeling script and *CPLEX* solver. Besides the optimal solutions, we also solved the problems repeatedly with different tolerance values, such that we may provide a fair comparison between the solutions by *CPLEX* and the approximated solution. For the approximation approach, we implemented the dual-ascent procedure based on the labeling method suggested in [8] to solve our problems. Results were compared with respect to time performance and solution quality.

C. Computational Results

Computations were done on a SUN Ultra10 machine with an UltraSparc-IIi 440Mhz CPU. The available physical memory was around 300MB with a total swap space of around 2GB. We tested five instances for each of the problems listed in Table I, with *FC/VC* ranging from 1 to 15, and *BC/FC* from 0 to 20. The results show that the computation time generally increases with



(a) Average time (b) Gap percentage

Fig. 2. Results of solving problem 3 by the dual-ascent procedure

the network size and the cost ratios. Fig. 2(a) shows the performance of using the dual-ascent procedure to solve problem 3. Due to the limited number of potential border routers in the problems, the cost ratio *BC/FC* does not affect the computation time as significantly as the *FC/VC* ratio in both approaches.

However, the *BC/FC* ratio does have impact on the solution quality of the dual-ascent procedure, which is shown in Fig. 2(b). The gap between the feasible solution and the dual solution increases with a higher *BC/FC* ratio. This can be explained as follows: in the case of very small border router cost (and thus small *BC/FC* ratio), the optimal design should include as many border routers as possible so that the outbound (inbound) inter-domain traffic from (to) an internal node can take the shortest path to exit (enter) the AS through the nearest border router. Since the initial solution selected for the dual-ascent procedure is always generated by initializing all the nodes with a potential using the shortest-path algorithm, the initial solution is expected to be very similar to the final optimal solution in the cases of small border router cost. On the other hand, with increasing border router cost, the optimal solution is expected to increasingly deviate from a shortest-path-based initial solution and therefore lead to poorer performance of the dual-ascent heuristics. Fortunately, the practical range of the *BC/FC* ratio is about 1 to 10 [12], and we expect the dual-ascent approach to perform reasonably well in practice.

Table II shows the comparison between results from *CPLEX* and the dual-ascent procedure. We present the results as the average of five instances for each problem as described earlier, with realistic *FC/VC* ratio set to 1, 5 and 15, and the *BC/FC* ratio set to 5. The third column in Table II shows the computation time and the resulting gap when the solution tolerance was set to 5% in *CPLEX* solver, which means the branch and bound procedure will stop when a feasible solution yields less than 5% gap. The fourth column shows the best results obtained by solving the problems to optimal. The results represent the solving time and the corresponding gap upon completion or before the process was terminated due to memory exhaustion. No feasible solution could be obtained for problems 5 and 7 before the termination of the process. Note that the time needed for a tighter solution gap, especially in large-scale problems, is much higher than solving a problem with relaxed tolerance. The last column shows the corresponding results provided by the dual-ascent procedure.

TABLE II

COMPARISON OF RESULTS BETWEEN THE TWO SOLUTION APPROACHES.

Problem ID	$\frac{FC}{VC}$	CPLEX (5% Tolerance) Time (sec)	CPLEX (5% Tolerance) Gap (%)	CPLEX (Best Solution) Time (sec)	CPLEX (Best Solution) Gap (%)	Dual-Ascent Time (sec)	Dual-Ascent Gap (%)
1	1	0.36	0.77	45	0	0.12	0.01
	5	0.51	2.28	96.3	0	0.14	0.34
	15	0.48	4.02	30.77	0	0.15	1.75
2	1	4.43	1.19	4899	0	0.67	0.01
	5	4.11	0.83	151845	0	0.84	0.49
	15	6.51	3.06	62527	0.09	1.08	1.11
3	1	111.86	1.42	157800	0.13	14.52	0.03
	5	122.19	0.83	101880	0.79	17.95	0.47
	15	156.25	4.74	120000	4.7	23.45	1.33
4	1	803.6	0.66	72000	0.15	225.26	0.05
	5	811.4	2.53	111600	0.45	241.52	0.62
	15	1172.4	2.55	69000	3.58	278.68	2.17
5	1	N/A	N/A	N/A	N/A	1150.07	0.04
	5	N/A	N/A	N/A	N/A	1229.07	0.47
	15	N/A	N/A	N/A	N/A	1392.64	2.25
6	1	80.56	0.62	9010	0	9.155	0.19
	5	99.19	3.71	110000	0.25	14.905	1.46
	15	165.93	4.5	88000	4.2	24.89	6.43
7	1	N/A	N/A	N/A	N/A	207.77	0.3
	5	N/A	N/A	N/A	N/A	613.16	2.3
	15	N/A	N/A	N/A	N/A	1595.87	9.14

TABLE III

COMPARISON OF OBJECTIVES FROM THE TWO SOLUTION APPROACHES.

