Connectionless Building Blocks For Intra- and Inter-Domain Traffic Engineering

Shiv Kalyanaraman, Hema T. Kaur, Mehul Doshi, Ayesha Gandhi, Niharika Mateti
ECSE Department, Rensselaer Polytechnic Institute,
Troy, NY-12180
{shivkuma, hema}@networks.ecse.rpi.edu,
{doshim2, gandhia}@rpi.edu

Abstract

We propose new building blocks for connectionless intra-domain and inter-domain traffic engineering (TE) in the Internet. The key idea is to capture an intra-domain path, AS-path or an exit route as a 32-bit hash in the packet header. This hash allows explicit "source" directed routing without signaling or high per-packet overhead, while enabling an incremental upgrade strategy for OSPF and BGP. This paper overviews the building blocks and focuses on the inter-domain case, where we consider the problems of a) explicit border router choice and b) an explicit AS path choice. The latter problem in general requires a tradeoff in terms of increased inter-AS control traffic, whereas the former problem can be solved within an AS with partial upgrades. Such explicit routing has the potential to allow a more direct, finer-grained policy control of how traffic is mapped to routes. Simulation and implementation/experimental results illustrate the operation of these building blocks.

1 Introduction

Traffic engineering (TE) is defined as *...that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks...* [1], [2]. We take a *more limited, weaker* view of TE, but consider a *broader* range of deployment scenarios. In particular, we are interested in explicit source-directed path choice on a per-packet basis. To achieve this, we propose a parsimonious encoding of the path choice in packets *without* the need for *signaling* or fork-lift router upgrades in the network. The key idea of the paper is capture an intra-domain path, inter-domain AS-path or an exit route from an AS as a 32-bit hash in the packet header. Our proposed building blocks can be adapted for either intra-domain or inter-domain operation.

2 PathID Concept

Consider a network modeled as a graph in Figure 1 with links and nodes, where links are given weights (not necessarily unique). Consider a path from node i to node j, which passes through nodes $i, 1, 2, ..., m-1, j$ and links of weights $w_1, w_2, ..., w_m$. This path can be represented as a sequence: $[i, w_1, 1, w_2, 2, ..., w_m, j]$, and a path suffix of this path from node k to j represented as the sequence: $[k, w_{k+1}, ..., w_m, j]$. This path sequence can be parsimoniously represented by a *hash function of the elements of the sequence (or a subsequence)*. These concepts are illustrated in Figure 1.

In the case of intra-domain routing (e.g. OSPF or IS-IS), observe that the node IDs (i.e. router IDs) and link weights are known at all routers (i.e. they are *globally known constants*). In the case of BGP, if we are interested in choosing explicit AS-paths, then the node IDs above could be considered as the AS numbers (ASNs) which are well known for each AS-path that is available. If we are interested in an exit AS-border router (a.k.a Exit-ASBR), then the hash is simply the exit-ASBR IP address. Hashing such a sequence of *globally known* quantities allows us to avoid signaling because each upgraded router on the path can unambiguously interpret the hash. Recall that one purpose of signaling in ATM and MPLS is to map global IDs (ad-

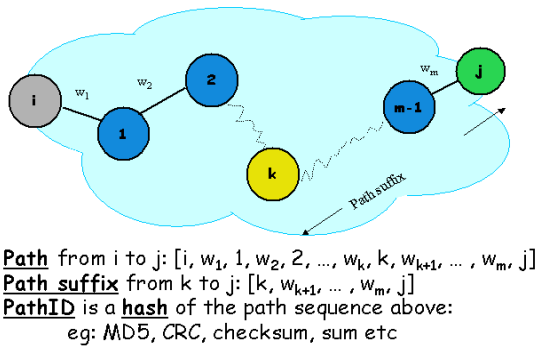

Figure 1: Path, Path Suffix and PathID Concepts

dresses, path specifications) to local IDs (labels). Since we obtain our local IDs from the global IDs through the hashing procedure, *signaling for path selection is not necessary*.

