A Closed-Loop Control Traffic Engineering System for the Dynamic Load Balancing of Inter-AS Traffic

Mina Amin, Kin Hon Ho, George Pavlou & Michael Howarth
Centre for Communication Systems Research, Dept. of Electronic Engineering, University of Surrey, Guildford, Surrey, GU2 7XH, UK

Abstract: Inter-AS outbound traffic engineering (TE) is a set of techniques for controlling inter-AS traffic exiting an autonomous system (AS) by assigning the traffic to the best egress points (i.e. routers or links) from which the traffic is forwarded to adjacent ASes towards the destinations. In practice, changing network conditions such as inter-AS traffic demand variation, link failures and inter-AS routing changes occur dynamically. These changes can make fixed outbound TE solutions inadequate and may subsequently cause inter-AS links to become congested. In order to overcome this problem, we propose the deployment of a closed loop control traffic engineering system that makes outbound traffic robust to inter-AS link failures and adaptive to changing network conditions. The objective is to keep the inter-AS link utilization balanced under unexpected events while reducing service disruption and reconfiguration overheads. Our evaluation results show that the proposed system can successfully achieve better load balancing with less service disruption and re-configuration overhead in comparison to alternative approaches.

Keywords: traffic engineering, load balancing, inter-domain traffic, closed-loop control system

1. INTRODUCTION

The current Internet consists of a large collection of autonomous systems (ASes) or domains, an AS being a network or group of networks managed by a single administrative authority. Neighboring ASes exchange route information using the de facto inter-AS routing protocol, the border gateway protocol (BGP) [1], with each route consisting of an AS path vector and other attributes. Since Internet end users are associated with different ASes, most of the Internet traffic is forwarded from source to destination through a sequence of ASes. Multiple connections between ASes, also called multi-homing, are now a fundamental part of the Internet architecture, enabling ASes to achieve load balancing and resilience with respect to their inter-AS links. As a result, many internet service provides (ISPs) choose inter-AS routes by adjusting BGP route attributes in order to optimize their operational IP network performance, for example to satisfy the capacity constraints of links between neighboring ASes and to load balance inter-AS traffic [2]. This set of techniques is known as BGP traffic engineering (TE).

Border gateway protocol *outbound* TE, which has become increasingly important and has been well studied in the literature [3–5] is a set of techniques for controlling inter-AS traffic exiting an AS by assigning the traffic to the best egress points from which the traffic is forwarded to adjacent ASes towards the destinations. It should be noted that the terms egress point and inter-AS link are used interchangeably in this paper. It is commonly believed that inter-AS links are the bottleneck in the Internet. This is primarily due to two reasons: (1) the rapid growth of Internet traffic, in particular of peer-to-peer [6] and video streaming (e.g. youTube) traffic, that consumes the major part of inter-AS link bandwidth; (2) the fact that the capacity of inter-AS links is generally small compared to that of backbone intra-AS links that are often well over-provisioned. Moreover, an inter-AS link is relatively more difficult to upgrade than an intra-AS link due to time-consuming and complicated negotiations between the adjacent domains involved. As a consequence, network operators employ outbound TE techniques to control the routing of their egress traffic and use optimally the bandwidth of inter-AS links.

In practice, network conditions change dynamically and this can make the deployed outbound TE solutions ‘‘obsolete’’ and subsequently cause inter-AS links to become congested over time. One such dynamic change is inter-AS traffic variation, which is typically caused by changes in user or application behaviour or by routing changes from other ASes (i.e. change of prefix-to-egress point mapping) [7]. In addition to these traffic variations, transient or long-lasting inter-AS peering link failures may also occur. According to [8], transient inter-AS link failures last for less than a few minutes and are fairly common. For instance, out of approximately 10,000 eBGP peering link failures in a transit ISP over a period of 3 months, 82% of them lasted for no more than 3 min [8]. Upon the failure of a peering link, a large amount of traffic will be shifted to other available egress points (EP), leading potentially to congestion on these new serving EPs. In theory, although it is possible to perform

outbound TE based on various proposals in the literature [3–5], re-computing the outbound configuration in the case of unexpected changes may induce large computational overheads and involve a large number of EP re-configurations. As a result there can be excessive service disruption that is detrimental to the perceived quality for real-time services. In summary, most existing TE solutions are engineered for long-term off-line network configurations and are not appropriate for dynamic changes and rapid reconfigurations. As such, the focus of this paper is to make outbound TE more adaptive to changing IP network conditions by considering operation and management constraints such as time-efficiency, minimal service disruption and reconfiguration overhead.

In this work we propose an inter-AS outbound TE (**IOTE**) system that can be adopted by network operators to optimize their inter-domain link bandwidth utilization under changing network conditions. More specifically, the system consists of two re-optimization components: (1) primary egress point (PEP) reoptimizer, which is designed for handling dynamic traffic variations and routing changes; this determines the best primary EP selection under the normal state (NS), i.e. no inter-AS link failure; (2) backup egress point (BEP) re-optimizer, which is designed for managing inter-AS link failures so as to achieve robustness in terms of load balancing and fast rerouting recovery in case of a failure; this determines the best backup EP selection under a failure state (FS), i.e. with a single inter-AS link failure. A time-efficient heuristic algorithm is proposed for each of these two reoptimizers. The overall objective of the **IOTE SYSTEM** is to *balance the load among inter-AS links under both the NS and FSs, while reducing reconfiguration overheads and service disruptions*.

To the best of our knowledge, there is no integrated closed-loop control traffic engineering approach that addresses both primary and backup outbound TE in case of failures while taking dynamic network condition changes such as inter-AS traffic variation and routing changes into account. Existing proposals consider only failure-free conditions and do not take network changes into consideration [3–5]. The authors in [9] propose a multi-objective outbound inter-AS TE re-optimization that handles changes in the expected traffic demand and/or routing failures with a minimal burden on BGP. However, they do not optimize the performance under transient inter-AS link failures. On the other hand, the authors in [10, 11] propose an intra-AS TE solution that is robust to transient intra-AS link failures and argue that relying on a reactive robust solution may not be appropriate or even feasible, since computing and deploying a new robust solution in a fairly fast time scale can be challenging. Consequently, they propose a proactive robust solution to achieve their intra-AS TE objective. In a similar fashion, changing the EP configuration dynamically to avoid a transient failure is not a practical solution since this needs to be done under hard real-time constraints so that relevant traffic exits the AS from another egress point until the transient failure is restored. As such, in order to avoid re-configuration and achieve fast recovery from a transient or non-transient inter-AS link failure, we pursue a proactive robust TE approach to manage inter-AS link failures through the ‘‘informed’’ pre-computation of back-up egress points.

We compare the performance of the **IOTE SYSTEM** with two alternative strategies. The first strategy does not consider PEP and BEP re-optimization at all, while the second one only considers PEP re-optimization. In our evaluation model, we generate a series of random events to be handled by the proposed system and the alternative strategies we evaluate, attempting to emulate realistic changes in network conditions. Relevant events include traffic variations, routing changes and transient & non-transient inter-AS link failures. Our results demonstrate that the **IOTE SYSTEM** has the following key advantages over the other two alternative approaches: (a) in spite of the occurring changes, it maintains better load balancing on inter-AS links under both NS and FSs, which results in increasing the ability of the network to accommodate more traffic demand without the need for capacity upgrading; (b) it limits the service disruptions and reconfiguration overheads, achieving better network stability.

This paper extends our previous work in [12] by enhancing the **IOTE SYSTEM**, presenting an implementation solution based on the BGP route selection process, considering the relevant network monitoring requirements in detail and presenting a more comprehensive performance analysis and evaluation. The paper is organized as follows. In Section 2, we present the proposed approach and **IOTE SYSTEM** in detail. Section 3 presents the optimization problem handled by the PEP and BEP reoptimizer, including details of the proposed heuristic algorithms and implementation solutions. The operational procedure of the **IOTE SYSTEM** is presented in Section 4. Section 5 presents two alternative strategies for performance comparison. We then present our evaluation methodology and results in Sections 6 and 7, respectively. We finally conclude the paper in Section 8.

2. INTER-AS OUTBOUND TRAFFIC ENGINEERING SYSTEM

The architecture of the **IOTE SYSTEM** is shown in Fig. 1. It requires network monitoring and traffic measurement at egress nodes in order to identify particular conditions. When the latter are met (i.e. the current inter-AS link utilization exceeds a congestion threshold or the assignment of BGP prefixes to egress routers have changed substantially), the PEP and BEP re-optimizers are triggered to possibly produce a better network configuration. The optimized PEP and BEP solutions are then implemented in the network if pre-defined performance targets are satisfied (e.g. if the resulting inter-domain link utilization is lower than the current one by a particular margin). The system consists of three functional blocks: monitoring, optimization and implementation.

