Robust Egress Interdomain Traffic Engineering

Jian Qiu and Lixin Gao
Department of ECE, University of Massachusetts, Amherst, MA 01003
{jqiu,lgao}@ecs.umass.edu

***Abstract*— BGP is traditionally configured to implement traffic engineering objectives without considering potential network dynamics. This might result in undesirable traffic distribution when network failures occur. In this paper, we present algorithms for interdomain traffic engineering that achieve the interdomain traffic engineering objectives under network failures. That is, we aim to configure routing policies so that traffic is distributed evenly. More importantly, the configuration is robust in the sense that it is able to achieve the specified traffic engineering goals despite network failures. We first investigate the coarse-grained robust configurations. The derived configuration can achieve optimal robust traffic engineering objectives for most network failures. Further, we develop a greedy algorithm to derive robust BGP configuration for any traffic distribution and link capacities. We use simulations to evaluate the robustness of the derived BGP configurations by applying the algorithm to both transit and stub ASs under realistic traffic demands. Our results show that the derived BGP configurations can improve the default configuration significantly in terms of achieving the robust traffic engineering objectives. Furthermore, our algorithm achieves robust traffic engineering goals without diminishing other routing objectives.**

I. Introduction

The Internet is divided into more than 20,000 independently administrated Autonomous Systems (ASs). The interior Gateway Protocols (IGPs) maintain and exchange routing information within each AS, while BGP (Border Gateway Protocol) [22] is the *de facto* interdomain routing protocol interconnecting the ASs. One of the key features of BGP is the policy-based routing. Routing policies are specified in order to conform to the contractual agreements between ASs, as well as to achieve traffic engineering goals of an AS. However, specifying high-level goals such as traffic engineering are challenging. The primitive BGP configuration language is not expressive enough for fulfilling high-level routing objectives [13]. Further, BGP configurations are typically specified without considering the potential network failures or dynamics. However, network failures are part of the daily routines in the operational networks [16]. For example, Bonaventure *et al* observed 9452 eBGP link failures in three months in a transit ISP and 82% of them lasted no more than 3 minutes [4]. The network dynamics are so frequent and transient that it is infeasible to adjust BGP configurations whenever a network failure happens. Nonetheless, failing to adapt to the changing network states might degrade network performance and even make some networks unreachable.

In this paper, we explore the potential of BGP configurations to be robust to the network failures while preserving the interdomain traffic engineering goals. More specifically, we derive BGP configurations that are able to balance load at AS egress links during the occurrence of egress link failures. We formulate the robust interdomain traffic engineering problem as an optimization problem which minimizes the maximal link load and the maximal load difference among egress links given any combination of the failures of a small number of links. We present algorithms for deriving the robust BGP configuration for various settings. The major contributions are summarized as follows:

- We formulate the coarse-grained interdomain traffic engineering as a combinatorial problem and analytically show that the optimally robust configuration can be derived if no more than two egress links fail simultaneously.
- We develop a greedy algorithm to derive BGP configuration robust to egress link failures for the general settings with any traffic demands and egress link capacities. The simulation results show that the derived configurations outperform the default ones significantly. More importantly, the solutions not only achieve the robust traffic engineering goals but also preserve the constraints imposed by other routing objectives.

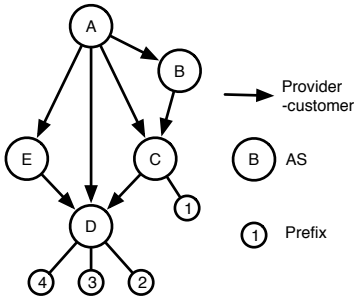

Fig. 1. Example topology

The rest of the paper is organized as follows. The problem of robust interdomain traffic engineering is formulated in Section 2. We derive the robust configurations for coarse-grained load balancing in Section 3. Section 4 provides a greedy algorithm that derives the robust BGP configurations for any traffic demands and link capacities. Simulations are also performed in Section 4 to evaluate the performance of the greedy algorithm. The related work is reviewed in Section 5. Finally, section 6 concludes the paper.

II. Problem Formulation

We first use a simple example to show the necessity and complexity to achieve robust egress interdomain traffic engineering. As shown in Figure 1, prefixes 1, 2, 3 and 4 are four

1-4244-0593-9/06/$20.00 ©2006 IEEE

prefixes hosting some popular content providers. AS A wants to distribute its traffic to them among the four egress links AB, AC, AD and AE respectively. Suppose that the traffic demand to each to the four destinations is approximately d and each egress links is able to accommodate traffic demand $2d$. If there is no link failure, the load balancing configuration is trivial. That is, each destination is assigned to a different egress link. However, suppose that some links fail, if we want to maintain the reachability to the prefixes and also ensure that the links are not overloaded and preserve the load balancing among the rest of the links, we need to be careful to configure the alternative egress links for each destination prefix. For example, suppose the four prefixes 1, 2, 3 and 4 are assigned to AB, AC, AD and AE respectively. In order to bear the failure of any two links, we should assign each of the four prefixes a backup egress link, e.g. AC, AD, AE and AB respectively, and a second backup link, e.g. AD, AE, AB and AC respectively. However, this configuration does not ensure load balancing for any two link failure. For example, if AD and AE fail at the same time, link AB will be overloaded. Instead, if we set the first alternative egress links for 1, 2, 3 and 4 to AC, AB, AE and AD and the second alternative egress links for 2, 3 and 4 to AE, AB and AC, no link will become overloaded for any one or two egress link failure (note that if both AB and AC failed, prefix 1 would be unreachable). This example shows that, to some extents, we can achieve the robust traffic engineering goals by carefully ranking the egress links for each prefix. However, it is not trivial to do so even in a toy setting like this example. Therefore, it is worthy of exploring the systematic methodologies to derive the robust configurations preserving the interdomain traffic engineering goals during network failures.

In this paper, we formally propose the problem of deriving the BGP configuration that achieve interdomain traffic engineering goals despite transient egress link failures. Meanwhile, Nucci *et al* formulate the robust IGP configurations for intradomain traffic in [17].

A. Problem Settings and Constraints

At first, we present the input of the problem.

Formally, suppose that an AS has m egress links, denoted by $\mathbf{E} = \{1, \ldots, m\}$, and it distributes the outbound traffic to n destination prefixes $\mathbf{P} = \{1, \ldots, n\}$ in other ASs over these egress links. Each link i has a link capacity c_i and each prefix j has a traffic demand d_j. The link capacity vector for $\mathbf{E}$ and the traffic demand vector for $\mathbf{P}$ are represented by $\mathbf{C}$ and $\mathbf{D}$ respectively.

Due to network topology constraints and the AS's import/export routing policies, the routes to a prefix j might not be available at some egress links. The set of prefixes which are reachable through egress link i is denoted by $\mathbf{P}_i \subset \mathbf{P}$; the set of egress links where the route of prefix j present is denoted by set $\mathbf{E}_j \subset \mathbf{E}$. Meanwhile, due to the constraints from certain routing policies and routing optimization objectives, such as the class-based routing [10], the geographic-constrained routing or the min-cost routing [5], for a destination j, its available egress links might have been partially sorted. For example, for a destination j on the east coast, the egress links on the east coast, say a, b and c, are superior to those on the west coast, say d and e. In this case, the available egress links of j is represented by a partially ordered set $\mathbf{E}_j^{>} = \{(a, b, c), (d, e)\}$, where the links in the same parenthesis are ranked same but superior to those in the parentheses to its right. The *typical routing policy* is another example, in which an AS prefers its customer routes over its peer routes, which are then preferred over its provider routes. The *route availability constraint* is represented by $\mathbf{A} = \{\mathbf{E}_1^{>}, \ldots, \mathbf{E}_n^{>}\}$. The route availability constraint encodes the ranking and filtering of the routes as an input of the problem and enables the solutions not only achieve the goal of robust traffic engineering but also comply with the constraints imposed by other routing policies and optimization objectives.