The choice of the hash function is dictated by the need to minimize the collision probability which directly affects the uniqueness and utility of the hash. A simple hash of the path sequence is the sum of link weights, but it may lead to a high collision probability. Therefore for intra-domain routing, we propose to use a 128-bit MD5 hash of the nodeIDs along the path, followed by a 32-bit CRC of the 128 bit MD5 hash to result in a 32-bit hash field. This (MD5 + CRC-32) hash in conjunction with the destination address (j) already in the packet forms our concept of intra-domain PathID. If the sequence of node IDs along the path is unique, then by the properties of the MD5 and CRC-32 hash functions, the hash and the PathID tuple = [j, hash] is very highly likely to be unique. For the inter-domain case, to represent AS-paths, we propose to take a similar (MD5 + CRC-32) hash of the sequence of ASNs in the AS-path. For exit router choice in inter-domain routing, the hash is simply the IP address of the exit ASBR.

The abstract forwarding concept is as follows. Routers which are not upgraded have forwarding table entries of the form [destination prefix, outgoing interface], as usual. Upgraded routers have forwarding table entries of the form [destination prefix, incoming hash, outgoing interface, outgoing hash]. The "incoming hash" field represents the hash of the explicit path starting from the current router to the destination prefix. The "outgoing hash" field is the hash of the path suffix from the *next upgraded router* on the explicit path to the destination. Incoming packets at upgraded routers may come with a path ID = [j, hash] if an explicit path is chosen for the packet, or with a regular destination address field j and no hash field. The hash field may be stored in a new routing header in IP packets.

The router first matches the destination IP address using the longest prefix match procedure, and then it does an *exact match* of the hash for that destination. If matched, the hash in the packet is replaced with the outgoing hash, and the packet is sent to the outgoing interface. Observe that this procedure is a hybrid of IP's longest prefix match and MPLS's label swapping, but using the hash instead of labels and without the need for signaling. If the exact match is not found (i.e. errant hash value in packet), then the hash value in the packet is set to zero, and the packet is sent on the default path (i.e. shortest path in OSPF or default policy route in BGP).

3 Explicit Forwarding in OSPF and BGP

In the following three subsections, we deal only with the forwarding issues and postpone the discussion of important control plane issues to section 4. In other words, we assume that upgraded routers have forwarding entries for the paths corresponding to pathIDs indicated in packets.

3.1 Explicit Path Forwarding in OSPF

The forwarding in an intra-domain setting closely follows the abstract algorithm described earlier. Figure 2 shows the topology of a simple validation experiment conducted on Utah's Emulab [8] testbed with the Linux Zebra version of OSPF upgraded with our traffic engineering building blocks. The forwarding plane was implemented in Linux using MIT's Click Modular Router package [7].

Table 1 illustrates a partial forwarding table at node 1 (IP addr = 192.168.1.1) for destination 3 (192.186.3.3). The PathIDs are the (MD5 + CRC-32) hashes of the router IDs (i.e. IP addresses of nodes) on the path. For example, the PathID 2084819824 corre-