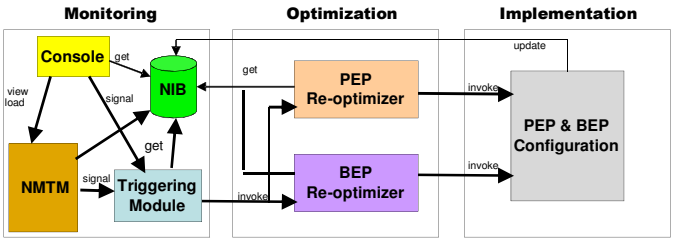

Figure 1. IOTE System

Figure 1 shows the overall architecture of the **IOTE SYSTEM**. The network monitoring and traffic measurement (NMTM) block monitors the load of inter-AS links, including the load per significant destination prefix, and the BGP configuration of the edge routers and stores relevant data in the network information base (NIB). When a pre-defined threshold is crossed, the triggering module is signaled to invoke the PEP and BEP re-optimizers. The latter use current NIB data and they may invoke PEP and BEP re-configuration if the new computed configuration is significantly better than the current one. The console allows the human network manager to view the current load, examine historical data e.g. view utilization histograms, BGP changes, etc., and also to trigger directly the re-optimization process. We explain all these components in detail below.

2.1. MONITORING BLOCK

The key function of the Network Monitoring and Traffic Measurement (NMTM) block is to obtain real-time views of traffic conditions as required by the PEP and BEP re-optimizers. These include inter-AS link load, BGP routing data and inter-AS outbound traffic per destination prefix. Given that the PEP and BEP re-optimizers require this data in real-time, a key issue behind the design of the NMTM block is to generate these real-time traffic views with relative accuracy and a relatively small impact on the managed network. We discuss relevant issues in detail as they are relevant for the deployability of our approach in operational ISP networks.

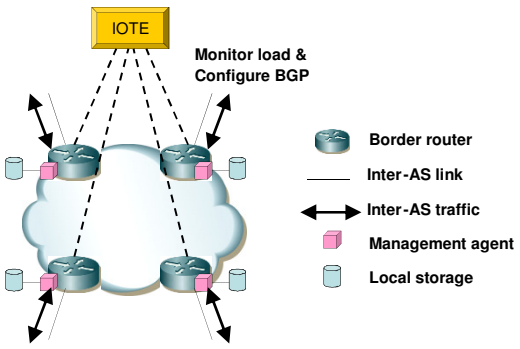

Figure 2. Network monitoring and traffic measurement infrastructure

Figure 2 illustrates the network monitoring and traffic measurement infrastructure. The following data need to be monitored at each border router:

1. *Inter-AS link load.* The outbound load of inter-AS links attached to border routers. This can be obtained by retrieving periodically the relevant byte count from the local management agent through the Simple Network Management Protocol (SNMP) and

calculating the load. Another possibility is to obtain this through vendor-specific traffic measurement tools, such as Cisco's Netflow. In the latter case, the load calculation is performed locally.

2. *BGP routing data.* The set of destination prefixes a border router is aware of, as well as the associated local preference values. With the availability of prefix reachability and local preferences, PEP and BEP solutions can be identified (as explained in Sect. 3.3). Relevant data can be obtained from the SNMP BGP MIB [13] that mirrors the BGP routing table. The latter stores the BGP routes, as advertised from the adjacent ASes, and the relevant path attributes [14].

3. *Inter-AS outbound traffic load per destination prefix.* The border router needs to measure and make available inter-AS traffic load per destination prefix, which can only be obtained using vendor-specific traffic measurement tools such as Cisco's Netflow.

Relevant retrieved data are stored in the Network Information Base (NIB) held in the central management node that performs monitoring, hosts the logic of the **IOTE SYSTEM** and implements the relevant decisions by reconfiguring BGP. This reconfiguration affects all the border routers and guides them to direct traffic for a destination prefix to a chosen primary egress router in the NS and to a chosen backup egress router in the FS. Note that because of multi-homing, most destination prefixes are typically reachable through multiple border routers.

We discuss now the issues in obtaining the required information. The inter-AS link load can be obtained by retrieving periodically the *ifOutOctets* SNMP objects from the *interfaces* table of the border routers. This information needs to be retrieved only for those interfaces connecting to the inter-AS links. The default polling period could be relatively long, e.g. 1 min or more, but it can become shorter, e.g. 10 s, when the calculated load is close to the triggering threshold. Given that the number of inter-AS links is relatively small for tier-3 and tier-2 ASes, the relevant polling overhead is relatively inexpensive. For tier-1 ISPs the number of inter-AS links can be 2,000–3,000, which means that significant polling load is required. On the other hand, tier-1 ISP networks are substantially over-provisioned, so there should be spare capacity for the required polling load. An alternative is to obtain this information through vendor-specific traffic measurement tools, given that the same will be needed in any case for the inter-AS load per destination prefix. In this case, the measured load could be sent to the network management system periodically, e.g. per 15 min, but an alarm with guaranteed delivery should be sent when the relevant threshold is crossed. Such an event-based approach requires much less bandwidth to perform the monitoring tasks.

The BGP routing data needs to be periodically retrieved from the border routers, so the management system can see from where particular destination prefixes are reachable and configure the local routing policy accordingly in order to optimize inter-AS link utilization. The BGP routing information base (RIB) can be very large and retrieving the corresponding SNMP BGP MIB table through subsequent *GetBulk* operations can be both expensive in terms of bandwidth and also time consuming in terms of the overall latency. An alternative is to retrieve this information through more efficient vendor-specific management tools or to use FTP for a vendor-specific ''flat file'' representation of the BGP RIB. This information needs to be retrieved periodically, e.g. every 4–6 hours, given that minute changes of BGP information are not of interest. The management system needs this information in order to know from where particular prefixes are reachable and to trigger BEP re-optimization when significant changes have taken place. When a new configuration is derived, this can be implemented by setting the local preference attribute of a destination prefix in order to force it to use a particular egress router. While the BGP MIB contains the local preference attribute per route entry, it does not allow a management system to set it as the relevant SNMP permission is *read-only* [13], so a proprietary mechanism such as Cisco's CLI or Juniper's JUNOScript, to name two existing popular methods, needs to be used.

Finally, the inter-AS outbound traffic per destination prefix can only be obtained through vendor-specific traffic measurement tools given that such information is not available through SNMP MIBs. Note that there are currently a few hundred thousand prefixes in the Internet and collecting real-time load data for them is challenging. As suggested in [2] though, a fairly small number of prefixes are responsible for a very large volume of the overall traffic, e.g. Google, YouTube, CNN, etc., so traffic data needs to be measured only for those popular prefixes, which reduces significantly the monitoring complexity and makes relevant real-time data generation more efficient. A practical approach on how to measure inter-AS outbound traffic is proposed in [7]. The collected data needs to be accessible by

the management system that should retrieve them periodically, e.g. through FTP, in order to build a picture of the average volume of traffic load per important destination prefix.

Given that this data is available in the NIB, the triggering module determines whether the PEP and BEP re-optimizers should be invoked according to the maximum inter-AS link utilization (MLU). Another trigger for re-optimization relates to significant changes in the advertised prefixes from adjacent ASes. The triggering module takes as input the current inter-AS outbound traffic per prefix and BGP reachability data and calculates the MLU under the normal and every possible failure state for each EP. The worst-case (highest) MLU among all the EP failure states is identified. The triggering module invokes the re-optimizers if particular network conditions are met. A triggering policy can be categorised into the following two types:

a) *Event-driven:* the re-optimization is invoked if an event occurs. In this paper, we use this event-driven policy for triggering the PEP and BEP re-optimizers as follows: (1) The PEP re-optimizer is invoked if the MLU under NS exceeds a tolerance threshold value $\alpha 1$; (2) The BEP re-optimizer is invoked if the worst-case MLU exceeds a tolerance threshold value $\alpha 2$. We believe that triggering a re-optimization due to exceeding a tolerance network utilization threshold is a common policy since network providers often take actions to avoid congestion in their networks. The tolerance threshold value can be determined by the network operator's policy on the maximum utilization of inter-AS links and/or its agreement with the downstream neighboring ASes in terms of the maximum allowable volume of traffic to be sent. In summary, the PEP and BEP re-optimizers aim to maintain the network utilization under NS and any potential FS below the tolerance threshold values.

b) *Schedule driven:* the re-optimization is invoked according to the schedule defined by network operators, e.g. periodically or at specified times. In this case significant network changes may occur in the interim, resulting in poor network performance until the next re-optimization point.

In general, there is a trade-off between accuracy and monitoring overhead: the higher the accuracy, the higher the relevant overhead. Network operators may choose the most appropriate strategy according to their operational objectives.

2.2. OPTIMIZATION BLOCK

The optimization block consists of PEP and BEP re-optimizers and requires as input the latest network and traffic information from the NIB. The task of PEP re-optimizer is to re-assign the primary EPs to traffic under NS and is designed to manage dynamic traffic variation and routing changes. The key objective is to achieve inter-AS load balancing while reducing reconfiguration overheads and service disruptions. On the other hand, the task of BEP re-optimizer is to pre-compute a set of optimal backup EPs for the traffic and is designed to manage inter-AS link failure. Upon failure of an inter-AS link, the traffic affected by the failure will be shifted to the backup EPs. The key objective is to achieve inter-AS load balancing under any single inter-AS link failure while reducing backup reconfigurations.

Since changing primary EPs may cause service disruption, the operator might restrict the total number of actual PEP reconfigurations in order to reduce service disruption. On the other hand, changing backup EPs does not cause any service disruption since the primary BGP routes remain intact. However, for each re-optimization only a limited number of configuration changes might be required. Therefore, the operator might limit the total number of re-configurations (i.e. the actual number of PEP and BEP reconfigurations) per re-optimization. Details of the PEP and the BEP re-optimizers will be presented in Section 3.