Problem ID	FC/VC	OBJ_{da}/OBJ_{cplex}[3]
1	1	1.000
	5	1.003
	15	1.015
2	1	1.000
	5	1.007
	15	1.004
3	1	0.994
	5	1.002
	15	0.977
4	1	1.000
	5	1.016
	15	1.006
6	1	1.002
	5	1.011
	15	1.042

Although the problems 5 and 7 could not be solved by *CPLEX* due to exhaustive memory usage during computation, the dual-ascent procedure could find the feasible solution reasonably fast within the limited computing resources. The results show that the dual-ascent procedure generally provides a feasible network design much faster than the best solution provided by *CPLEX*, and even if the tolerance is relaxed. The gap percentage for each solution approach is included to show its sensitivity. Except in problems 6 and 7, the dual-ascent procedure also gives a gap less than 3% in most problems. This indicates that it can solve a large-scale network design problem fast while providing a tight lower bound.

Since the gap for the *CPLEX* approach and the dual-ascent one are based on different lower-bounds, a smaller gap does not necessarily imply a better (lower cost) objective. To compare the solution quality of the two approaches, we define OBJ_{da}/OBJ_{cplex} as the ratio between the final value of the objective function from the dual-ascent procedure and that from *CPLEX*. Table III shows the corresponding ratios. Here, the OBJ_{cplex} is the best objective obtained via *CPLEX*. Most of the results show a ratio close to one, which indicates the solutions provided by the dual-ascent procedure are indeed very close to the one obtained from *CPLEX*. To summarize, the dual-ascent

---

[3] OBJ_{da} is the final value of the objective result from the dual-ascent procedure, while OBJ_{cplex} is the best objective obtained when solving the problems to optimal using *CPLEX*.

procedure can solve our problem effectively and generate high-quality solutions even for large-scale networks.

VI. BGP POLICY CONFIGURATIONS

In this section, we outline how to configure various BGP attributes based on the outcome of our network design optimization. The emphasis is on the configuration of crucial protocol parameters that control inbound and outbound traffic routes. Such attributes include *Multi-Exits Discriminator(MED)*, *LOCAL-PREF* and *AS-PATH* [1]. There are several assumptions before configuring appropriate policies. Since the problem we are solving is multi-homing to one or more providers (i.e. a customer is connected to multiple providers), we assume the ISPs will not violate the pre-configured BGP policies, such as following the *MEDs* announced by the customers. We also assume the provider will deliver traffic to the customer via the link closer to the destination (shortest AS Path). However, since different ISPs will use different policies and result in different basis in traffic routing, we assume that announcing the non-transitive *MED* attribute to different ISPs will not have any effects.

In practice, the network performance and the configurations of inter-domain routing protocols are closely coupled. The actual traffic routing between its source and destination is tightly constrained by intra-domain as well as inter-domain routing protocol. For intra-domain routing, shortest-path-based approach is commonly adopted. The major task for intra-domain routing configuration is therefore the determination of link-weight used by the shortest-path algorithm. Fortunately, due to our choice of the uncapacitated network design formulation, a true optimal set of link-weight will be generated as a by-product of the optimization. This is because in the optimal solution of our design problem, each demand should be routed along the shortest path using the per-unit flow cost, i.e. ρ_{ij}, as link metric. In other words, we can simply apply the ρ_{ij}'s as the shortest-path weights for the intra-domain routing protocol, in order to realize the designed optimal traffic routing pattern.