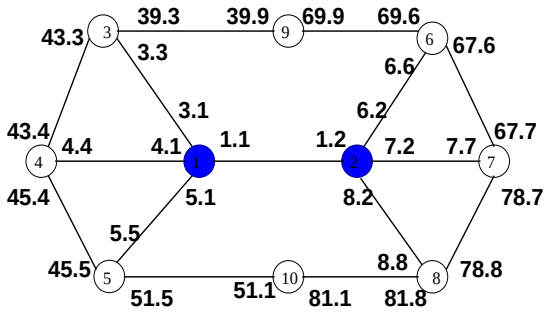

Figure 2: Experimental Topology on Utah Emulab using Linux Zebra/Click Platforms

sponds to a hash of the set of Router IDs {192.168.1.1, 192.168.1.2, 192.168.6.6, 192.168.39.9, 192.168.3.3 }. The Router ID is statically defined to be same as ip-address of one of the router interfaces. However, for simplicity, we have chosen the smallest ip-address interface to define the Router ID. The Path Suffix ID is the hash of the suffix set formed after omitting 192.168.1.1. If the path goes through other nodes which are not upgraded (e.g. 1-4-3), the suffix path ID is the hash of the suffix path starting from the next upgraded router on the path. In the case of the path 1-4-3, both nodes 4 and 3 are not upgraded, so the suffix path ID is zero.

Outgoing I/f	Path	PathID	PathSuffixID
192.168.1.1	1-2-6-9-3	2084819824	664104731
192.168.3.1	1-3	599270449	0
192.168.4.1	1-4-3	4183108560	0
192.168.5.1	1-5-4-3	1365378675	0

Table 1: Partial routing table at 192.168.1.1 for destination 192.186.3.3

3.2 Explicit AS-Path Forwarding in BGP

For inter-domain TE using BGP, we consider two cases: *explicit AS-Path* forwarding and *explicit exit* forwarding. The explicit AS-path model extends the explicit path model for intra-domain routing, but uses AS-path instead. The explicit exit model allows only a choice of Exit AS border router (ASBR), but it cannot specify an explicit choice of an AS-path at that ASBR. Obviously the latter model, though limited in scope, is more realistic on the short-term since BGP is a policy routing protocol used independently by AS'es. However, we explore both models in this paper. As mentioned earlier, this section will explore only the forwarding plane issues; control plane issues are discussed in Section 4. We have implemented both models in SSFnet simulation.

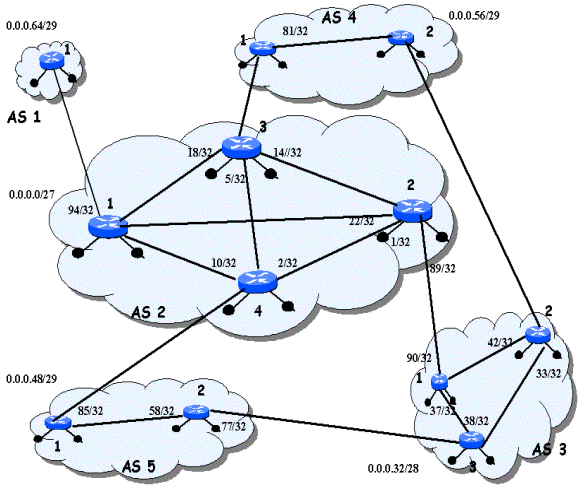

Figure 3: Inter-Domain TE Simulation Topology In SSFnet

Unlike nodes in our abstract forwarding model, AS'es contain multiple BGP routers. We assume that the AS-path is encoded as a hash, that is matched as usual at the *entry* AS border router. But, we propose to *swap the incoming hash with the outgoing AS-path-suffix hash only at the exit AS border router*. The exit border router for that AS-path is the BGP router which learns the AS-path from an *external peer*, i.e., its origin attribute is EGP for that AS-path.

Consider the AS-graph topology in Figure 3, and assume that we would like to send traffic from AS1 to AS5, i.e. to the IP prefix 0.0.0.48/29. The AS-paths available are AS1-AS2-AS5, AS1-AS2-AS4-AS3-AS5, AS1-AS2-AS3-AS5. For simplicity, consider the AS-path AS1-AS2-AS5, represented as (1 2 5) that is chosen at router 1 in AS1. The suffix AS-path is (2 5) whose hash is 4038336721 as indicated in Table 2; this value is placed in a field called EPATHID in the outgoing IP packet. Note that, in the tables, we omit the leading zeros in the IP addresses. The next hop for this packet is 94/32 that is an entry router in AS2, and Table 3 is consulted. The second entry of Table 3 matches the destination prefix and EPATHID.