2.3. IMPLEMENTATION BLOCK

The implementation block enforces the solutions produced by the PEP and BEP re-optimizer into the network based on some performance policies. A benefit-based performance policy can be applied as follows. The PEP and BEP solutions are enforced if there is a gain in reducing the worst-case MLU compared to the current configuration. To maintain the latest network information, the new PEP and BEP configurations are updated in the BGP routing virtual table in the NIB. Note that there is a tradeoff between the gain that can be obtained by reducing the EP utilization and up-to-date PEP and BEP configuration. If a large gain is chosen, not many re-optimization solutions can satisfy the required gain. This leads to less frequent PEP and BEP reconfiguration (i.e. increasing the lifetime of the current solution) and the PEP and BEP configuration tend to become obsolete. This results in a less

load balanced network, especially in the presence of failures. On the other hand, if a small gain is chosen, more re-optimization solutions with small improvement can satisfy the required gain. This results in more frequent PEP and BEP reconfiguration while keeping the configuration updated and more load balanced network. Hence, the choice of gain for solution implementation depends on how often the network operators are willing to change the network configuration and how evenly balanced they want their network to be.

3. PRIMARY AND BACKUP EGRESS POINT OPTIMIZATION

In this section, we present the optimization problem to be addressed by the PEP and BEP re-optimizers in the **IOTE SYSTEM**. We focus our TE re-optimization objective on inter-AS resources due to the reasons given in the introduction section. Table 1 shows the notation used in this paper.

Table 1. Notation used in this paper

NOTATION	DESCRIPTION
K	A set of destination prefixes, indexed by k
J	A set of egress points, indexed by j
S	A set of states $S=\{\varnothing U (\forall j \in J)\}$, indexed by s
I	A set of ingress points, indexed by i
$t(i,k)$	Bandwidth demand of traffic flows at ingress point $i \in I$ destined to destination prefix $k \in K$
$Out(k)$	A set of egress points that have reachability to destination prefix k
c^{j}_{inter}	Capacity of the egress point j
x^{j}_{sk}	A binary variable indicating whether prefix k is assigned to the egress point j in state s
y^{jl}_{sk}	A binary variable indicating whether prefix k is re-assigned to the egress point j' in state s due to re-optimization
u^{j}_{s}	Utilization on non-failed egress point j in state s. Its value is zero when $s=j$
$U_{max}(s)$	Maximum egress point utilization in state s
U_{worst}	Worst-case maximum egress point utilization across all states
R	Total primary and backup egress point reconfiguration limit
X	Total primary egress point reconfiguration limit
r_{PEP}, r_{BEP}	The number of actual primary and backup egress point reconfigurations per re-optimization

3.1. PEP RE-OPTIMIZER

3.1.1 PROBLEM FORMULATION

The PEP re-optimizer requires as input from the NIB the inter-AS traffic and BGP routing data. Note that the current selected EP for each destination prefix can be obtained from the BGP routing information.

The task of the PEP re-optimizer is to re-assign the best primary EPs for destination prefixes, with the objective of balancing the utilization among inter-AS links under normal state (s=∅) while reducing EP reconfiguration overheads and service disruptions. More specifically, the objective of inter-AS load balancing can be achieved by minimizing the inter-AS Maximum Link Utilization (MLU). Minimizing the MLU ensures that traffic is moved away from congested to less utilized links and is balanced over the links. However, minimizing inter-AS MLU and reducing EP changes (i.e. reconfigurations) are contradictory objectives: increasing the number of EP changes can reduce (i.e. improve) inter-AS MLU. As a result, balancing their trade-off is non-trivial. We therefore resort to using the $\in$-constraint method [15], which is one of the most favored methods of resolving conflicting bi-objective solutions. According to the $\in$-constraint method, the performance of one objective is

optimized, while the other one is constrained so as not to exceed a tolerance value. Since primary EP changes result in service disruption and reconfiguration overhead, we therefore choose to place a constraint on the number of EP reconfigurations that are attained by the PEP re-optimization while minimizing the inter-AS MLU. Hence, the optimization problem to be tackled by the PEP re-optimizer can be formulated with the objective:

$$Minimize\ U_{max}(\varnothing) = Minimize \underset{\forall j \in J}{Max}(u^j_{\varnothing}) = Minimize \underset{\forall j \in J}{Max}\left(\frac{\sum_{i \in I}\sum_{k \in K} x^j_{\varnothing k} t(i,k)}{c^j_{inter}}\right) \quad (1)$$

subject to the following constraints:

$$r_{PEP} \leq X \quad (2)$$

$$\forall k \in K: \sum_{j \in Out(k)} x^j_{\varnothing k} = 1 \quad (3)$$

$$\forall j \in J, k \in K: x^j_{\varnothing k} \in \{0,1\} \quad (4)$$

Constraint (2) ensures that the number of PEP reconfigurations does not exceed the PEP reconfiguration limit X. Constraints (3) and (4) ensure that only one EP is selected for each destination prefix as the PEP. The PEP re-optimization is an NP-hard problem since it is a special case of the well-known makespan problem, which is known to be NP-hard. In the rest of Section 3.1, we present a strategy to determine the PEP reconfiguration limit and an efficient algorithm to solve the problem.

3.1.2 DETERMINING THE PEP RE-CONFIGURATION LIMIT

Note that there are two possible ways of determining the PEP reconfiguration limit X. One is operator-based in which the limit can be defined according to the decision of the network operator based on its objectives. The other one is performance-based in which the limit is computed based on examining the tradeoff between minimizing the MLU and reducing the number of PEP reconfigurations. In fact, the larger the number of PEP reconfigurations, the better the expected value of the objective function (1). The examination can start with a suboptimal PEP selection solution (i.e. congestion on several EPs), then improving the solution by increasing the number of reconfiguration. As shown in Figure 3, a convex curve of MLU as a function of actual number of PEP reconfiguration can be obtained by this examination. The knee of this convex shape curve is the point that further reconfiguration beyond that point results in very small EP utilization reduction (i.e. load balancing improvement). This point can be chosen as the PEP reconfiguration limit X.

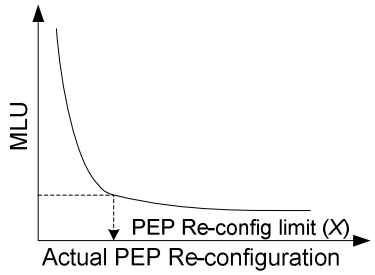

Figure 3. Determining the PEP re-configuration limit

3.1.3 PEP RE-OPTIMIZATION HEURISTIC

Since the PEP re-optimization problem is NP-hard, we need to resort to heuristic approaches. Local search algorithms have been shown to produce good results for many combinatorial optimization algorithms [15]. We therefore propose an iterative local search algorithm for the PEP re-optimizer as the following steps:

Step 1. Set r_{PEP} to zero and identify EPs with the maximum and minimum utilization ($U_{max}(\varnothing)$,$U_{min}(\varnothing)$).

Step 2. Among all the prefixes whose PEP is the EP with maximum utilization ($U_{max}(\varnothing)$), search for the prefix that by reassigning it to the EP with minimum utilization ($U_{min}(\varnothing)$) reduces the maximum EP utilization by the maximum value. Re-assign the prefix to that EP, update both values of $U_{max}(\varnothing)$ and $U_{min}(\varnothing)$, and set $r_{PEP} = r_{PEP} + 1$.

Step 3. Repeat step 2 until either r_{PEP} reaches the limit X or there is no performance improvement for $U_{max}(\varnothing)$ in comparison to the previous iteration.

3.2. BEP RE-OPTIMIZER

3.2.1 PROBLEM FORMULATION

The BEP re-optimizer requires as input the current BEP configuration as well as those inputs required by the PEP re-optimizer. The task of the BEP re-optimizer is to re-assign backup EPs for destination prefixes, with the objective of minimizing the worst-case inter-AS MLU across all FSs (we assume single inter-AS link failures) while reducing the number of backup EP reconfigurations. As mentioned earlier, changing backup EPs does not cause any service disruption. But the network operator might be able to handle only a limited total number of EP reconfigurations at each re-optimization. Therefore, if we denote the total number of PEP and BEP reconfigurations limit by R, and taking into account the actual number of PEP reconfigurations r_{PEP} imposed by PEP re-optimizer and limited to X, the total number of backup EP reconfigurations will be limited to ($R - r_{PEP}$). Hence, in a similar fashion to the PEP re-optimizer, we place a constraint on the number of backup EP reconfigurations while minimizing the worst-case inter-AS MLU. Therefore, the optimization problem in the BEP re-optimizer can be formulated with the objective:

$$Minimize\ U_{worst} = Minimize \underset{\forall s \in S}{Max} U_{\max}(s) \tag{5}$$

where

$$\forall s \in S : U_{\max}(s) = \underset{\forall j \neq s}{Max}(u_s^j) = \underset{\forall j \neq s}{Max}\left(\frac{\sum_{i \in I}\sum_{k \in K} x_{sk}^j t(i,k)}{c_{\text{inter}}^j}\right) \tag{6}$$

subject to the following constraints:

$$r_{SEP} \leq R - r_{PEP} \tag{7}$$

$$\forall k \in K, s \in S : \sum_{j \in Out(k)} x_{sk}^j = 1 \tag{8}$$

$$\forall j \in J, k \in K, s \in S : x_{sk}^j \in \{0,1\} \tag{9}$$

$$\forall j \in J, k \in K \quad if \quad x_{\varnothing k}^j = 1 \quad then \begin{Bmatrix} x_{sk}^j = 1 & \forall s \in S/\{j\} \\ x_{sk}^j = 0 & \forall s = j \end{Bmatrix} \tag{10}$$