TABLE I
LIST OF NOTATIONS

$\mathbf{P}$	the set of destination prefixes
$\mathbf{E}$	the set of egress links
$\mathbf{D}$	the traffic demands of the prefixes
$\mathbf{C}$	the link capacities of the egress links
$\mathbf{A}$	the route availability constraint
$\mathbf{G}_j$	the configuration for prefix j
$\mathbf{G}$	the configuration for all prefixes, $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_n)$
$\mathbf{R}_i$	the set of prefixes routed through link i
$\mathbf{F}$	the set of failed links, $\mathbf{F} \in 2^{\mathbf{E}}$
$\mathbf{R}_{i/\mathbf{F}}$	the set of prefixes routed through link i given the failure of links in $\mathbf{F}$
$\mathbf{u_{i/F}}$	the link utilization of link i given the failure of links in $\mathbf{F}$, $u_{i/\mathbf{F}} = \sum_{s \in \mathbf{R}_{i/\mathbf{F}}} d_s/c_i$
$\mathbf{R}_{s,i}$	the set of prefixes switching from link i to link s after the failure of i
n	the number of prefixes in $\mathbf{P}$
m	the number of egress links in $\mathbf{E}$
k	the number of failed links, $k = \lvert\mathbf{F}\rvert$
α, β	$n = \alpha m + \beta$
α', α''	$\alpha = \alpha'(m-1) + \alpha''$
ω	the link utilization bound, $\max u_{i/\mathbf{F}} \leq \omega$
δ	the balance bound, $\max \lvert u_{i/\mathbf{F}} - u_{j/\mathbf{F}}\rvert \leq \delta$

B. Robust Interdomain Traffic Engineering Objectives

Next, we identify the objectives of robust interdomain traffic engineering.

Suppose that the set of prefixes whose traffic is routed through egress link i is $\mathbf{R}_i$. The *link utilization* of i is defined as $u_i = \sum_{s \in \mathbf{R}_i} d_s/c_i$. One of the natural and intuitive objective for interdomain traffic engineering is to minimize the maximal utilization of all egress links (called *min-max-utilization* objective), i.e. to find a minimized ω such that $u_i \leq \omega, \forall i \in \mathbf{E}$. Further, *load-balancing* is another more sophisticated objective, which is to find a minimized δ such that $\lvert u_i - u_j\rvert \leq \delta, \forall i, j \in \mathbf{E}$. δ is called *balance bound* and represent the minimal difference of the link utilization across links. Some other routing optimization objectives, such as min-cost routing within an AS [5], have already been encoded as a constraint of the problem. The ultimate goal of robust egress interdomain traffic engineering is to derive BGP configurations that ensure the min-max-utilization and the load-balancing for the minimized ω and δ in spite of network failures.

Although network failures are ubiquitous in the Internet, from the perspective of interdomain traffic engineering, we would rather focus on the failures of the egress interdomain links of the examined AS. The interdomain traffic engineering usually operates in a coarse-grained level and mainly deal with the "elephant" traffic to the popular destinations. Majority of the network failures taking place outside of the egress links have almost negligible impact on the traffic on the egress links. Besides, the impact of large scale network failures outside of the egress links on the egress interdomain traffic are usually equivalent to or less significant than those caused by the complete egress link failures. For example, in terms of the min-max-utilization objective, the impact of the egress link failures is the worst case. Accordingly, we model the network failures as the sets of failed egress links.

Formally, suppose that the egress links in a link set $\mathbf{F} \in 2^{\mathbf{E}}$ fail, where $2^{\mathbf{E}}$ denotes the power set of $\mathbf{E}$. After the failure, the set of prefixes routed through a link $i \in \mathbf{E} - \mathbf{F}$ is represented by $\mathbf{R}_{i/\mathbf{F}}$. Accordingly, the link utilization of i is denoted by $u_{i/\mathbf{F}} = \sum_{s \in \mathbf{R}_{i/\mathbf{F}}} d_s / c_i$. Obviously, if $\mathbf{F}' \subseteq \mathbf{F}$, for a link $i \in \mathbf{E} - \mathbf{F}$, $\mathbf{R}_{i/\mathbf{F}'} \subseteq \mathbf{R}_{i/\mathbf{F}}$ because the failures of the links in $\mathbf{F} - \mathbf{F}'$ cannot trigger the prefixes in $\mathbf{R}_{i/\mathbf{F}'}$ to shift to other links. The goal of robust interdomain traffic engineering is to find $\min \omega$ and $\min \delta$ such that the inequality (1) and (2) hold for $\forall i, j \in \mathbf{E}, \forall \mathbf{F} \in 2^{\mathbf{E}}$.

$$\textbf{(Min-max-utilization)} \qquad \min \omega, \text{ s.t. } u_{i/\mathbf{F}} \leq \omega \qquad (1)$$

$$\textbf{(Load-balancing)} \qquad \min \delta, \text{ s.t. } |u_{i/\mathbf{F}} - u_{j/\mathbf{F}}| \leq \delta \qquad (2)$$

In this way, the robust interdomain traffic engineering problem is model as a two-objective optimization problem. The problem is very hard. Even the problem of robust min-max-utilization or robust load-balancing alone is **NP**-Complete. First, the subproblem of robust min-max-utilization when $\mathbf{F} = \emptyset$ is **NP**-Complete, which can be proved by showing that it is not only **NP** but also reducible from the known **NP**-Complete problem Bin-Packing. In parallel, the sub-problem of robust load-balancing when $\mathbf{F} = \emptyset$ is also **NP**-Complete because it is not only **NP** but also a generalization of the known **NP**-Complete problem PARTITION [12]. Therefore, we need to simplify the problem.

Although we already shrink the network failure cases into those of egress links, there are still too many cases to be handled. Fortunately, the prevalent heavy-tail distribution in the Internet saves us from considering all possible egress link failure cases. According to the existing network failure measurement results [4], [16], [33], the possibility that multiple egress links fail simultaneously is extremely small. Actually, the one or two egress link failure cases already account for the majority. Thus, we focus on the configurations robust to the common one or two egress link failures mainly.

Further, the objectives of min-max-utilization and load-balancing is compatible to each other. Note that a balance bound δ satisfies inequality (2) also gives inequality (1) a sub-optimal $\omega_{SUB} = \max_{\mathbf{F}} \bar{u}_{\mathbf{F}} + \delta$, where $\max_{\mathbf{F}} \bar{u}_{\mathbf{F}} = \max_{\mathbf{F}} \sum_{j \in \mathbf{P}} d_j / \sum_{i \in \mathbf{E}-\mathbf{F}} c_i$. Obviously, $\max_{\mathbf{F}} \bar{u}_{\mathbf{F}}$ is a constant if $|\mathbf{F}| \leq 2$. Also, for the optimal ω_{OPT}, we have $\omega_{OPT} \geq \max_{\mathbf{F}} \bar{u}_{\mathbf{F}}$, which implies that $\omega_{SUB} - \omega_{OPT} \leq \delta$. Thus, a minimized δ for the load-balancing objective also provides a sub-optimally minimized upper bound ω_{SUB} for the maximal link utilization ω for the min-max-utilization objective. Therefore, we will mainly focus on solving the robust load balancing problem in the rest of the paper.

1	2	3	4
AB	AC	AD	AC
AC	AB	AE	AD
ϵ	AD	AB	AC

Fig. 2. $2/1$-configuration for AS A in Figure 1

C. BGP Configuration Representation

As shown in the example in Figure 1, the outcome of the robust traffic engineering problem is the ranking of the available egress links for each prefix. We use an ordered vector of egress links to represent the configuration for a prefix and use a matrix composed of the ordered egress link vectors to represent the configuration for all the prefixes.

The configuration for a prefix j is represented by an ordered vertical vector of egress links $\mathbf{G}_j = (e_{j1}, e_{j2}, \dots)^T$, and $e_{ji} \in \mathbf{E}_j$, sorted in the descending order of preferences. The first one e_{j1} is the default egress link for prefix j. Once e_{j1} is unavailable, prefix j will switch to the second choice e_{j2} if this link is available. If e_{j2} is unavailable, j will witch to the next available egress link and so forth. Apparently, if $s \neq t$, $e_{js} \neq e_{jt}$. Note that if the egress links for j are given by a partially ordered $\mathbf{E}_j^{>}$, the configuration $\mathbf{G}_j$ has to comply with the partial order. Also, if k out of m egress links fail simultaneously, the size of the configuration $\mathbf{G}_j$ must be no less than $k+1$. However, due to the route availability constraint, it is possible that the number of available egress links $|\mathbf{E}_i|$ is less than $k+1$. In order to get a configuration of more than k, we introduce an empty link ϵ to "pad" the configuration $\mathbf{G}_j$ such that $|\mathbf{G}_j| > k$ even if $|\mathbf{E}_j| \leq k$. In this case, if all the links to prefix j fail at the same time, prefix j will switch to an "empty" link ϵ, which means that prefix j becomes unreachable. Note that if $e_{jk} \neq \epsilon$, then $e_{j1}, \dots, e_{j(k-1)} \neq \epsilon$ either. That is, except that all of its available egress links fail, the reachability of prefix j is always guaranteed by the configuration. For all n prefixes in $\mathbf{P}$, the configuration for k egress link failures is a matrix of egress links links $\mathbf{G} = (\mathbf{G}_1, \dots, \mathbf{G}_n)$ of size $r \times n$, where $r \geq k+1$.