The next hop is 10/32 that is an exit router from AS2 to reach AS3. Observe that since this route (at Router 1 in AS 2) is learned from IBGP, the EPATHID is left unchanged. The EPATHID will be swapped only at the exit ASBR (i.e. Router 4 in AS2). At this exit router, the exact match of the prefix and EPATHID results in a next hop of 85/32 in AS5, and outgoing EPATHID will be set to 0 since it is the final AS.

Dst	NextHop	Learnt From	EPATHID	AS-PATH	Outgoing EPATHID
48/29	94/32	EBGP	-	2 5	4038336721
48/29	94/32	EBGP	-	2 3 5	1044010488
48/29	94/32	EBGP	-	2 4 3 5	3884942939

Table 2: Part of forwarding table at EXIT ROUTER 1@AS1

Dst	NextHop	Learnt From	EPATHID	AS-PATH	Outgoing EPATHID
48/29	22/32	IBGP	1044010488	2 3 5	1044010488
48/29	10/32	IBGP	4038336721	2 5	4038336721
48/29	18/32	IBGP	3884942939	2 4 3 5	3884942939

Table 3: Part of forwarding table at ENTRY ROUTER 1@AS2

Dst	NextHop	Learnt From	EPATHID	AS-PATH	Outgoing EPATHID
48/29	1/32	IBGP	1044010488	2 3 5	1044010488
48/29	85/32	EBGP	4038336721	5	0
48/29	5/32	IBGP	3884942939	2 4 3 5	3884942939

Table 4: Part of forwarding table at Router 4@AS2

3.3 Explicit Exit Forwarding in BGP

A simplifying assumption made in previous section is that the entry ASBR is directly connected to the exit ASBR. If not, the packet needs to be sent explicitly to the exit ASBR. In this section, we consider the problem of explicitly routing a packet from either an EBGP router or an IBGP router to an exit ASBR. One simple way of accomplishing this objective is through IP-in-IP tunneling, or using the loose-source-routing IP option. We propose to use an alternative technique which bears similarity to the MPLS label stacking feature that also achieves the same objective.

In particular, we propose a 32-bit "address stack" field in the routing header. The EBGP or IBGP router that decides an explicit exit for a destination prefix will simply "push" the destination IP address into the address stack field and replace it with the exit ASBR's IP address. The IP checksum is also updated appropriately. This is equivalent to using the hash of the exit ASBR in the packet. Any other upgraded IBGP router on the path will observe that the destination address has already been stacked. Non-upgraded IGP or IBGP routers will merely see the packet as destined to the exit ASBR and forward the packet normally. When the packet reaches the Exit ASBR (assumed to be upgraded), it will observe the destination address on the stack, and simply pop it out back into the IP destination address field (and adjust the IP checksum), before performing its EPATHID processing as described in the earlier section. This address stacking procedure operates in the fast processing path at all routers (both upgraded and non-upgraded) unlike tunneling and loose-source-routing. Moreover, it allows flexibility for only a subset of routers to be upgraded to support such explicit exit choice.

For example, in Figure 3, if multi-AS-path were not available, but the IBGP router (say router 1 in AS2) gets a packet that originates in AS2 and is destined to AS5, i.e. to 0.0.0.48/29. It can choose one of three exit routers (router 2, router 3 or router 4) for the packet, thus implying the choice of an AS-path for each case (2-3-5, 2-4-3-5 or 2-5 respectively). The packet can then be effectively tunneled to the exit router using the address stacking procedure described above (even if exit router were not directly connected to router 1). Note, however, that for incoming packets from another AS, only the default exit should be chosen (unless an AS-path is specified through the EPATHID), else the packet may be sent in loops.