The term $x_{sk}^j t(i,k)$ consists both of flows that are assigned to EP j as their PEP and also flows that are assigned to EP j as their BEP. Constraint (7) ensures that the number of BEP reconfigurations does not exceed the limit R-r_{PEP}. This BEP reconfiguration limit can be determined using the approach described in Section 3.1.2 for the PEP reconfiguration limit. Constraints (8) and (9) are equivalent to constraints (3) and (4), ensuring that only one EP is selected for each destination prefix as the BEP under each FS. Constraint (10) ensures that if prefix k is assigned to EP j under NS, then this prefix remains on j for all FSs except when the current FS is the failure on j. Note that, in comparison to the PEP re-optimization problem that minimizes the MLU only under NS, the BEP re-optimization problem optimizes the worst-case MLU across all the states as expressed by objective function (5). It is not surprising that the BEP re-optimisation problem is NP-hard, since it is an extension of the PEP re-optimisation problem, which itself is NP-hard.

3.2.2 BEP RE-OPTIMIZATION HEURISTIC

As with the PEP re-optimization, we also propose an iterative local search algorithm for the BEP re-optimizer. The following steps outline the proposed algorithm:

Step 1. Set r_{BEP} to zero and calculate the maximum EP utilization under each potential FS ($U_{max}(s)$).

Step 2. Identify the EP *j'* with the worst-case maximum link utilization U_{worst} under all FSs (i.e. the link with the highest $U_{max}(s)$ for all FSs). Calculate the utilization of EP *j^* with the minimum link utilization ($U_{min}(s)$) for the state when *j'* has the maximum utilization.

Step 3. Among all the prefixes whose BEP is *j'*, search for the prefix that by re-assigning it to *j^* within that state would minimize the worst-case maximum EP utilization by the maximum value. Re-assign the prefix to *j^*, update both values of $U_{max}(s)$ and $U_{min}(s)$, and set $r_{BEP} = r_{BEP} + 1$.

Step 4. Repeat steps 2 to 3 until either r_{SER} reaches the limit (R-r_{PEP}) or there is no pre-defined performance improvement for the worst-case performance in comparison to the previous iteration.

3.3. SOLUTION IMPLEMENTATION

Due to the increasing use of multi-homing by ASes, most destination prefixes can be reached through multiple EPs. When multiple routes through different EPs are present, routers select the best one according to the BGP route selection process. The BGP route selection process is based on path attributes such as *local-preference,* AS path length etc. A detailed explanation of this process can be found in [16]. The highest criterion in the BGP route selection process is the *local-preference*: the route assigned with the largest *local-preference* value is chosen as the best route to the destination prefix. Therefore, the traffic destined to a destination prefix will exit the AS through the EP that has the largest *local-preference* value.

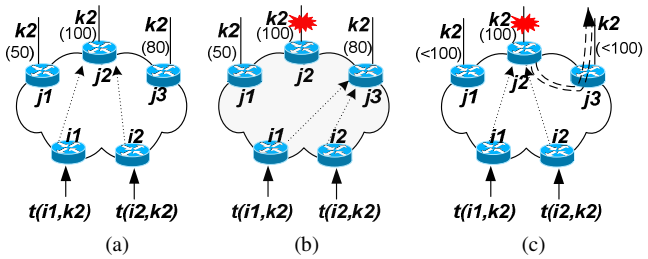

Figure 4. Traffic demand assignment (a) under NS implemented by BGP local-preference, (b) under FS implemented by BGP local-preference and (c) under FS implemented by IP tunneling for achieving fast failure recovery

The *local-preference* can be used for a simple implementation of the PEP and BEP solutions as follows: for each prefix, we assign the largest value of BGP *local-preference* for its selected PEP and the second largest value for its selected BEP. Whenever a PEP fails, the EP with the next largest *local-preference* (i.e. the BEP) becomes the exit point for the traffic towards the destinations. Figure 4 shows an example of this implementation, comprising ingress points *i1* and *i2*, egress points *j1*, *j2* and *j3*, traffic demands *t(i1,k2)* and *t(i2,k2)* and destination prefix *k2* that can each be reached through all the three EPs. Figure 4a shows the traffic demand assignment by BGP *local-preference* setting. In this case, the largest value of BGP *local-preference*, e.g. *100*, is assigned to its selected PEP (i.e. EP *j2*), the second largest value, e.g. *80*, is assigned to its selected BEP (i.e. EP *j3*) and any BGP *local-preference* value less than *80*, e.g. *50*, can be assigned to the remained EP (i.e. EP *j1*). As a result, as shown in Figure 4a the traffic demand assignment under NS is: $PEP_{t(i1,k2)} \rightarrow j2$ and $PEP_{t(i2,k2)} \rightarrow j2$. Also, as shown in Figure 4b under FS (i.e. *s=j2*) the traffic demand assignment is: $BEP_{t(i1,k2)} \rightarrow j3$ and $BEP_{t(i2,k2)} \rightarrow j3$. However, measurements from a BGP/MPLS VPN environment [17] have revealed a long BGP convergence time and network instability after the failure of an EP. Several proposals [18,19] have been made to reduce the convergence time by reducing the number of BGP messages that must be exchanged after a failure. However, as they rely on the exchange of messages, the achieved convergence time does not typically meet the stringent requirements of real time services.

In order to avoid the long BGP convergence time and achieve fast failure recovery, the fast rerouting approach proposed in [8] can be implemented in which IP tunnels are used to protect inter-AS link failures by diverting the traffic from the failed PEPs to ingress routers of the downstream ASes via the pre-computed BEP. In this approach, the IP tunnel is pre-established at the PEP and terminates at the ingress point of the downstream AS with which the pre-computed BEP is connected. An example of using IP tunneling is illustrated in Figure 4c. Assume that EP*j2* and EP*j3* are the PEP and BEP for prefix *k2*. The dash path indicates the IP tunnel. When EP*j2* first detects a failure at its attached inter-AS link, it suppresses the advertisement of this route failure to any other routers in the network. As a result, BGP convergence and its problems are eliminated and the BGP routing tables of all other routers

except EP*j2* remain intact. Afterwards, EP*j2* activates the IP tunnel and diverts the traffic to the tunnel rather than traversing the failed link. The affected traffic is delivered through the tunnel via EP*j3* and terminates at the ingress point of the downstream AS. In this case the BGP *local-preference* value of EP*j1* and EP*j3* can be any value below the BGP *local-preference* value of EP*j2*. By using IP tunneling, the traffic on failed inter-AS links can be recovered within 50 milliseconds [8]. This rapid recovery time is sufficient to sustain QoS for most of the stringent real-time services such as VoIP. Some implementation approaches of IP tunneling are discussed in [8].

4. IOTE SYSTEM PROCEDURE

Having described all the components of the **IOTE SYSTEM** in detail, we present an overall system operation as illustrated in Figure 5.

Step 1. *Network monitoring and traffic measurement*: first of all, NMTM is activated to generate global views of network and traffic conditions.

Step 2. *PEP re-optimization triggering decision making:* The triggering module is signaled to calculate the inter-AS MLU under the NS according to objective function (1). If the current MLU under the NS exceeds the tolerance threshold α1, the procedure proceeds to the next step. Otherwise it is diverted to the BEP re-optimization triggering decision making block in step 7.

Step 3. *PEP re-optimization:* the PEP re-optimizer is invoked to optimize the current PEP solution by using the proposed PEP local search heuristic algorithm.

Step 4. *PEP re-optimization stopping decision making:* if the number of required PEP reconfiguration exceeds the total PEP reconfiguration limit *X* or there is no significant performance improvement in the last iteration of local search (i.e. $\left|\frac{U_{max}^{Iteration(n)}(\varnothing)-U_{max}^{Iteration(n-1)}(\varnothing)}{U_{max}^{Iteration(n-1)}(\varnothing)}\right|<\gamma 1$), the algorithm proceeds to the next step. Otherwise the procedure goes back to the PEP re-optimization and repeats steps 3 and 4 (i.e. the PEP re-optimization cycle) till one of the stopping criteria is met.

Step 5. *PEP re-configuration decision making:* this step determines the new MLU under NS $U_{max}^{new}(\varnothing)$ from the PEP re-optimization cycle and computes the performance gain (i.e. $\left|\frac{U_{max}^{new}(\varnothing)-U_{max}^{current}(\varnothing)}{U_{max}^{current}(\varnothing)}\right|$. If the desired gain $\beta 1$ is achieved the solution is passed to the next step. Otherwise the procedure is diverted to step 7.