Definition 1 (k/δ-robust configuration): Given an AS with m egress links $\mathbf{E}$ and a configuration $\mathbf{G}$ for n destination prefixes $\mathbf{P}$ under the route availability constraint $\mathbf{A}$. If the inequality (2) holds for the given balanced bound δ before and after the failures of links in any $\mathbf{F}$ with $|\mathbf{F}| = k$, the configuration $\mathbf{G}$ is said to be δ-balanced for any k link failure; if the configuration $\mathbf{G}$ is δ-balanced for t link failure and $t = 0, 1, \dots, k$, the configuration is said to be k/δ-robust.

A k/δ-robust configuration should be a matrix of size $r \times n$, where $r \geq k+1$. Definition 1 also implies that a k/δ-robust configuration is also $(k-1)/\delta$-robust and so on.

Following the above definition and representations, Figure 2 gives an example representation of a $2/1$-robust configuration

1	2	3	...	i	...	$n-1$	n
1	2	3	...	i	...	$n-1$	n
3	4	5	...	$i+2$	...	1	2
2	3	4	...	$i+1$	...	n	1

Fig. 3. Circulant scheme of $2/1$-robust configuration for $n = m$

...	$2i-1$	$2i$	...	$n-2$	$n-1$	n
...	$2i-1$	$2i$	...	$n-2$	$n-1$	n
...	$2i$	$2i-1$	...	$n-1$	n	$n-2$
...	$2i+1$	$2i+2$	...	1	2	2

Fig. 4. 2-block scheme of $2/1$-robust configuration for $n = m$

for AS A in Figure 1 (with $c_i = d_j = 1$ for all i, j). For AS A, the loads on its 4 egress links are always 1-balanced for any no more than two link failures. And also, the min-max-utilization objective is also satisfied for $\omega = 2$ for any no more than two link failures.

III. Coarse-grained Robust Traffic Engineering

Interdomain traffic engineering normally concerns the interdomain traffic to the "popular" destinations [9]. In this granularity, the subtle difference of traffic demands to different destinations and in capacities of different egress links is not very important and can be omitted. At the same time, these popular prefixes usually have such rich connectivity that they are reachable through almost any egress link equally, i.e. nearly no route availability constraint presents for these prefixes. Formally, for the coarse-grained robust traffic engineering problem, for any prefix $i \in \mathbf{E}$, $j \in \mathbf{P}$, $c_i = 1$, $d_j = 1$ and $\mathbf{E}_j = \mathbf{E}$. Accordingly, the link utilization of an egress link is equal to the number of prefixes routed through it, i.e. $u_{i/\mathbf{F}} = |\mathbf{R}_{i/\mathbf{F}}|$. Therefore, the robust traffic engineering objectives can be simplified as:

$$\begin{cases} |\mathbf{R}_{i/\mathbf{F}}| \le \omega \\ \big||\mathbf{R}_{i/\mathbf{F}}| - |\mathbf{R}_{j/\mathbf{F}}|\big| \le \delta \end{cases}, \forall \mathbf{F} \in 2^{\mathbf{E}}, \forall i, j \in \mathbf{E} - \mathbf{F}. \quad (3)$$

Obviously, in this setting ω and δ are integers. We will show that if the number of prefixes and egress links are equal, there always exists a $2/1$-robust configuration for any link failures $\mathbf{F}$ with $|\mathbf{F}| \le 2$. Further, there always exists a $2/2$-robust configuration for any number of prefixes and egress links for any link failures $\mathbf{F}$ with $|\mathbf{F}| \le 2$. And these results are the optimal solutions for the robust load-balancing objective. At the same time, for the robust min-max-utilization objective, the resulting configurations also achieve the optimal link utilization upper bound ω_{OPT}.

A. Same number of prefixes and links

Our discussion of the simplest setting, in which $n = m$, begins with several schemes of $2/1$-robust configurations, then demonstrates that we cannot derive any $k/1$-robust configuration if $k > 2$ and $n = m > 5$.

1) $2/1$-robust configuration: A $2/1$-robust configuration can be implemented for any $n = m$. There are several feasible schemes.

Figure 3 shows a *circulant scheme* of a $2/1$-robust configuration. In this configuration, suppose two links i and j fail and $i < j$. If $j - i = 2$, prefix i will switch from link i to link $i + 1 \pmod n$ and prefix j will switch from link j to link $j + 2 \pmod n$. $\because j - i = 2, \therefore i + 1 = j - 1 \neq j + 2 \pmod n$ ($j > 3$). Otherwise, prefix i will switch from link i to link $i + 2 \pmod n$ and prefix j will switch from link j to link $j + 2 \pmod n$. $\because j = i, \therefore i + 2 \neq j + 2 \pmod n$. Therefore, the configuration shown in Figure 3 is $2/1$-robust.

Figure 4 shows another scheme of a $2/1$-robust configuration, called a 2-*block scheme*. In this configuration, the n links are divided into $\lfloor n/2 \rfloor$ blocks. Each block has 2 or 3 links. In a two link block, if one of the links in a block fails, the prefix of the failed link switches to the other link in the same block. If both links in a two link block fail, the two prefixes shift to another two links in one of the other blocks. If n is odd, there is a block that contains 3 links the configuration of which is shown in the last 3 columns in Figure 4. If one of the links fails, the prefix of this link switches to another link in the this block; if two links fail at the same time, one of the prefix switches to a link outside of the block and the other one switches to the third link in this block. Therefore, the configuration shown in Figure 4 is also $2/1$-robust.

Note that $2/1$-robust configuration is the best solution we can get for no more than 2 link failure cases as δ is an integer and $\delta \ge 1$ if $n = m$ and $n > 5$ for $k \le 2$.

2) $k/1$-robust configuration: Although several schemes of the $2/1$-configuration can be found for the $n = m$ case, no $k/1$-robust configuration can be constructed if $k > 2$ and $n = m > 5$.

Theorem 1: If $n = m$ and $n > 5$, a $3/1$-robust configuration cannot be implemented.

Before we prove Theorem 1, we first prove the following lemma.

Lemma 1: If $n = m$, in a $2/1$-robust configuration, $\forall i \neq j$, $e_{ik} \neq e_{jk}$, where $k = 1, 2$.

Proof: It is trivial to show that $e_{i1} \neq e_{j1}$, $\forall i \neq j$. Otherwise the configuration is even not 1-balanced when there is no link failure. Then we show $e_{i2} \neq e_{j2}, \forall i \neq j$ by contradiction. Assume that $\exists i \neq j$ such that $e_{i2} = e_{j2}$. Because $e_{i1} \neq e_{i2}$, $e_{j1} \neq e_{j2}$ and $e_{i1} \neq e_{j1}$, the failure of links e_{i1} and e_{j1} will shift 3 prefixes to link e_{i2} (e_{j2}), which is not 1-balanced. ■

The proof of Theorem 1 is as follows.

