BGP Prefix Independent Convergence (PIC) Technical Report

Published November 2007

Clarence Filsfils, Pradosh Mohapatra, John Bettink, Pranav Dharwadkar, Peter De Vriendt, Yuri Tsier
Cisco Systems
{cf, pmohapat, jbettink, pranavd, pdevrien,ytsier}@cisco.com

Virginie Van Den Schrieck, Olivier Bonaventure, Pierre Francois
Université catholique de Louvain (UCL)
{FirstName.LastName}@uclouvain.be

ABSTRACT

With the explosive growth of the Internet and the growing deployment of layer 3 virtual private networks (L3VPN), the size of the Border Gateway Protocol (BGP) routing table has dramatically increased over the past years (in the 750K range in early 2007 counting Internet and L3VPN routes in a typical major Service Provider). Studies have shown that dynamics of BGP may cause several minutes of packet loss during network failures. This duration increases further as the routing table grows, as the traditional convergence operation scales with the number of prefixes. In this paper, we present an alternate solution to provide BGP convergence and demonstrate that it is possible to limit the traffic loss period to sub-second for any failure occurring within the network of a service provider (SP) or on peering links with redundantly-connected peers. This covers the vast majority (if not all) of business models involving tight BGP convergence requirements. We term this alternate solution as BGP Prefix Independent Convergence (PIC) since it works by triggering an immediate and prefix-independent dataplane rerouting of the BGP destinations via the alternate path at Interior Gateway Protocol (IGP) convergence time. We present experimental results of the convergence behavior based on a benchmark of a commercially available carrier router that supports the BGP PIC solution, and on BGP data provided by a Tier-1 ISP.

1. INTRODUCTION

A typical Service Provider (SP) network deploys two flavors of dynamic routing protocols: IGPs such as IS-IS or OSPF are used for routing within the Autonomous Systems (ASs) while BGP [25] is the de facto protocol for global Internet routing between ASs. In addition to the inter-domain routing, the pervasive deployment and flexibility of BGP has also led to its usage as a general purpose transport infrastructure [2, 13, 26]. Many new types of network layer reachability information have been added to BGP [2] for implementing a wide variety of features and applications. Examples include IPv6, L3VPNs [26] and different flavors of Layer 2 VPNs [13] among others. The growth of the Internet and these applications has contributed to an explosion in the BGP routing table and this trend is expected to continue [20].

Several studies have shown that intradomain or core [34, 19] and interdomain or edge links [3, 22] fail frequently. Analysis of the packet loss [16, 33] and routing messages exchanged by routers has also shown that BGP convergence resulting from these events lasts too long [5, 33, 15].

The main reason for this slow convergence comes from the classical dataplane Forwarding Information Base (FIB) organization of router architectures. In such architectures, the routing information on which packet processing engines (PPE) base the forwarding of packets is reduced to its simplest form, due to historical lookup performance issues. These design decisions sacrifice the responsiveness of routers during a convergence for the sake of simplicity and performance, but cannot deal with tight convergence requirements as the size of the BGP routing table keeps increasing.

In this paper, we first characterize and highlight the suboptimality of classical dataplane FIB organizations. Then, we propose and evaluate a hierarchical dataplane FIB organization that enables BGP dataplane convergence upon core or edge failures whose duration does not depend on the number of affected prefixes. This new FIB organization is the main focus of the paper and its core contribution. We demonstrate the realism of such a proposal by evaluating its implementation in a commercially available high-end carrier router.

Our convergence solution for edge failures relies on the availability of alternate paths to BGP destinations at the border routers. Thanks to peering establishment habits, alternate paths are widely available at the border of an AS. Unfortunately, this path diversity is not well distributed among the routers of an AS due to the network design of iBGP routers with route reflectors (RR). We review several solutions for the path diversity problem and how they enable routers to learn an alternate path for each BGP destination. By using these alternate paths, we eliminate the very slow per-prefix control plane path exploration that was required.

In summary, this paper proposes incrementally deployable forwarding and routing techniques to allow BGP routers in a large AS to recover within less than a second from the loss of connectivity experienced by data packets destined to a BGP destination, upon any possible modification to their path. We call this duration "BGP data-plane convergence time". Our

solution reroutes packets in the dataplane in an amount of time which no longer scales with the BGP table size. We thus call this proposal BGP Prefix Independent Convergence (BGP PIC).

This paper is organized as follows. We first present some requirements and definitions in section 2. Section 3 describes the proposed hierarchical organization of the FIB that allows for significant gains in scaling and robustness and is the basis for our PIC solution. Section 4 analyzes and characterizes the application of PIC to core failures while section 5 focuses on edge failures. We describe various schemes for the routers to learn alternate paths for the L3VPN and Internet scenarios in section 6. Section 7 discusses the prefix-dependent convergence. It forms a basis for comparing the normal convergence operation with the proposed PIC solution and also describes its seamless integration with PIC. We defer our discussion of related work until section 8 in order to have the necessary context. The paper concludes with a summary in section 9.