Step 6. *PEP configuration:* this step enforces the r_{PEP} configuration produced by the PEP re-optimizer. It updates the current MLU under NS (i.e. $U_{max}^{current}(\varnothing)=U_{max}^{new}(\varnothing)$, calculates and updates the current worst-case MLU across all potential FSs and updates the NIB with new PEP reconfigurations.

Step 7. *BEP re-optimization triggering decision making:* The triggering module calculates worst-case inter-AS MLU across all FSs (i.e. $U_{max}^{current}(\varnothing), U_{worst}^{current}$). If the current worst-case MLU exceeds the tolerance threshold $\alpha 2$, the procedure proceeds to the next step. Otherwise it is diverted back to step 1 to continue network monitoring and traffic measurement.

Step 8. *BEP re-optimization:* the BEP re-optimizer is invoked to optimize the current BEP solution by using the proposed BEP local search heuristic algorithm.

Step 9. *BEP re-optimization stopping decision making:* if the number of required BEP reconfiguration exceeds the total BEP reconfiguration limit *R*-r_{PEP} or there is no significant performance improvement in the last iteration of local search (i.e. $\left|\frac{U_{worst}^{Iteration(n)}-U_{worst}^{Iteration(n-1)}}{U_{worst}^{Iteration(n-1)}}\right|<\gamma 2$), the procedure proceeds to the next step. Otherwise the procedure goes back to the BEP re-optimizer block and repeats steps 8 and 9 (i.e. the BEP re-optimization cycle) till one of the stopping criteria have been met.

Step 10. *BEP re-configuration decision making:* this step determines the new worst-case MLU across all potential FSs (i.e. U_{worst}^{new}) from the BEP re-optimization cycle and computes the performance gain (i.e. $\left|\frac{U_{worst}^{new}-U_{worst}^{current}}{U_{worst}^{current}}\right|$). If the desired gain $\beta 2$ is achieved the solution is passed to the next step. Otherwise the procedure is diverted back to step 1 to continue network monitoring.

Step 11. *BEP configuration:* this step enforces the r_{BEP} configuration obtained from the BEP re-optimization cycle. Updates the current worst-case MLU across all potential FSs (i.e. $U_{worst}^{current} = U_{worst}^{new}$). Update the NIB with the new BEP reconfigurations. Then the procedure goes back to step 1 to continue network monitoring and traffic measurement.

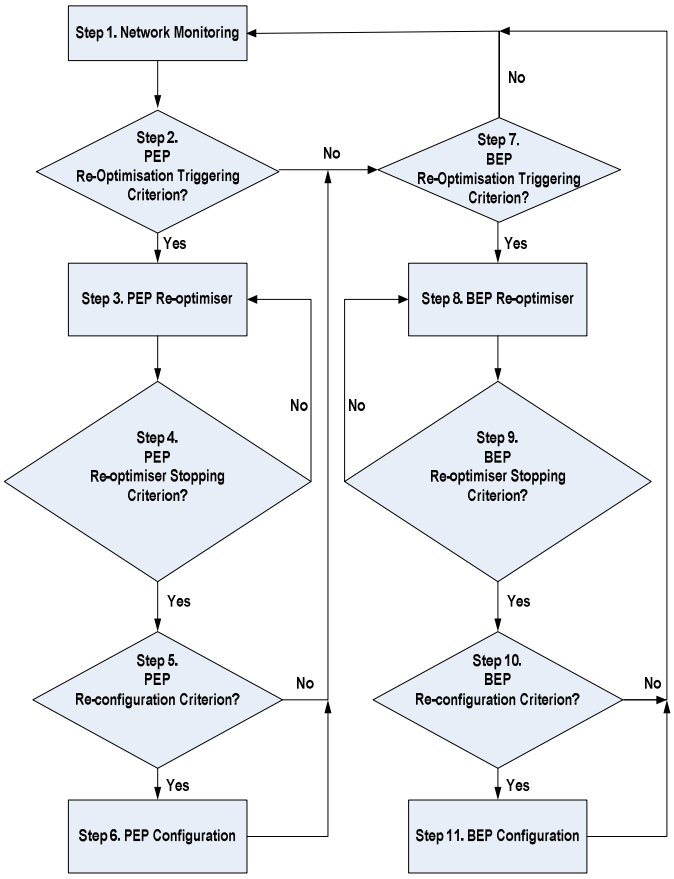

Figure 5. IOTE system procedure

5. ALTERNATIVE STRATEGIES

In this section, we present two alternative outbound TE strategies:

1) **NO-REOPT**: In this strategy neither the PEP nor the BEP re-optimization is considered. Therefore, in the presence of any network changes, the PEP and BEP configurations are always fixed.

2) **PEP-REOPT-ONLY**: this strategy solely considers the PEP re-optimization. Therefore, in case of an EP failure (transient or non-transient) and also in case of routing changes, the affected traffic will be shifted in accordance with the current BEP configuration. In comparison to **NO-REOPT**, this strategy attempts to reactively improve the network performance under non-transient FSs and routing changes, if the latest network performance obtained by the monitoring violates the threshold criterion (i.e. the network utilization exceeds the tolerance threshold). In fact, in this case, the PEP re-optimization is triggered to minimize the inter-AS MLU under the particular FS (i.e. in this special case that EP *j* has failed we have $s=j$ instead of $s=\varnothing$) or under the corresponding routing changes. Note that this strategy cannot improve the network performance in case of a transient failure due to the very short duration of the failure.

6. EVALUATION METHODOLOGY

6.1. NETWORK TOPOLOGY AND DESTINATION PREFIXES

According to [20], the typical number of ISPs that a content provider would multi-home to is not higher than 10-20. We therefore perform our experiments on topologies with 5 and 20 EPs. We assume equal EP capacities and consider their value to be OC-48 (2.5Gbps). However, our approach can be

directly applied to EPs with different capacities. For scalability and stability reasons, outbound TE can focus only on a small fraction of Internet destination prefixes, which are responsible for a large fraction of the traffic [2]. In line with [3,21], we consider 1000 such popular destination prefixes. In fact, each of them may not merely represent an individual prefix but also a group of distinct destination prefixes that have the same set of candidate EPs [22] in order to improve network and TE algorithm scalability. Hence, the number of prefixes we consider could actually represent an even larger value of actual prefixes. Recall that, according to Table 1, we can denote $|Out(k)|$ as the total number of EPs that have reachability to destination prefix k. Without loss of generality, we consider Half Prefix Reachability (HPR) i.e. $|Out(k)|=0.5|J|$ which means that each prefix k is reachable through only half of the total EPs.

6.2. INTER-AS TRAFFIC MATRIX

We generate synthetic traffic matrix for our evaluation. Our traffic matrix consists of a set of inter-AS traffic flows that originate from each ingress point towards each of the destination prefixes. Previous work has shown that inter-AS traffic is not uniformly distributed [23]. According to [22], the volume of inter-AS traffic demand is top-heavy and it can be approximated by a Weibull distribution with the shape parameter equal to 0.2-0.3. We generate the inter-AS TM following this distribution with the shape parameter equal to 0.3. We remark that our TM generation process is a simple attempt to model inter-AS traffic, as no real network based model can be found in the literature.

6.3. IOTE SYSTEM PARAMETERS

We realized that by setting the **IOTE SYSTEM** parameters to the following values we can achieve sufficiently good results: we set the tolerance thresholds α1 and α2 to α1=α2=50% as the borderline of congestion to trigger PEP and BEP re-optimizations. This chosen tolerance threshold value is inline with the resource management rule of some ISPs such as Sprint that aim to maintain the average utilization of any link under 50% [24]. For the re-configuration limits, we opt for the operator-based approach since ISPs generally aim to reduce service disruptions [25]. Hence, we assume that for each PEP re-optimization only up to 10% of the total destination prefixes can be disrupted. In other words, $X = 0.1 \times 1000 = 100$. We also assume that for each re-optimization only up to 30% of the total destination prefixes can be changed or reconfigured. In other words, $R = 0.3 \times 1000 = 300$. This results in the fact that depending on the actual number of PEP re-configurations, only between 20% and 30% of the total destination prefixes can have their BEP to be reconfigured by the BEP re-optimizer at each re-optimization. For the stopping criterion of the local search of PEP and BEP re-optimizer, we consider the pre-defined performance improvement γ1 and γ2 to be γ1=γ2=5%.

6.4. PERFORMANCE METRICS

The following metrics are used in our evaluation. For all these metrics, lower values are better than high values.

- **Inter-AS MLU**: This refers to both $U_{max}(\varnothing)$ under NS and the $U_{max}(s)$ under FS s in objective functions (1) and (6) respectively.
- **Service disruption per re-optimization**: A traffic flow (service) is disrupted if it is shifted from EP j to EP j' due to re-optimization. We calculate this metric by adding the volume of all traffic flows disrupted for the PEP re-optimization. This metric can be formulated as follows:

$$ServiceDisruption = \sum_{j \in J} \sum_{i \in I} \sum_{k \in K} x_{sk}^{j} t(i,k) \sum_{j' \in J / \{j\}} y_{sk}^{j'} \tag{11}$$

- **Number of actual PEP and BEP reconfigurations per re-optimization:** These refer to r_{PEP} in (2) and r_{BEP} in (7) respectively.