Proof: We prove the theorem by showing that we can always find 3 links whose failure leads to a 2-balanced configuration. Because $3/1$-robust implies $2/1$-robust, according to Lemma 1, $e_{ik} \neq e_{jk}, i = j, k = 1, 2$. We also know that $e_{i1} \neq e_{i2}$ and $e_{j1} \neq e_{j2}$. For prefix i, because $e_{i1} \neq e_{i2}$, assume that link e_{i1} and e_{i2} fail, prefix i will switch to link e_{i3}. Then because $e_{i2} \neq e_{j2}, \forall i \neq j$, there must exist a prefix j such that $e_{j2} = e_{i3}$. If $e_{j1} = e_{i2}$, we can find another link k, such that $e_{k1} \neq e_{i1} \neq e_{i2}(e_{j1})$, then the failure of links e_{i1}, $e_{j1}(e_{i2})$ and e_{k1} will lead to a 2-balanced configuration in which both prefixes i and j switch to the same egress link $e_{i3}(e_{j2})$. If $e_{j1} \neq e_{i2}$, the failure of links e_{i1}, e_{i2} and e_{j1} will lead to a 2-balanced configuration in which both prefixes i and j switch to the same egress link e_{i3}. ■

According to Theorem 1 and Definition 1, we have the following corollary.

Corollary 1: A $k/1$-robust configuration cannot be implemented if $k \geq 3$ and $n = m > 5$.

B. Any number of prefixes and links

In this section, we investigate the scenario where n and m are arbitrary numbers. We first derive a $2/2$-robust configuration incrementally. Then we will show that a better configuration for the cases with no more than 2 failed links, namely a $2/1$-robust configurations, cannot always be constructed. In other words, $2/2$-robust configuration is also optimal in this setting in the sense that it can always be implemented.

1) $2/2$-robust configuration:

a) $0/1$-robust configuration: Because n can be written as $n = \alpha m + \beta$, where $\alpha \geq 0$ and $0 \leq \beta < m$, if the configuration ensures 1-balanced load balancing when there is no link failure, the n prefixes should be evenly distributed among the m links such that there are β links which have $\alpha+1$ prefixes each and $m - \beta$ links each of which has α prefixes. Without loss of generality, we assume that the configuration ensures the first $m - \beta$ links have α prefixes each and these links are called αlink. The rest β links have $\alpha + 1$ links each and these links are called α^+link. Note that this scheme is the only way to construct a 1-balanced configuration for the no link failure case.

For any link i, if i is an αlink, $|\mathbf{R}_i| = \alpha$; if i is an α^+link, $|\mathbf{R}_i| = \alpha+1$. Therefore, given any two links i and j, $i \neq j$, we have $|\mathbf{R}_{i/\emptyset}| - |\mathbf{R}_{j/\emptyset}| \leq 1$, i.e. the configuration is 1-balanced.

b) $1/1$-robust configuration: We further rewrite n as $n = [\alpha'(m-1) + \alpha'']m + \beta$, where $\alpha = \alpha'(m-1) + \alpha''$, $\alpha' \geq 0$ and $0 \leq \alpha'' < m - 1$. If the configuration ensures 1-balanced load balancing when one link fails, the prefixes of the failed link need to be evenly distributed among the remaining $m-1$ links. A feasible scheme is that the first $\alpha'(m-1)$ prefixes be distributed among the other $m - 1$ links such that each link is assigned with α' prefixes. Suppose that the failed link is an αlink. If $\alpha'' \leq m - \beta - 1$, the remaining α'' prefixes can be distributed among the other $m-\beta-1$ αlinks; if $\alpha'' > m-\beta-1$, the first $m-\beta-1$ of the α'' prefixes can be placed on the other $m-\beta-1$ αlinks and the rest $\alpha''-(m-\beta-1) = \alpha''-m+\beta+1$ links be placed among the β α^+ links. In parallel, suppose that the failed link is an α^+link. If $\alpha'' + 1 \leq m - \beta$, i.e. $\alpha'' \leq m-\beta-1$, after the first $\alpha'(m-1)$ prefixes are assigned, the remaining $\alpha''+1$ prefixes should be distributed among the $m-\beta$ αlinks; if $\alpha''+1 > m-\beta$, the first $m-\beta$ prefixes should be placed on the $m-\beta$ αlinks and the rest $\alpha''+1-(m-\beta) = \alpha'' - m + \beta + 1$ prefixes are placed among the other $\beta - 1$ α^+links.

If we use term $\mathbf{R}_{s,i}$ to represent the difference of the prefixes on link s before and after the failure of link i, i.e. $\mathbf{R}_{s,i} = \mathbf{R}_{s/\{i\}} - \mathbf{R}_{s/\emptyset}$. The above $1/1$-robust prefix assignment ensures the following properties: If any link i fails, for any other two distinct link s and t, $\big||\mathbf{R}_{s/\{i\}}| - |\mathbf{R}_{t/\{i\}}|\big| \leq 1$ and $\big||\mathbf{R}_{s,i}| - |\mathbf{R}_{t,i}|\big| \leq 1$.

c) $2/2$-robust configuration: We will show that on the basis of the above $1/1$-robust configuration, a $2/2$-robust configuration can be constructed.

Since the configuration has already been $1/1$-robust, the first two rows of the configuration matrix $\mathbf{G}$ have been determined. Our task is to determine the third row of the configuration matrix $\mathbf{G}$. Suppose that two links i and j fail, we use a prefix set $\mathbf{P}_{ij}$ to represent a set of special prefixes which are originally appointed to link i (or j) before the failure and would switch to link j (or i) as the first choice after the link failure, i.e. $\mathbf{P}_{ij} = \{p | (e_{p1} = i \wedge e_{p2} = j) \vee (e_{p1} = j \wedge e_{p2} = i)\}$. These prefixes are the only prefixes whose third row of the configuration needs to be determined. Since they can be arbitrarily arranged, we do not consider these prefixes at the beginning. For the other prefixes, after i and j fail, their alternate links have been determined by the first two rows of the $1/1$-robust configuration. Since the arrangement of these prefixes has been fixed, the balance bound of the resulting prefix arrangement determines the upper bound of the final balance bound. We will show that the resulting prefix assignment is 2-balanced as follows. Without loss of generality, we assume that link i fail first then j follow. Given any other two distinct links s and t, both of which are not equal to i or j, after the failure of i, $\big||\mathbf{R}_{s/\{i\}}| - |\mathbf{R}_{t/\{i\}}|\big| \leq 1$. Then after the failure of j, we have

$$\begin{aligned}
&\big||\mathbf{R}_{s/\{i,j\}}| - |\mathbf{R}_{t/\{i,j\}}|\big| = \big||\mathbf{R}_{s/\{i\}} \cup \mathbf{R}_{s,j}| - |\mathbf{R}_{t/\{i\}} \cup \mathbf{R}_{t,j}|\big| \\
&= \big|(|\mathbf{R}_{s/\{i\}}| + |\mathbf{R}_{s,j}|) - (|\mathbf{R}_{t/\{i\}}| + |\mathbf{R}_{t,j}|)\big| \\
&\leq \big||\mathbf{R}_{s/\{i\}}| - |\mathbf{R}_{t/\{i\}}|\big| + \big||\mathbf{R}_{s,j}| - |\mathbf{R}_{t,j}|\big| \leq 2.
\end{aligned}$$

Therefore, the balance bound is 2.

Finally, we consider the prefixes in $\mathbf{P}_{ij}$. Since they can be arbitrarily arranged, we arrange these prefixes one by one in such a way that each of them is assigned to one of the links that have the least number of prefixes. Since we always increase $\min_{s \neq i,j} |\mathbf{R}_{s/\{i,j\}}|$ while $\max_{t \neq i,j} |\mathbf{R}_{t/\{i,j\}}|$ keeps constant, the balance bound $\delta = \max |\mathbf{R}_{s/\{i,j\}}| - \min |\mathbf{R}_{t/\{i,j\}}|$ will not grow. Therefore, the balance bound of the resulting configuration is guaranteed no more than 2.

2) 2/1-robust configuration: We will show that $2/1$-robust configurations cannot be derived for arbitrary n and m. It depends on the relationship between n and m.

Suppose two links fail, according to the $1/1$-robust configuration for the one link failure case, the first $\alpha'(m-1)$ prefixes have been evenly distributed. Thus, the balance bound is decided by the arrangement of the remaining α'' or $\alpha''+1$ prefixes if $\alpha'' \leq m - \beta - 1$, or the remaining $\alpha'' + 1 - m + \beta$ links if $\alpha'' > m - \beta - 1$.

As shown in Figure 5, let γ represent the maximal number of overlapped links that the remaining α'' or $\alpha''+1$ or $\alpha''+1-m+\beta$ prefixes of any two failed links switch to, we have the following theorem.