4 Control Plane Issues in BGP

Due to space reasons, this section only examines the control plane issues for the inter-domain case. Briefly, for the intra-domain case, we have shown the existence of polynomial complexity algorithms to compute valid subsets of all possible multi-paths assuming partial upgrade conditions. For example, nodes could compute k-shortest paths, and have a second polynomial phase to validate the existence of forwarding for each of these paths.

For the inter-domain case, we have two sub-cases.

The first sub-case is the case of setting up explicit exit ASBRs for chosen destination prefixes (even if EPATHID is not supported). This capability can be achieved completely within a single AS using partial router upgrades, without any expectation from other AS'es. This explicit exit capability would allow ISPs to do fine-grained outbound load-balancing for traffic generated within their own AS'es. The second sub-case is to allow explicit AS-path choice by extending the BGP interactions between AS'es.

4.1 Explicit Exit Routing: Control Plane Issues

To enable explicit exit routing, all we need the upgrade of selected IBGP and EBGP (and corresponding exit ASBR) routers that participate in the explicit exit process. *Only the set of upgraded IBGP and EBGP routers need to synchronize on a subset of exits for a selected set of destination prefixes*. All BGP routers (upgraded or otherwise) participate in the usual BGP process, i.e., synchronize on a *default* policy route (and exit ABSR) to *every* destination prefix. The upgraded IBGP routers then can autonomously choose a subset of the exits available for a destination prefix and install these entries in the forwarding table. As specified earlier in the prior section, these explicit exits may not be chosen for packets originating in other AS'es and not carrying an explicitly initialized EPATHID field. For all these packets, the default exit must be chosen. Therefore, the upgraded IBGP router must always install the default exit ASBR in its forwarding table. The filtering at EBGP routers (to advertise the availability of exits to particular prefixes) or at IBGP routers (to install exits to particular prefixes) is an autonomous local policy matter. Observe that once the upgraded routers synchronize on the available exits, any traffic mapping decision is done autonomously at the IBGP router, possibly on a packet-by-packet or a flow-by-flow basis. Moreover, no new control traffic is required for making TE changes as long as the underlying exits and implied AS-paths are stable (unlike the use of LOCAL_PREF for outbound load balancing that results in IBGP control traffic).

4.2 Explicit AS-Path Routing: Re-advertisement and Synchronization

BGP is a path-vector routing protocol, and hence extra control traffic is needed to convey the existence of multiple AS-paths between neighboring AS'es. Given the scalability and instability issues with adding control traffic, ISPs may choose to only advertise a small set of multiple AS-paths only for a small subset of destination prefixes. This advertisement will be fruitless unless the neighbor AS is upgraded to take advantage of the multi-AS path feature. Moreover, if neighbor AS'es do not relay (re-advertise) at least a *subset* of the multi-AS-paths available from an AS, remote ASs will not be able to take advantage of such multi-AS-paths. By this, we mean that EBGP routers of the neighbor AS must store a subset of the multiple AS-paths to a prefix in their Routing Information Bases (RIBs), and re-advertise them, even though they need not support multi-path forwarding entries in their Forwarding Information Bases (FIBs). Another issue is that, *within a multi-AS-path capable AS*, at least the entry and exit ASBRs need to be upgraded and synchronized on the multiple AS-paths available through the AS before such re-advertisements are made to other AS'es. Observe that other IBGP routers within this AS need not be aware (or upgraded) about the multiple AS-paths to chosen destination prefixes. We believe that due to these issues above, the explicit AS-path selection model is significantly more complex and requires more coordination between AS'es compared to the explicit-exit-ASBR model that can be implemented within a single AS.