6.5. GENERATED EVENTS

Since no realistic model has been investigated for changes in network conditions, such as traffic variations, routing changes, inter-AS transient failures (TF) and non-transient failures (NTF), we generate various series of random events that attempt to emulate those realistic changes by assigning an occurrence probability to each event. In addition, due to possibly changes of user behaviour and varying demand for different services [7], gradual traffic changes are quite frequent. As a result we consider half of the event intervals to include the gradual changes in traffic while the other half to include just small traffic fluctuations. We assume that these intervals are randomly distributed.

Moreover, according to several relevant findings in [2,7,8,10,26], events such as TF, NTF, Sudden Traffic Increase (STI), Sudden Traffic Decrease (STD) and Routing Changes (RC) occur in addition to the small traffic fluctuations and gradual changes. By summarizing the references, we found out that TF is the most common event [8]. Hence, a high occurrence probability may be assigned to it. While events like NTF and RC happen quite rarely. For example there are rare possibilities of fiber-cut which are responsible for NTFs [10] and rare possibilities of routing changes due to the stable nature of popular prefixes [2]. In addition, sudden traffic variations (STI, STD) are relatively rare [26]. This is not surprising because large ISPs carry significant volumes of highly aggregated traffic. However, some traffic matrix elements vary by a significant amount several times a week [7]. These traffic variations can have many causes including flash crowd, denial-of-service attacks and routing changes in other ASes [26]. As a result, equal low occurrence probabilities may be assigned to NTF, RC, STI and STD. The performance of all the strategies under these events is investigated in the next section.

7. EVALUATED RESULTS

7.1. INTER-AS MLU

In this section, we investigate the performance of all the strategies under two sets of various events for 5-EP and 20-EP topologies. Each set consists of ten intervals with randomly generated events based on their occurrence probabilities.

Figures 6a-6c show the set of randomly generated events, the total underlying traffic during the events and the transient and non-transient failures occurred during the events respectively for 5-EP topology. The randomly generated events occur in the following order: The first interval is a period of small traffic fluctuations together with 1 TF. The second interval starts with a sudden traffic increase followed by a period of small traffic fluctuations together with 2 TF and 1 NTF. The third interval starts with sudden routing changes followed by a period of gradual traffic decrease together with 2 TFs. The fourth interval is a period of gradual traffic increase together with 2 TFs. The fifth interval starts with sudden routing changes followed by a period of gradual traffic decrease together with 3 TFs. The sixth interval starts with a sudden traffic increase followed by a period of small traffic fluctuations together with 1 NTF and 1 TF. The seventh interval starts with a sudden downward traffic surge followed by a period of gradual traffic increase together with 2 TFs. The eight interval starts with a sudden downward traffic surge followed by a period of traffic decrease together with 2 TFs. The ninth interval starts with a sudden routing changes followed by a period of small traffic fluctuations together with 2 TFs and 1 NTF. The tenth interval starts with a sudden downward traffic surge followed by a period of small traffic fluctuations together with 2 TFs. Furthermore, Figures 7a-7c show the inter-AS MLU under NS and FSs achieved by **NO-REOPT**, **PEP-REOPT-ONLY** and **IOTE SYSTEM** respectively. The x axis represents the positions of the random events from time t_0 till time t.

Figures 7a-7c show that during the first interval all the strategies perform identical both under NS and FS. This is due to our assumption that all the strategies start with the same initial solutions for fair comparisons. However, once the monitored performance violates the re-optimization triggering threshold value (i.e. 50%), they start to react differently.

Figure 7a shows that the **NO-REOPT** is the worst performer under all the events and not only cannot keep the inter-AS MLU under NS below the threshold value but also its MLU under FSs has dramatically poor performance. This phenomenon was expected due to the fact that this strategy does not perform any re-optimization to achieve load balancing. As a result, its initial PEP and BEP solutions become vulnerable to the subsequent changes in the network conditions such as accumulation of traffic matrix variations and routing changes.

In contrast, Figure 7b shows that the **PEP-REOPT-ONLY** can keep the inter-AS MLU only under NS below the threshold value[1]. However, since this strategy ignores BEP re-optimization, its MLU under FSs becomes poor and gets worse after subsequent events. Nevertheless, the overall FS network performance degradation in the **PEP-REOPT-ONLY** is less severe than in **NO-REOPT**. This result is expected since the **NO-REOPT** does not apply any re-optimization as a result the failure of a congested EP and the

[1] Note that in **PEP-REOPT-ONLY** and **IOTE SYSTEM**, the inter-AS MLU under NS or FS might exceed the tolerance threshold due to sudden changes. Nevertheless, both strategies are able to minimize the utilization below the tolerance threshold after the re-optimization under the condition where there exist sufficient capacity to accommodate the latest overall traffic demands.

assignment of its traffic flows over the non-optimized BEP may result in the re-assignment of a large number of traffic flows over already congested EPs, causing a significant performance degradation. On the contrary, in the **PEP- REOPT-ONLY,** an EP failure and the re-assignment of its flows over the non-optimized BEP does not lead to much performance degradation due to the fact that the EPs are balanced under NS by PEP re-optimization. Moreover, the **PEP-REOPT-ONLY** improves the MLU by PEP re-optimization when it exceeds the threshold value after NTFs in intervals 2, 6 and 9. In total, Figure 7b shows 7 PEP re-optimizations to improve the MLU after the traffic variations (2 PEP re-optimizations), after routing changes (2 PEP re-optimizations) and after the 3 NTFs (3 PEP re-optimizations). However, Figure 7c shows that the **IOTE SYSTEM** can keep the MLU not only under NS but also under most of the FSs below the threshold value by re-optimizations[1]. In fact, it can improve the MLU both for TFs and NTFs by BEP re-optimization. Its FS worst-case performance is respectively 35% and 15% better than the FS worst-case performance of the **NO-REOPT** and the **PEP-REOPT-ONLY**.

Note that in the **IOTE SYSTEM** the inter-AS MLU under FSs is proactively re-optimized for both TFs and NTFs. In other words, in this system the backup EPs for all the potential FSs are pre-computed according to the network dynamic changes in order to balance the link load under these states and alleviate link congestion due to failure. By comparison, in the **PEP-REOPT-ONLY**, there is no re-optimization for TFs due to their very short duration[2] but there are reactive re-optimizations for NTFs. As a result, the significant performance degradation shown in Figure 7b due to TFs and NTFs do not occur in Figure 7c. Furthermore, in the **IOTE SYSTEM** the network performance degradation under sudden routing changes in intervals 3, 5 and 9 are not as serious as the one in the **PEP-REOPT-ONLY**. The reason for this phenomenon is that after routing changes (i.e. changes of some destination prefixes reachability at some EPs), the affected traffic flows will be shifted to their current BEP. The BEP re-optimization performed in the **IOTE SYSTEM** at the earlier stages (i.e. before routing changes) alleviates the performance degradation in comparison to no BEP re-optimization in the **PEP-REOPT-ONLY**. However, in both approaches if the network performance after the re-assignment of traffic flows exceeds the tolerance threshold, PEP re-optimization is triggered. In this case, the BEP re-optimization for **IOTE SYSTEM** might be triggered as well if the worst-case MLU across all the potential FSs exceeds the tolerance threshold . In total, Figure 7c shows 4 PEP and 7 BEP re-optimizations. Note that among the 7 required BEP re-optimizations, 4 of them happen immediately after their corresponding PEP re-optimizations while another 3 BEP re-optimizations occurs individually. The reason is that under some certain network conditions the inter-AS MLU only under the potential FSs might exceed the tolerance threshold value. In this case only BEP re-optimization is required.

The other set of randomly generated events with their underlying TM, TFs and NTFs are shown in Figures 8a-8c for 20-EP topology. Figures 9a-9c show the inter-AS MLU under NS and FS *s* achieved by **NO-REOPT**, **PEP-REOPT-ONLY** and **IOTE SYSTEM** respectively based on the events shown in Figure 8a for the 20-EP topology. An overall comparison of Figures 7a-7c with 9a-9c reveals that the same conclusions can be derived for 20-EP topology. On the whole, our proposed **IOTE SYSTEM** achieves (1) much better performance in terms of the inter-AS MLU under NS in comparison to the **NO-REOPT** (i.e. its worst-case NS performance is 14% and 30% better for 5 and 20-EP respectively) and almost the same performance as the **PEP-REOPT-ONLY**, (2) significantly better performance in terms of the inter-AS MLU under FSs compared to **NO-REOPT** (i.e. its worst-case FS performance is 35% and 41% better for 5 and 20-EP respectively) and better performance compared to **PEP-REOPT-ONLY** (its worst-case FS performance is 15% and 10% better for 5 and 20-EP respectively).

---

[2] If a TF happens at the time of network conditions monitoring and results to the tolerance threshold violation, the PEP re-optimization is triggered. However, since the TF has a very short duration, it is recovered earlier than the configuration can take place. At this point network operator could simply ignore such re-optimization. In this paper, we assume that the network operator takes care of this task and therefore no re-optimization is applied due to TFs.