Theorem 2: If $\alpha'' \leq m-\beta-1$ and $2\alpha''-\gamma+2\alpha' < m-\beta+2$ or $\alpha'' > m-\beta-1$ and $2(\alpha''+1-m+\beta)-\gamma+2\alpha' < \beta-2$, there is no 1-balanced configuration for any two link failure.

Proof: We first consider the situation where $\alpha'' \leq m - \beta - 1$. Note that, for any link failure, the remaining α'' or

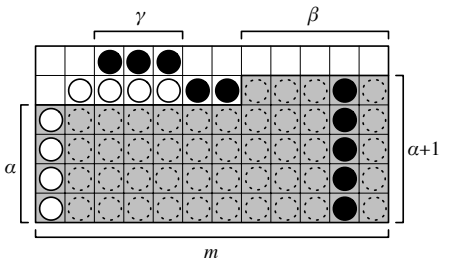

Fig. 5. Illustration of prefix overlapping after the failure of two links

$\alpha''+1$ prefixes have to be placed among the first $m-\beta$ αlinks. Assume that the two failed links are αlinks, there are $2\alpha''-\gamma$ links among the first $m-\beta$ αlinks being occupied with one or two prefix from the remaining α'' prefixes of each of the two failed links. Also, note that there are $2\alpha'$ "free" prefixes that were initially assigned to the links that are now failed. These $2\alpha'$ prefixes can be assigned to any links to balance the prefix distribution. If $2\alpha''-\gamma+2\alpha' < m-\beta-2$, there exist at least one link that is assigned $\alpha+2\alpha'$ prefixes, but those links that have the overlapped prefixes have $\alpha+2\alpha'+2$ prefixes, which lead to a balance bound $\delta \geq 2$. Similarly, we can prove the other cases where the two failed links are both α^+links, or one is an αlink and the other is an α^+link.

When $\alpha'' > m-\beta-1$, for any link failure, the remaining $\alpha''+1-m+\beta$ prefixes can be assigned to any links. We consider two cases. *Case I*: We assume that the γ links are among the β α^+links only. Suppose that the two failed links s and t are αlinks. Similar to the above proof procedure, we can show the statement holds. *Case II*: We assume that the γ links are not only among the β α^+links but also the first $m-\beta$ αlinks. In this case, the overlapped links among the αlinks can have as high as $\alpha+2\alpha'+4$ prefixes while some links among the α^+links can have as low as $\alpha+2\alpha'+1$ links before the "free" prefixes are assigned. In other words, given γ overlapped links, the situation that γ links are among the β α^+links is the best case in the sense that it needs the least number of "free" prefixes to balance the prefix distribution. Therefore, if the "free" prefixes cannot 1-balance the prefix distribution in Case I where the γ links are α^+links, it definitely cannot in Case II. All in all, if conditions $\alpha'' > m-\beta-$ and $2(\alpha''+1-m+\beta)-\gamma+2\alpha' < \beta-2$ are satisfied, there is no 1-balanced configuration. ■

Specially, we develop the following lemma to identify a sufficient condition that there exist two links whose failures will lead to at least γ overlapped links.

Lemma 2: Given a set $\mathbf{E}=\{1,\ldots,m\}$ and a subset $\mathbf{E}\subseteq\mathbf{E}$, $\forall i \in \mathbf{E}$ has a *target set* $\mathbf{E}'_i \subseteq \mathbf{E}'$ and $i \notin \mathbf{E}'_i$. Given a number $\gamma \leq \min\{|\mathbf{E}'_i|\}$, if

$$\sum_{i=1}^{m}\binom{|\mathbf{E}'_i|}{\gamma} > \binom{|\mathbf{E}'|}{\gamma} \tag{4}$$

then $\max\{|\mathbf{E}'_i \cap \mathbf{E}'_j|\} \geq \gamma$ for $\forall i,j$.

Proof: The elements in $\mathbf{E}'$ have totally $\binom{|\mathbf{E}'|}{\gamma}$ different combinations of size γ. The elements in a target set $\mathbf{E}'_i$ has $\binom{|\mathbf{E}'_i|}{\gamma}$ combinations of size γ. If two target sets share a common combination, they have at least γ overlapped elements. According to the pigeonhole principle, if the inequality (4) holds, there must exist two target sets that share a common combination. ■

1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	2	2	3	3	4	4	5	5	6	6	6
2	4	3	4	2	5	1	5	1	4	1	2	3
5	2	1	1	1	1	3	2	3	3	5	5	4

Fig. 6. $2/1$-robust configuration for 13 prefixes and 6 egress links

In the context of this paper, Lemma 2 can be interpreted in the following way. Given a link i, a set of its prefixes of size $|\mathbf{E}'_i|$ will shift to the link set $\mathbf{E}'_i$ if i fails. The target links are limited to the first $m-\beta$ αlinks or the last β α^+links, which refer to the link set $\mathbf{E}'$. Lemma 2 gives a sufficient condition that if the inequality (4) holds, for any configuration, there must exist at least two links i and j whose failures will lead the relevant prefixes to $\mathbf{E}'_i \cup \mathbf{E}'_j$ alternative links but the size of the overlapped links is $|\mathbf{E}'_i \cap \mathbf{E}'_j| \geq \gamma$.

Lemma 2 together with Theorem 2 helps us to identify a subset of the settings which have no 1-balanced configurations. Here gives an example setting that a $2/1$-robust configuration cannot be constructed. Suppose that we distribute the traffic of 12 prefixes among 6 egress links, i.e. $n=12$ and $m=6$. Since $12=[0\times(6-1)+2]\times 6+0$, $\alpha=2$, $\beta=0$, $\alpha'=0$ and $\alpha''=2$, $\mathbf{E}'$ consists of all the six links and $|\mathbf{E}'_i|=2$ for $i=1,\ldots,6$. Because $6\times\binom{2}{1}=12>6=\binom{6}{1}$, there must exist at lest two links, whose prefixes will switch to at least $\gamma=1$ overlapped alternative link after the links fail. Because $\alpha''=2<5=m-\beta-1$ and $2\alpha''-\gamma-2\alpha'=3<4=m-\beta-2$, according to Theorem 2, there is no 1-balanced configuration.

On the contrary, if the conditions in Theorem 2 is unsatisfied, we are able to construct a 1-balanced configuration. For example, we assign 13 prefixes to 6 egress links, i.e. $n=13$ and $m=6$. In this case, $\alpha=2$, $\beta=1$, $\alpha'=0$ and $\alpha''=2$. The set $\mathbf{E}'$ consists of the first 5 links. We are able to find six target sets $\mathbf{E}'_i$ for each link $i=1,\ldots,6$, such that $\gamma=\max|\mathbf{E}'_i\cap\mathbf{E}'_j|=1$. The six target sets are $\mathbf{E}'_1=\{2,4\}$, $\mathbf{E}'_2=\{3,4\}$, $\mathbf{E}'_3=\{2,5\}$, $\mathbf{E}'_4=\{1,5\}$, $\mathbf{E}_5=\{1,4\}$ and $\mathbf{E}'_6=\{1,2,3\}$. Because $\alpha''=2<4=m-\beta-1$ and $2\alpha''-\gamma+2\alpha'=3=m-\beta-2$, which does not violate the condition in Theorem 2, we can find a $2/1$-robust configuration as shown in Figure 6.

Finally, the configuration derived in this section is also optimal with regard to the robust min-max-utilization objective since $\max|\mathbf{R}_{i/\mathbf{F}}|$ is always minimized.

IV. Fine Grained Robust Traffic Engineering

In this section, we present a greedy algorithm that derives robust configurations for the general settings where traffic demands and link capacities are various and route availability constraint is deployed. Because both the problem of robust min-max-utilization and robust load-balancing in this setting are **NP**-Complete, we have to develop heuristic algorithms to solve the problem. The input of the algorithm consists of the

```
GREEDYASSIGN(P, E, D, C)
1  Sort P such that d_1 ≥ d_2 ≥ ··· ≥ d_n
2  for p ∈ P
3  do Choose i ∈ E s.t. (Σ_{s∈R_i} d_s + d_p)/c_i is minimized
4     R_i ← R_i ∪ {p}
```

Fig. 7. Pseudocode of GREEDYASSIGN algorithm

prefix set $\mathbf{P}$, the egress link set $\mathbf{E}$, the traffic demands $\mathbf{D}$, the link capacities $\mathbf{C}$ and the route availability constraint $\mathbf{A}$. The output of the algorithm is a configuration $\mathbf{G}$ such that $u_{i/\mathbf{F}} \leq \omega$ and $u_{i/\mathbf{F}} - u_{j/\mathbf{F}} \leq \delta$ hold for sub-optimal ω_{alg} and δ_{alg} for $\forall i, j \in \mathbf{E} - \mathbf{F}$, $\forall \mathbf{F} \in 2^{\mathbf{E}}$.