For the inter-domain routing, as the dual-ascent procedure provides a dedicated path for each of the commodities, the straightforward method to configure the desired policy for the inter-domain traffic is by using route filtering. An AS can control the routes of its outbound traffic uses by specifying the routes it accepts from its neighbors. Conversely, it can also restrict or accept the inbound traffic from other neighbors by specifying the list of routes it advertises to its neighbors.

Besides such an intensive route filtering method, there are other possible ways to enforce the desired policy through attribute manipulations. We considered the outbound and inbound traffic separately. For the outbound traffic, once the traffic reaches the appropriate border router, we are able to decide the ISP, through which it can reach the destination with minimum cost. This decision in fact has already been decided in our solution. The procedure ends with a solution that includes the shortest distance of one border router to an ISP. This problem can therefore be solved by using the *LOCAL-PREF* attribute, whose value is inversely proportional to the inter-domain link cost in our problem. Now, the problem becomes how to make sure the desired outbound traffic can reach the appropriate border router. There are two possible results for the traffic patterns: the traffic patterns will follow the intra-domain metrics (e.g. *OSPF*) to one border router, or otherwise. In both cases, we may adjust the corresponding *LOCAL-PREF* attribute by setting a higher preference value to the desired exit. Although there are multiple border routers, we may assign a large preference value for the desired exit while keeping others small.

Finally, the inbound traffic is mainly affected by the way customers advertise their networks to providers. In our multi-homing scenario, the non-transitive *MED* attribute can only affect the behavior of one provider, but not between providers. We propose inserting redundant entries in the *AS-PATH* to affect the AS path length, and hence affecting the provider's decisions dynamically. Moreover, since our solution will generate path information for each commodity, we may enforce the inbound traffic pattern by selectively advertising the source nodes to the corresponding transit nodes identified in the design.

VII. CONCLUSION

In this paper, a problem formulation with practical configuration constraints imposed by inter-domain routing (BGP) was proposed. Since our problem is NP-hard, the exact solution method by *CPLEX* was proved not scalable, and therefore, an approximation method was used to solve the problem. The formulation was first restructured into a general *UNDP* by splitting the potential border routers and explicitly setting the flow costs to enforce the corresponding inter-domain forcing constraints. We solved the formulation by the dual-ascent procedure, and the results showed that it could solve our problem effectively and generate 'good' solutions to large-scale network design problems. Due to our choice of the formulation and approximation methods, the optimal design solution can be used to guide or even automate the BGP configuration operations.

REFERENCES

[1] Y. Rekhteran and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, DDN Network Information Center, March 1995.

[2] B. Halabi and D. McPherson, *Internet Routing Architectures, 2nd ed.*, Cisco Press, 2000.

[3] R. Dube, "A Comparison of Scaling Techniques for BGP," *ACM Computer Communication Review*, vol. 29, no. 3, July 1999.

[4] R. Govindan and A. Reddy, "An Analysis of Inter-Domain Topology and Route Stability," *in INFOCOMM'97*, 1997.

[5] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani, "Chapter 18 - Network design," *Annotated Bibliographies in Combinatorial Optimization*, John Wiley & Sons, 1997.

[6] M. Lee and J. Yee, "A Logical Topology and Discrete Capacity Assignment Algorithm for Reconfigurable Networks," *Operations Research*, vol. 43, pp. 102-116, 1995.

[7] T. Ng and D. Hoang, "Joint Optimization of Capacity and Flow Assignment in a Packet-switched Communications Networks," *IEEE Transaction on Communications*, vol. 35, pp. 202-209, 1987.

[8] A. Balakrishnan, T. L. Magnanti, and R. T. Wong, "A Dual-ascent Algorithm for Large Scale Uncapacitated Network Design Problem," *Operations Research*, vol. 37, pp. 716-740, 1989.

[9] R. T. Wong, "Dual Ascent Approach for Steiner Tree Problems on a Directed Graph," *Mathematical Programming*, vol. 28, pp. 271-287, 1984.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*, W. H. Freeman, 1979.

[11] D. Erlenkotter, "A dual-based procedure for uncapacitated facility location," *Operations Research*, vol. 26, no. 6, pp. 992-1009, 1978.

[12] F. Lam, W. C. Lau, and V. O. K. Li, "Internet Inter-Domain Traffic Engineering and Optimization," *Report No. 2001-08-02*, Multimedia Networking Lab, the University of Hong Kong, August 2001.