5 Related Work

In the area of intra-domain TE, most work focuses on optimizing OSPF by either managing link metrics [13, 11], using equal-cost multi-path (ECMP) or extending intra-domain routing algorithms for multi-path support [9, 12, 4]. Fortz and Thorup [11] use local-search algorithms, and optimize OSPF by changing weights of only few links. Narvaez et al [9] and Vutukury et al [12] propose simple multi-path algorithms that can operate in partially upgraded DV or LS environments, but do not compute all possible paths. Chen et al[4] propose an interesting framework for multi-path forwarding and propose multi-path extensions to

LS and DV routing. MPLS-TE [1, 2] offers signaled explicit label switched paths (LSPs) which can be set up using an arbitrary control algorithm. These authors either require signaling for path setup, full upgrades to networks, or offer a limited set of multipaths. Our building blocks do not have any of these constraints.
In the inter-domain area, the IRTF is considering requirements documents for a future inter-domain protocol, and the traffic engineering problem figures in key drafts [5]. Prior inter-domain TE work includes in-bound/out-bound load-balancing between adjacent AS's using BGP attributes (e.g. MED, LOCAL_PREF, stuffed AS-PATHs)[5], provider-selection, or map-distribution based approaches (NIMROD) [3]. In contrast to parameter tuning approaches, our building blocks approach the traffic engineering objectives in a direct manner. Unlike Nimrod, we do not require signaling or huge per-packet overhead to encode paths.

6 Conclusions

This paper proposes new building blocks for a limited set of intra-domain and inter-domain traffic engineering goals. The key idea is to capture an intra-domain path, AS-path or an exit route as a 32-bit hash in the packet header. In other words, we propose a new routing header with three 32-bit fields: intra-domain hash, EPATHID, and an address stack field. The hash concept allows explicit "source" directed routing without signaling or high per-packet overhead, while enabling an incremental upgrade strategy for OSPF and BGP. We have discussed how to keep the tradeoffs in terms of state, computation and control traffic complexity manageable. Validation using SSFnet simulation and Linux/Zebra implementation is presented. It is important to stress that we have only investigated the issues at the "building block" level. Future research will focus on how to achieve various TE objectives by composing these building blocks, and investigate how end-to-end applications can leverage the availability of multiple paths. We will also focus on quantifying the tradeoffs and more rigorously evaluating the framework.

References

[1] D. Awduche et al, "Overview and Principles of Internet Traffic Engineering," *IETF Internet Draft draft-ietf-tewg-principles-02.txt*, Work-in-progress, Jan 2002.

[2] D. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communications Magazine*, Vol. 37, No. 12, pp. 42-47, 1999.

[3] I. Castineyra, N. Chiappa, M. Steenstrup, "The Nimrod Routing Architecture," *IETF RFC 1992*, August 1996.

[4] J. Chen, P. Druschel, D.Subramanian, "An Efficient Multipath Forwarding Method," in *INFOCOM'98*, March, 1998.

[5] G.Huston, "Commentary on Inter-Domain Routing in the Internet," *IRTF Routing Research Draft, draft-iab-bgparch-02*, Work-in-progress, Sept 2001.

[6] N. Feamster, J. Borkenhagen, J. Rexford, "Controlling the Impact of BGP Policy Changes on IP Traffic," *Technical Report, AT&T Research Labs*, 2001.

[7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Transactions on Computer Systems*, Vol. 18, No. 3, August 2000, pages 263-297.

[8] J. Lepreau, "The Utah Emulab Network Testbed," http://www.emulab.net/

[9] P. Narvaez, K. Y. Siu, "Efficient Algorithms for Multi-Path Link State Routing," *ISCOM'99*, Kaohsiung, Taiwan, 1999.

[10] J. Stewart III, "BGP-4 Inter-domain routing in the Internet," *Addison Wesley*, 1999.

[11] B. Fortz, M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights, in *Proceedings of the INFOCOM 2000*, pp. 519-528, 2000.

[12] S. Vutukury and J. J. Garcia-Luna-Aceves, "A simple approximation to minimum-delay routing," *Proceedings of ACM SIGCOMM*, pp. 227-238, Boston, MA, September 1999.

[13] Z. Wang, Y. Wang, L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," *INFOCOM'01*, April 2001.