A. Greedy Algorithms

1) Without route availability constraint and link failure: For clarity, we first derive the configuration for no link failure case under the assumption that there is no route availability constraints $\mathbf{A}$. The algorithm runs as follows: we first sort the prefixes in descending order of traffic demands, then we assign prefixes to an egress link one by one such that in each step the link utilization of the destination link is minimized. The pseudocode is shown in Figure 7. The time complexity of this algorithm is $O\big(n(\ln n + m)\big)$. Theorem 3 shows that this algorithm provides a bounded δ_{alg}.

Theorem 3: In algorithm GREEDYASSIGN, $\delta_{alg} \leq d_{\max}/c_{\min} = d_{\max}$.

Proof: Suppose that the n prefixes in $\mathbf{P}$ are assigned by the algorithm in the order of $\{1, 2, \ldots, n\}$. Without loss of generality, we assume that the links are ordered in such a way that after the assignment the first link 1 is the link whose utilization is the maximum and the last link m is the minimum. Therefore, $\delta_{alg} = u_1 - u_m$. For a link i, assume that its prefixes are assigned in the order of $t_{i1}, t_{i2}, \ldots, t_{is_i}$. Let $\mathbf{R}_i^l$ denote the prefix set assigned to link i when the lth prefix has been assigned. We have $\sum_{s\in\mathbf{R}_i^l} d_s \leq \sum_{s\in\mathbf{R}_j^{l+\Delta}} d_s$ where $\Delta = 1, 2, \ldots$. For the first link, after we assign the last prefix n, we have

$$u_1 = \frac{\sum_{s\in\mathbf{R}_1^n} d_s}{c_1} = \frac{\sum_{s\in\mathbf{R}_1^{t_{1s_1}-1}} d_s + d_{t_{1s_1}}}{c_1}$$

According to the algorithm, prefix t_{1s_1} should be assigned in step t_{1s_1} such that the link utilization of link 1 is the minimal. Therefore,

$$\frac{\sum_{s\in\mathbf{R}_1^{t_{1s_1}-1}} d_s + d_{t_{1s_1}}}{c_1} \leq \frac{\sum_{s\in\mathbf{R}_j^{t_{1s_1}}} d_s + d_{t_{1s_1}}}{c_j}, \forall j \in \mathbf{E}_{t_{1s_1}} \tag{5}$$

Since there is no route availability constraint, i.e. $\mathbf{E}_{t_{1s_1}} = \mathbf{E}$, we have

$$\begin{aligned} u_1 &\leq \frac{\sum_{s\in\mathbf{R}_m^{t_{1s_1}}} d_s}{c_m} + \frac{d_{t_{1s_1}}}{c_m} \\ &\leq \frac{\sum_{s\in\mathbf{R}_m^n} d_s}{c_m} + \frac{d_{t_{1s_1}}}{c_m} = u_m + \frac{d_{t_{1s_1}}}{c_m} \end{aligned}$$

Therefore,

$$\delta_{alg} = u_1 - u_m \leq \frac{d_{t_{1s_1}}}{c_m} \leq \frac{d_{\max}}{c_{\min}} \tag{6}$$

```
GREEDYROBUSTBALANCE(P, E, D, C, A)
1  G ← {}
2  for F ∈ 2^E
3  do Initialize R_i, i = 1, ..., n w.r.t. G and F
4     Derive P_F w.r.t. F and F
5     Sort P_F such that (−|E_1|, d_1) ≥ (−|E_2|, d_2) ≥ ...
6     for p ∈ P_F
7     do Choose i ∈ E_p − G_p s.t. (Σ_{s∈R_i} d_s + d_p)/c_i is minimized
8        R_i ← R_i ∪ {p}
9        Update G_p
10 return G
```

Fig. 8. Pseudocode of the GREEDYROBUSTBALANCE algorithm

■

Note that in (6) δ_{alg} is upper bounded by $d_{t_{1s_1}}/c_m$. The algorithm is greedy in the sense that we first sort the prefixes in the descending order of traffic demands, which makes $d_{t_{1s_1}}$ as small as possible such that the upper bound for δ_{alg} as small as possible.

Similarly, for the min-max-utilization objective, we can prove that Algorithm GREEDYASSIGN also gives a bounded ω, as shown in Theorem 4.

Theorem 4: In algorithm GREEDYASSIGN, $\omega_{alg} \leq \bar{u}\big[1 + \frac{(m-1)d_{\max}}{\sum_j d_j}\big]$.

B. Incorporating route availability constraints without partial orders

If we further incorporate the route availability constraints without partial order into the greedy algorithm, the inequity (6) cannot be directly derived from the inequity (5) because $\mathbf{E}_{t_{1s_1}} \subseteq \mathbf{E}$ and link m is not necessarily in $\mathbf{E}_{t_{1s_1}}$. Therefore, we have to modify the algorithm GREEDYASSIGN as follows: at the beginning of the algorithm, we sort the prefixes according to the ascending order of tuple $(-|\mathbf{E}_i|, d_i)$, i.e. we first sort the prefixes in ascending order of the size of its available egress links, then in the descending order of the traffic demands. The algorithm is greedy in the sense that we first try to make $\mathbf{E}_{t_{1s_1}}$ close to $\mathbf{E}$ as much as possible such that the possibility that link m is in $\mathbf{E}_{t_{1s_1}}$ becomes as large as possible. If $m \in \mathbf{E}_{t_{1s_1}}$, we can get the upper bound shown in (6) with as small as possible $d_{t_{1s_1}}$; otherwise, since $\mathbf{E}_{t_{1s_1}}$ is close to $\mathbf{E}$ as much as possible, the difference between $\min_{j\in\mathbf{E}} u_j$ and u_m becomes as small as possible such that we can get an as good as possible approximation for the upper bound of $u_1 - u_m$.

The complete algorithm GREEDYROBUSTBLANCE that exploits above greedy procedure is shown in Figure 8. At first, the configuration $\mathbf{G}$ for all the prefixes is unknown. We start with the initial state that the failure link set $\mathbf{F} = \emptyset$. For each failure link set $\mathbf{F}$ and the configuration $\mathbf{G}$, we can get a set of "free" prefixes $\mathbf{P_F}$ whose destination cannot be determined by the current configuration $\mathbf{G}$ with regard to the failure of the links in $\mathbf{F}$. At the beginning, the configuration is not initialized, so $\mathbf{P_F} = \mathbf{P}$. By utilizing the aforementioned greedy procedure for the "free" prefixes in $\mathbf{P_F}$, we get the load balancing configuration for them. Then, we assume one of the link fails, and re-run the greedy procedure to get the configuration for the new free prefixes given the existing configuration and so on until we get all the configurations for

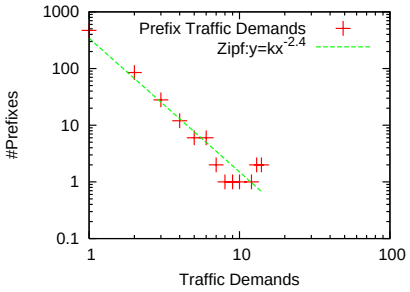

Fig. 9. Distribution of traffic demands follows the Zipf distribution

all the combination of failure links that we need. Accordingly, the general time complexity of the algorithm is $O(m^k n(\ln n + m))O(m^{k+1}n)$.

1) Incorporating route availability constraints: The GREEDYROBUSTBALANCE algorithm can be easily extended to the settings where partially ordered route availability constraint presents. In this case, when the algorithm is searching for the candidate egress link for a prefix j, only the currently highest ranked available egress links of j are considered. In this way, the algorithm assigns the prefixes among their egress links in the same rank to achieve the best balance bound, and at the same time, the partial order of the egress links imposed by the other routing objectives are preserved.