STI	RC			RC	STI	STD	STD	RC	STD	
1 TF	2 TF 1 NTF	2 TF	2 TF	3 TF	1 TF 1 NTF	2 TF	2 TF	2 TF 1 NTF	2 TF	
STF	STF	GTD	GTI	GTD	STF	GTI	GTD	STF	STF	

Figure 6a. The set of randomly generated events for 5-EP topology

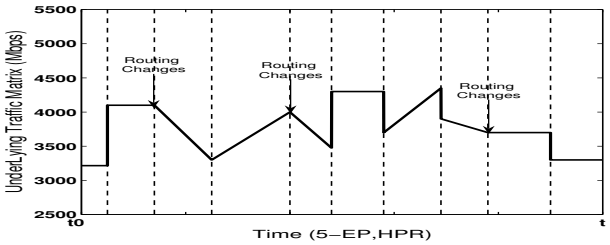

Figure 6b. Total underlying traffic during the events for 5-EP topology

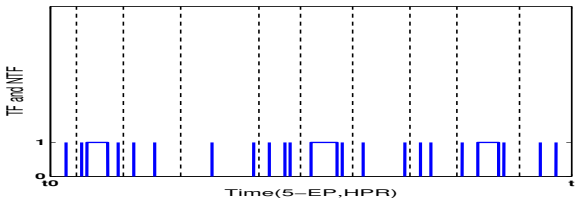

Figure 6c. Transient and Non-Transient Failures for 5-EP topology

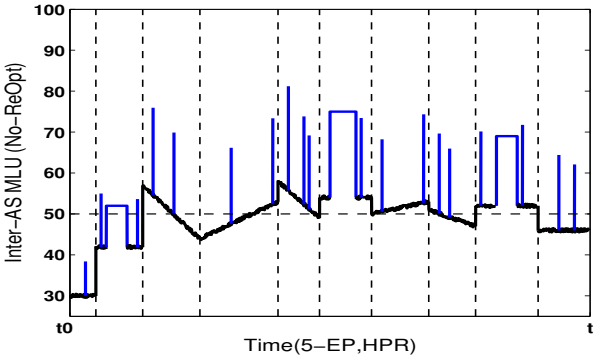

Figure 7a. Inter-AS MLU performance of **NO-REOPT** for 5-EP

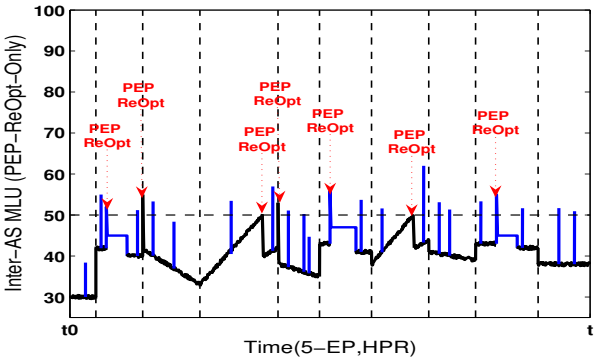

Figure 7b. Inter-AS MLU performance of **PEP-REOPT-ONLY** for 5-EP

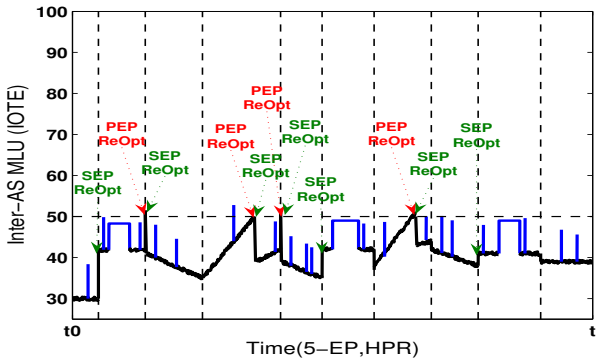

Figure 7c. Inter-AS MLU performance of **IOTE SYSTEM** for 5-EP

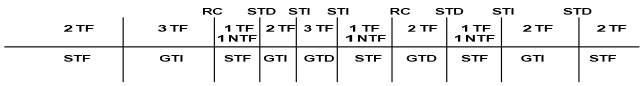

Figure 8a. The set of randomly generated events for 20-EP topology

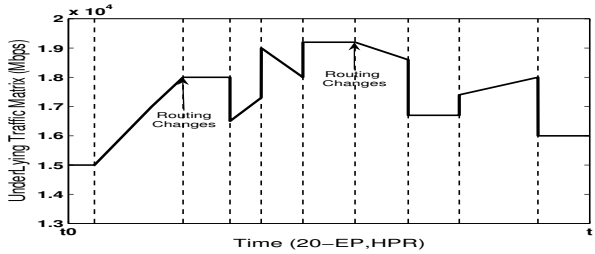

Figure 8b. Total underlying traffic during the events for 20-EP topology

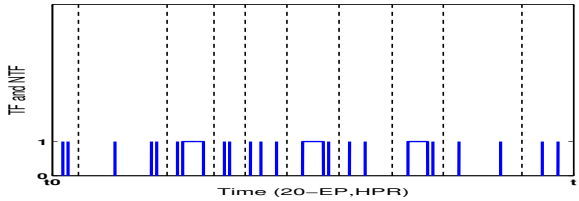

Figure 8c. Transient and Non-Transient Failures for 20-EP topology

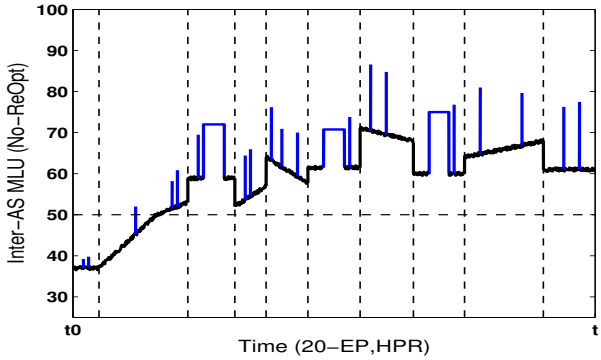

Figure 9a. Inter-AS MLU performance of **NO-REOPT** for 20-EP

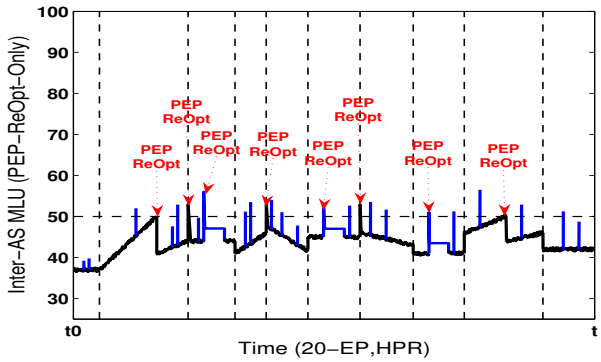

Figure 9b. Inter-AS MLU performance of **PEP-REOPT-ONLY** for 20-EP

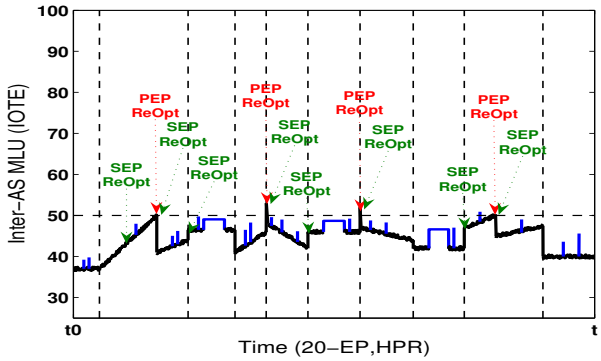

Figure 9c. Inter-AS MLU performance of **IOTE SYSTEM** for 20-EP

7.2. RE-OPTIMIZATON COST METRICS

In this section, we compare the re-optimization cost metrics (i.e. *Service Disruption,* r_{PEP}, r_{BEP}) of the **PEP-REOPT-ONLY** and the **IOTE SYSTEM**. Obviously, for the **NO-REOPT**, all these cost metrics are zero since this strategy does not perform any re-optimization.

Table 2. Re-optimization cost metrics for **PEP-REOPT-ONLY** and **IOTE SYSTEM** for 5-EP topology

Interval	*Event*	*Service Disruption*		r_{PEP}		r_{BEP}	
		PEP	***IOTE***	***PEP***	***IOTE***	***PEP***	***IOTE***
2	***STI***	*0*	*0*	*0*	*0*	*0*	*200*
2	***NTF***	*350*	*0*	*37*	*0*	*0*	*0*
3	***RC***	*675*	*525*	*100*	*95*	*0*	*200*
4	***GTI***	*500*	*550*	*100*	*100*	*0*	*150*
5	***RC***	*750*	*600*	*100*	*69*	*0*	*230*
6	***STI***	*0*	*0*	*0*	*0*	*0*	*175*
6	***NTF***	*445*	*0*	*100*	*0*	*0*	*0*
7	***GTI***	*400*	*350*	*70*	*60*	*0*	*200*
9	***RC***	*0*	*0*	*0*	*0*	*0*	*150*
9	***NTF***	*550*	*0*	*100*	*0*	*0*	*0*
Total	***-***	***3670***	***2025***	***607***	***324***	***0***	***1305***

Table 3. Re-optimization cost metrics for **PEP-REOPT-ONLy** and **IOTE SYSTEM** for 20-EP topology

Interval	*Event*	*Service Disruption*		r_{PEP}		r_{BEP}	
		PEP	***IOTE***	***PEP***	***IOTE***	***PEP***	***IOTE***
2	***GTI***	*0*	*0*	*0*	*0*	*0*	*190*
2	***GTI***	*1556*	*1410*	*65*	*51*	*0*	*200*
3	***RC***	*2000*	*0*	*100*	*0*	*0*	*210*
4	***NTF***	*850*	*0*	*48*	*0*	*0*	*0*
5	***STI***	*2150*	*1850*	*100*	*100*	*0*	*200*
6	***STI***	*0*	*0*	*0*	*0*	*0*	*175*
6	***NTF***	*950*	*0*	*43*	*0*	*0*	*0*
7	***RC***	*1575*	*800*	*100*	*90*	*0*	*200*
8	***NTF***	*785*	*0*	*28*	*0*	*0*	*0*
9	***STI***	*0*	*0*	*0*	*0*	*0*	*150*
9	***GTI***	*1000*	*450*	*87*	*68*	*0*	*100*
Total	***-***	***10866***	***4510***	***571***	***309***	***0***	***1425***

In Tables 2 and 3, each row represents the *N*th interval in which re-optimization occurs. The second column represents the type of event that causes the re-optimization and the other columns represent the re-optimization cost metrics. The two tables deal separately with 5-EP and 20-EP topologies. In each metric column, the first value corresponds to the **PEP-REOPT-ONLY** and the second value corresponds to the **IOTE SYSTEM**.