C. Evaluation of Greedy Algorithm

We examine the efficiency of the algorithm with simulations. We first describe the methodology to get the required input data for the problem, then evaluate the performance of the algorithm.

1) Data Acquisition: We derive the information of the popular prefixes, including their traffic demands and route availability constraints, from real Internet data. We first collected the IP addresses of the world top 1000 most visited web sites from a public ranking site [2] on September 5, 2005. Then we mapped these 1000 IP addresses into around 600 prefixes based on the BGP routing table of ROUTEVIEWS server [1] on the same date. We assume that the traffic demands for each web site is equal to 1 and the traffic demands for a prefix is the aggregated traffic demands of the web sites in the prefix. The distribution of the traffic demands of these prefixes is shown in Figure 9. It shows that the distribution roughly follows a Zipf distribution with $\lambda \approx -2.4$. The similar traffic demands distributions are also found in other Internet traffic measurement work [7]. Thus we believe that the traffic demands distribution for the evaluation matches the reality.

We choose six ASs for our evaluation. Three of them are Internet backbone transit ASs in the North America. The other three are stub ASs. One of them (STUB1) is a popular content provider and the other two are IT companies. The simulation needs the information of the BGP configuration of these ASs for the prefixes and their egress links. However, the information is unavailable from public data sources. Instead, we collect the inferred BGP routing information of these ASs for the examined prefixes from a web site providing AS path inference services to the public [18]. For each pair of destination prefix and source AS, the site provides a set of possible AS paths in the descending order of likelihood that the path is actually used by the source AS to reach the destination prefix. We assume that the order of the paths is the default configuration of the egress link preference for each AS and destination prefix pair. We will use this default configuration as the baseline for comparison in the later part.

TABLE II
NUMBER OF EGRESS LINKS AND DESTINATION PREFIXES FOR THE EXAMINED ASS

AS	TRAN1	TRAN2	TRAN3	STUB1	STUB2	STUB3
#egress	8	15	13	7	4	4
#prefix	567	574	548	609	618	613

Based on the collected BGP routing information, we derive a list of egress links for each AS, the route availability constraint for these prefixes and ASs, and the default configurations of each AS. The statistics on the number of prefixes and the egress links for these ASs are listed in Table II. We keep the AS path information through the peer links for the transit ASs and that through the provider links for the stub ASs. As a result, only the prefixes reachable through the peer links for the transit ASs and those reachable through provider links for the stub ASs are considered, which result in the inconsistency in the number of prefixes for the six ASs. In order to incorporate partial orders of the egress links, we assume that due to geographical constraints, for the top 10% prefixes that own the largest traffic demands, for each AS, its first two egress links in the default setting are superior to the rest of the available egress links. Note that, due to the limitation of the data source, the egress links are AS level links, i.e. there is at most one link between two ASs. However, the analysis and algorithm in this paper neither assume nor restrict to the AS level links.

The link capacity of the egress links are inferred in the following way. First of all, due to technical constraints [14], we assume that the choice of the link capacities are one of the following typical values: 155M (OC-3), 622M (OC-12), 1G (GE), 2.5G (OC-48) and 10G (10GE). For each egress link for an AS, according to the default configuration for all the prefixes, we can identify a set of prefixes that use this link as the default egress link in the case that there is no link failure. Based on the assumption that the default configuration should ensure load-balancing in certain degree, we assign an appropriate capacity value for this link such that it is roughly proportional to the total traffic demands of this set of prefixes.

After we inferred the values of d_i and c_j, we will normalize the values such that $\min \mathbf{D} = 1$ and $\min \mathbf{C} = 1$. The aforementioned procedures provide us a realistic input consisting of $\mathbf{P}$, $\mathbf{E}$, $\mathbf{D}$, $\mathbf{C}$ and $\mathbf{A}$.

2) Evaluation: Given above information, we run the algorithms for each ASs to derive the configurations and examine the value of corresponding balance bound. In the experiment, we examine the performance of algorithm for the scenarios with no more than 2 failed egress links. Given a combination

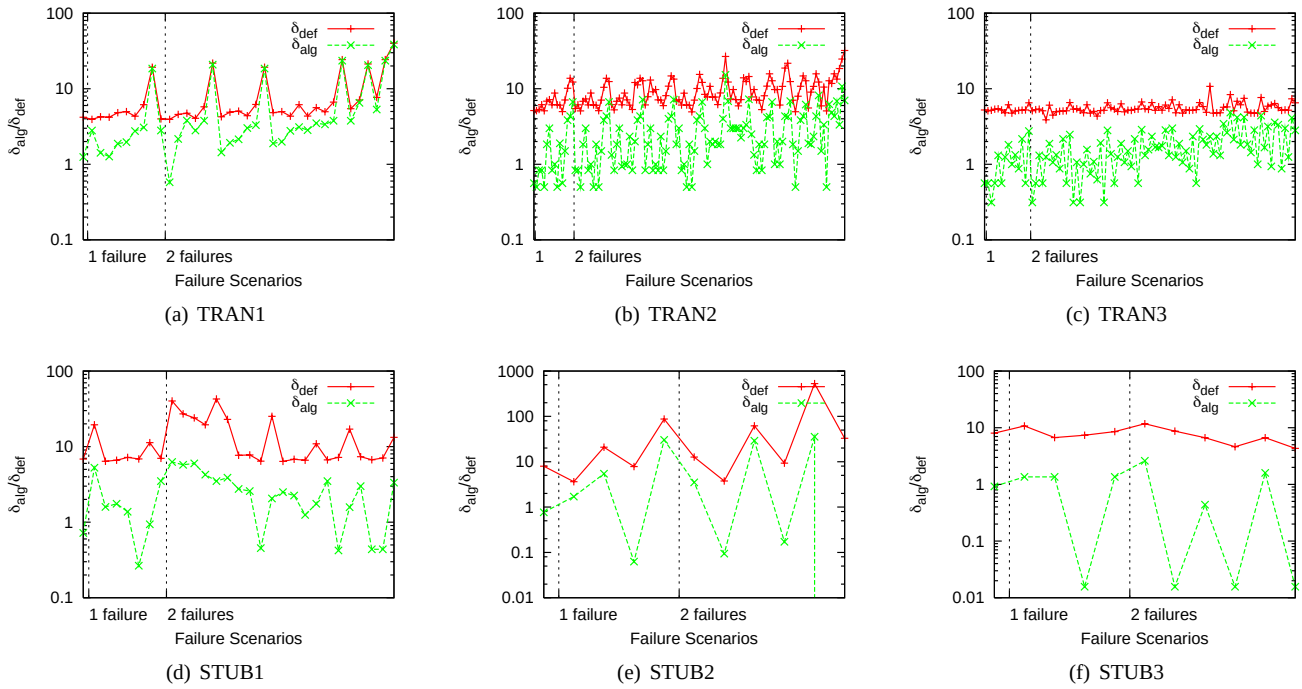

Fig. 10. Performance δ for the examined ASs

TABLE III
AVERAGE PERFORMANCE (δ) FOR EXAMINED ASS

ASs	TRAN1	TRAN2	TRAN3	STUB1	STUB2	STUB3
δ_{def}	8.7	9.6	5.6	13	71	7.6
δ_{alg}	6.7	2.8	1.7	2.5	9.7	0.88
Improve(%)	23	71	70	81	86	89

of the failure links **F**, the values of δ in different scenarios for the six ASs are shown in Figure 10. The x axis represents the index of the failure scenarios, which is sorted in ascending order of the number of failed links. The y axis indicates the value of balance bound δ. For comparison, we take into account two types of balance bounds: δ_{alg}, which shows the balance bounds of the algorithm-derived configurations, and δ_{def}, which shows the balance bounds that are calculated according to the default configurations derived from the original AS path inference results. The curves show that the configurations derived by the algorithm always outperform the default configurations even though we intentionally set the link capacities at the beginning such that the loads are roughly balanced if no failure presents and the default configuration is deployed. For example, as shown in Figure 10(e), δ_{def} can be as worse as 500, while the configuration derived by our algorithm always keeps the balance bound δ_{alg} in the order of $1 \sim 10$.

Further, Table III lists the average value of δ_{alg} and δ_{def} for each examined ASs. It shows that except TRAN1, the algorithm can reduce the balanced bound by more than 70% compared with the default settings.

V. RELATED WORK

Although it is said that the state-of-the-art practice of interdomain traffic engineering is still staying in the "trial-and-error" era, some fundamental efforts have been made. Quoitin *et al* summarize the current best practices in [21]. Feamster *et al* outlined the guidelines for interdomain traffic engineering [9]. A number of measurement works characterized the interdomain traffic patterns [6], [8], [23], [27]. The interaction between the inter- and intra- domain routing and traffic are also extensively studied in [3], [25]. Furthermore, Wang *et al* identified the possible instability caused by the route coordination in traffic engineering and suggested practical guidelines to perform safe interdomain traffic engineering in a hierarchical routing system [31], [32]. The potential instability caused by AS path prepending for the purpose of inbound traffic engineering was also studied by Wang *et al* [30].

Several new mechanisms have been proposed to address the limitations of BGP for interdomain traffic engineering [15], [19], [20], [24]. In particular, the optimization methodologies are widely employed to cope with the complexity in achieving the interdomain traffic engineering goals [5], [11], [26], [28]. Our work differs from theirs in that we try to derive not only an optimized configuration but also an optimized configuration that is robust to link failures. At the same time, an online intelligent systems "tweak-it" [29] was developed. The system keeps track of the network status inside the AS and reconfigure BGP whenever a routing dynamics is detected such that the BGP configurations are adapted to the network status in real-time to guarantee the traffic engineering goals. Our solution is different from theirs in that we derive the BGP configurations with considering the potential network failures such that the routing objectives are achieved without the reconfiguration of BGP. Our work is also orthogonal to that of Teixeira *et al* [24], which proposed a mechanism that provides

flexibility in BGP route egress-point selection.

VI. Conclusions

BGP routing policies are usually configured without considering routing dynamics while routing changes are ubiquitous in the Internet, which might result in undesirable routing objectives. Therefore, we need to configure BGP in such a way that it robustly preserve the routing objectives in spite of network dynamics. In this paper, we propose the problem of robust interdomain traffic engineering. We systematically investigate the methodologies to derive robust BGP configurations which are able to guarantee the min-max-utilization and load-balancing objectives on the AS egress links even if a small number of egress links fail. For the coarse-grained robust load balancing problem in which the traffic demands of the prefixes and the capacities of the egress links are approximately equal, we derive the optimal configurations. For the fine-grained robust load balancing in which the traffic demands and the capacities are various, we utilize a greedy algorithm to derive the robust configuration. The experiment results show that the algorithm can achieve the robust load balancing in the real settings for both transit and stub ASs. Meanwhile, the robust load balancing can be achieved while the other route objectives are preserved.

Acknowledgment

The authors are supported by the NSF grant CNS-0325868, ANI-0208116 and ANI-0085848, and Alfred P. Sloan Fellowship. The authors would like to thank Vijay Ramachandran for the helpful discussion, and are also grateful to the anonymous reviewers for their valuable comments.

References

[1] Route Views Project Page. http://www.routeviews.org.

[2] Top websites - Most visited websites - Website Ranking. http://www.ranking.com/.

[3] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot. The Impact of BGP Dynamics on Intra-Domain Traffic. In *Proceedings of ACM SIGMETRICS*, 2004.

[4] O. Bonaventure, C. Filsfils, and P. Francois. Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures. In *Proceedings of the 2005 ACM CoNEXT*, pages 31–42, Toulouse, France, 2005.

[5] T. Bressoud and R. Rastogi. Optimal Configuration for BGP Route Selection. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, USA, April 2003.

[6] H. Chang, S. Jamin, Z. Mao, and W. Willinger. An Empirical Approach to Modeling Inter-AS Traffic Matrices. In *Proceedings of ACM Internet Measurement Conference*, Berkeley, CA, USA, November 2005.

[7] M. E. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, 5(6), 1997.

[8] W. Fang and L. Peterson. Inter-AS Traffic Patterns and Their Implications. In *Proceedings of IEEE Global Internet Symposium*, December 1999.

[9] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for Interdomain Traffic Engineering. *ACM SIGCOMM Computer Communications Review*, October 2003.

[10] L. Gao and J. Rexford. A Stable Internet Routing without Global Coordination. *IEEE/ACM Transaction on Networking*, 9(6):681–691, December 2001.

[11] R. Gao, C. Dovrolis, and E. Zegura. Interdomain Ingress Traffic Engineering through Optimized AS-Path Prepending. In *Proceedings of IFIP Networking*, 2005.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[13] T. G. Griffin, A. D. Jaggard, and V. Ramachandran. Design Principles of Policy Languages for Path Vector Protocols. In *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[14] L. Li, D. Alderson, W. Willinger, and J. Doylex. A First-principles Approach to Understanding the Internet's Router-level Topology. In *Proceedings of ACM SIGCOMM*, pages 3–14, Portland, Oregon, USA, 2004.

[15] R. Mahajan, D. Wetherall, and T. Anderson. Negotiation Based Routing Between Neighboring Domains. In *Networked Systems Design and Implementation (NSDI)*, May 2005.

[16] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot. Characterization of Failures in an IP Backbone. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.

[17] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot. IGP Link Weight Assignment for Transient Link Failures. In *Proceedings of 18th International Teletraffic Conference (ITC)*, 2003.

[18] J. Qiu and L. Gao. AS Path Inference by Exploiting Known AS Paths. In *Proceedings of IEEE GLOBECOM*, San Francisco, CA, USA, November 2006.

[19] B. Quoitin and O. Bonaventure. A Cooperative Approach to Interdomain Traffic Engineering. In *1st Conference on Next Generation Internet Networks Traffic Engineering*, Rome, Italy, 2005.

[20] B. Quoitin, S. Uhlig, and O. Bonaventure. Using redistribution communities for Interdomain Traffic Engineering. In *QoFIS'02*, October 2002.

[21] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain Traffic Engineering with BGP. *IEEE Communications Magazine*, 41(5):122–128, May 2003.

[22] Y. Rekhter, T. Li, and S. H. Ed. A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, January 2006.

[23] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP Routing Stability of Popular Destinations. In *Proceedings of Internet Measurement Workshop*, November 2002.

[24] R. Teixeira, T. Griffin, M. G. C. Resende, and J. Rexford. TIE Breaking: Tunable Interdomain Egress Selection. In *Proceedings of CoNEXT*, October 2005.

[25] R. Teixeira, T. Griffin, A. Shaikh, and G. Voelker. Network Sensitivity to Hot-Potato Disruptions. In *Proceedings of ACM SIGCOMM*, August 2004.

[26] S. Uhlig. A Multiple-objectives Evolutionary Perspective to Interdomain Traffic Engineering. *International Journal of Computational Intelligence and Applications (IJCIA)*, 5(2):215–230, June 2005.

[27] S. Uhlig and O. Bonaventure. Implications of Interdomain Traffic Characteristics on Traffic Engineering. *European Transactions on Telecommunications, special issue on Traffic Engineering*, January 2002.

[28] S. Uhlig and O. Bonaventure. Designing BGP-based outbound traffic engineering techniques for stub ASes. *ACM SIGCOMM Computer Communication Review*, 34(5):89–106, October 2004.

[29] S. Uhlig and B. Quoitin. Tweak-it: BGP-based Interdomain Traffic Engineering for Transit ASes. In *1st Conference on Next Generation Internet Networks Traffic Engineering*, Rome, Italy, 2005.

[30] H. Wang, R. K. Chang, D.-M. Chiu, and J. C. Lui. Characterizing the Performance and Stability Issues of the AS Path Prepending Method: Taxonomy, Measurement Study and Analysis. In *ACM SIGCOMM Asia Workshop*, Beijing, China, April 2005.

[31] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz. On the Stability of Rational, Heterogeneous Interdomain Route Selection. In *Proceedings of 13th International Conference on Network Protocols*, Boston, MA, USA, October 2005.

[32] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz. Stable Egress Route Selection for Interdomain Traffic Engineering: Model and Analysis. In *Proceedings of 13th International Conference on Network Protocols*, Boston, MA, USA, October 2005.

[33] L. Xiao and K. Nahrstedt. Reliability Models and Evaluation of Internal BGP Networks. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.