Both Tables show that in total the **PEP-REOPT-ONLY** has higher service disruption and more PEP reconfigurations compared to the **IOTE SYSTEM**. This result was expected since the **PEP-REOPT-ONLY**

attempts to re-optimize the network performance degradation due to NTFs by PEP re-optimization after the failure, which results in three more PEP re-optimizations that corresponds to the 2nd , 6th , 9th intervals in Table 2 for 5-EP and 3nd , 6th , 8th intervals in Table 3 for 20-EP. In the **IOTE SYSTEM** the proactive BEP re-optimizations that occur at the beginning of these intervals take care of the NTFs and result to zero service disruption and PEP re-optimizations for these events. Moreover, since the **PEP-REOPT-ONLY** does not perform any BEP re-optimization, it requires more PEP reconfiguration for re-optimizing the network performance after sudden routing changes which corresponds to the 3rd and 5th intervals in Table 2 for 5-EP and the 3rd and 7th intervals in Table 3 for 20-EP. In these intervals the service disruption and PEP re-configuration are more than the **IOTE SYSTEM**. In fact, in the **IOTE SYSTEM** the proactive BEP re-optimizations alleviate the routing changes effects and result in less service disruption and re-configurations in the corresponding intervals. However, the routing changes themselves have led to BEP re-optimizations in **IOTE SYSTEM** to rebalance the load in case of the upcoming potential failures.

In summary, for the 5-EP topology, the **IOTE SYSTEM** incurs almost 45% less service disruptions, and 46% less PEP reconfigurations compared to the **PEP-REOPT-ONLY** at the cost of 1305 BEP reconfigurations, to keep the network performance under FSs more load balanced. Also for the 20-EP topology the **IOTE SYSTEM** incurs almost 58% less service disruptions and 46% less PEP reconfigurations compared to the **PEP-REOPT-ONLY** at the cost of 1425 BEP reconfigurations. We recall that the BEP reconfiguration does not cause service disruption. In addition, less service disruptions and PEP reconfigurations in our system may imply better network stability compared to the **PEP-REOPT-ONLY**.

8. CONCLUSION

In this paper, we have addressed the problem of existing outbound TE solutions in case of dynamic changes in network conditions such as traffic variations, routing changes and inter-AS link failures. Hence, we have proposed an Inter-AS Outbound Traffic Engineering (IOTE) system that aims to achieve robustness by balancing the load on inter-AS links under both normal and failure states, while at the same time reducing service disruption and reconfiguration overheads. We developed time-efficient heuristics to achieve the system objectives and compared its performance to two alternative strategies. Our evaluation results show that our proposed system performs better in comparison to the alternative strategies.

We believe that our work provides insights to network operators on how to keep a balanced network especially under inter-AS link failures in spite of traffic variations and inevitable routing changes by limiting egress point reconfigurations. The proposed approach is in line with the current ISP practice of off-line traffic engineering but goes a step further in continuously re-optimizing the TE configuration based on monitored BGP route changes and traffic load of inter-AS links through a closed-loop control approach. The latter lies in-between off-line proactive and on-line re-active approaches as it uses pro-active re-configuration driven by real-time monitoring. This continuous re-optimization results in a balanced network that has enough pre-planned capacity to handle a single, transient or non-transient, inter-AS link failure without congestion and subsequent service disruptions, something invaluable for real-time multimedia services but also beneficial for interactive data services.

REFERENCES

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

[2] N. Feamster, J. Borkenhagen and J. Rexford, "Guidelines for Inter-domain Traffic Engineering," ACM CCR, October 2003

[3] T.C. Bressound and R. Rastogi, "Optimal Configuration for BGP Route Selection," *Proc. IEEE INFOCOM*, 2003, pp. 916-926.

[4] S. Uhlig, O. Bonaventure and B. Quoitin., "Interdomain Traffic Engineering with Minimal BGP Configurations," *Proc. 18th International Teletraffic Congress*, 2003.

[5] K. Ho, N. Wang, P. Trimintzios and G. Pavlou, "Multi-objective Egress Router Selection Policies for InterAS Traffic with Bandwidth Guarantees," *Proc. IFIP Networking*, 2004, pp. 271-283.

[6] S. Saroiu, K. P. Gummadi, R.J. Dunn, S.D. Gribble and H.M. Levy, "An analysis of Internet Content Delivery System," Proc. *USENIX Operating Systems Design and Implementation (OSDI)*, 2002, pp. 315-327.

[7] R.Teixeira, N. Duffield, J. Rexford and M. Roughan, "Traffic Matrix Reloaded: Impact of Routing Changes," *Proc. Passive and Active Measurement Conference*, 2005, pp. 251-264.

[8] O. Bonaventure, C. Filsfils and P. Francois., "Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures," *IEEE/ACM Transactions on Networking*, 15(5), October 2007, pp. 1123-1135.

[9] S. Uhlig and O. Bonaventure, "Designing BGP-based Outbound Traffic Engineering Techniques for Stub ASes" *ACM SIGCOMM Computer Communication Review,* October 2004, pp. 89-106.

[10] A. Sridharan and R. Guerin., "Making IGP Routing Robust to Link Failures," *Proc. IFIP Networking*, 2005, pp. 634-646.

[11] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft and C. Diot, "IGP Link Weight assignment for Operational Tier-1 Backbones," *IEEE/ACM Transactions on Networking*, 15(4), August 2007, pp. 789-802.

[12] M. Amin, K. Ho, M. Howarth and G. Pavlou, "An Integrated Network Management Framework for Inter-domain Outbound Traffic Engineering", Proc. *IEEE/IFIP MMNS*, 2006.

[13] S. Willis, J. Burruss, and J. Chu, "Definitions of managed objects for BGP-4", IETF RFC 1657, 1994.

[14] R. Teixeira, T.G. Griffin, M.G.C. Resende and J. Rexford, "TIE Breaking: Tunable Interdomain Egress Selection," *IEEE/ACM Transactions on Networking*, 15(4), 2007, pp. 761-774.

[15] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice Hall-Inc., 1982.

[16] S. Halabi and D. McPherson, *Internet Routing Architectures*, Cisco Press, 2nd ed., 2001.

[17] C. Filsfils, "IGP and BGP fast convergence", Networkers' 2004, Cannes, France, December 2004.

[18] D. Pei, M. Azuma, D. Massey and L. Zhang, "BGP-RCN: Improving BGP Convergence through Root Cause Notification," *Computer Networks*, 48(2), 2005, pp. 175-194.

[19] T.G. Griffin and B.J. Premore, "An Experimental Analysis of BGP Convergence Time," Proc. *IEEE ICNP*, 2001, pp. 53-61.

[20] A. Dhamdhere and C. Dovrolis, "ISP and Egress Path Selection for Multi-homed Networks," *Proc. IEEE INFOCOM*, 2006, pp. 1-12.

[21] S. Uhlig and B. Quoitin, "Tweak-it: BGP-based Interdomain Traffic Engineering for Transit ASes," *Proc. NGI Conference,* 2005.

[22] A. Broido, Y. Hyun, R. Gao and KC. Claffy, "Their Share: Diversity and Disparity in IP Traffic," Proc. *Passive and Active Measurement Conference*, 2004, pp. 113-125.

[23] W. Fang and L. Peterson, "Inter-AS Traffic Patterns and their Implications," *Proc. IEEE GLOBECOM,* 1998, pp. 1859-1868.

[24] G. Iannaccone, C.N. Chuah, and C. Diot , "Feasibility of IP Restoration in a Tier-1 Backbone," *IEEE Network*, 18(2), March-April 2004, pp. 13-19.

[25] R.Teixeira and J. Rexford, "Managing Routing Disruptions in Internet Service Provider Networks," *IEEE Communication Magazine*, 44(3), March 2006, pp. 160-165.

[26] R.Teixeira, S. Agarwal and J. Rexford, "Routing Changes: Merging Views from Two ISPs," *ACM SIGCOMM Computer Communication Review*, October 2005, pp. 79-82.