2. TERMINOLOGY AND REQUIREMENTS

2.1 Terminology

Each routing protocol maintains a local table of their best paths to each of their known destinations. The routing information base (RIB) is the routing table of the router. It contains the best paths, across all routing protocols, to each known destination. The forwarding information base (FIB) is a summary of the RIB which only contains the information necessary to forward the packets. The routing processor (RP) is an entity of a router that runs the control plane protocols and other necessary infrastructure. A line card (LC) is an entity of a router that contains both a processor running software and the necessary hardware to forward packets. While the routing protocols and the RIB are supported by processes running on the central routing processor (in operating system context), the FIB table is maintained by the processor on each line card. The FIB process on each line card processor receives incremental modifications from the RIB process and uses them to update the software (SW) FIB and the hardware (HW) FIB tables on the LC. The HW FIB table resides in the packet processing engines (PPE). We use the term "control plane" to indicate all the processes that manage the data that is used for forwarding (i.e. all the processes residing on the RP and also the SW FIB process on the LC CPU). The term "data plane" is used to denote the HW FIB engine that forwards packets based on information in the HW FIB table. The control plane modifies the path used by packets by updating the HW FIB Table. The amount of information and the organization of this information does change significantly between the RIB, the SW FIB and the HW FIB.

We interchangeably use the terms destination, route, and prefix to mean an IP prefix or a BGP NLRI [25]. We further define a path as the representation of a data structure that implementations maintain to refer to a prefix advertised by a particular routing protocol neighbor or next-hop. Thus in general, a prefix has multiple paths, each path identifying a different next-hop from whom the prefix is learned.

We also use the notion of recursion to refer to the parent/child dependency between two prefixes. For example, if the RIB contains a BGP path for route "`Z`" whose BGP nexthop is `X2` and the RIB contains an IGP path for route "`X2`" whose outgoing interface is "`E1`", then we say that the RIB route `X2` is the parent of the route `Z`. We also say that `Z` depends on `X2`. We also characterize this routing structure as "hierarchical". We define as non-recursive a route which does not depend on any other route, i.e. it directly points to an outgoing interface (e.g. an IGP path, a locally connected path) while a recursive route is a route which depends on another route to be resolved (e.g. BGP route depending on an IGP route). This notion of recursion exists in RIB, SW FIB and HW FIB. However, it may be implemented in very different ways. These differences are central to the BGP PIC architecture. The RIB database always maintains the hierarchical dependency between the BGP paths and their parent IGP paths. Finally, we use the term "adjacency" to indicate the data structure which allows the forwarding of a packet to a connected next-hop. On an Ethernet interface, this is the MAC header to append to the IP packet together with the correct destination MAC address for the chosen layer-3 next-hop.

Figure 1 shows the typical reference environment that we consider in this paper.

We classify BGP path modification events into two types: **core** versus **edge**. We define core failure a failure of a node or link within AS X excluding the edge nodes (e.g. link (`X1`, `X4`) or node `X5`). We define edge link failure a failure of a BGP peering link ((`X2`, `Y2`) or (`X3`, `Y3`)) and edge node failure the failure of a peering node (`X2` or `X3`).

2.2 Requirements

Avoiding packet losses after link failures is a key requirement in large SP networks that need to provide stringent SLAs to their customers. Such SLAs are offered for various types of services. In this document, we focus on the two most common services: classical **IPv4 Internet access** and **BGP/MPLS L3VPNs**.

From a control plane viewpoint, a network providing Internet access can be organized in different ways [24]. The first organization, called Pervasive BGP in [24] consists in running BGP on all routers. This approach has been preferred in many networks that provide Internet access. However, it suffers from two important drawbacks. First it forces all routers to maintain a large FIB containing all BGP prefixes. Second, designing a correct iBGP organization is difficult [11] and may lead to deflections, routing loops, and forwarding loops [1]. Another approach leverages encapsulation from Border Router to Border Router to steer packets through the backbone without requiring BGP on the core

routers. Such encapsulations (MPLS, L2TPv3, GRE) are required to provide L3VPN services and can be performed at line rate by current routers.

Redundancy is a key characteristic of current SP networks for two reasons. First, high availability requires the avoidance of single points of failure. Second, the higher growth of traffic demands compared to interface speed has lead to deploying multiple links between Service Providers. This has been confirmed with measurements in [8].

In this paper, we focus on the availability of a path from X1 of SP X to a destination Z advertised by SP Y. In the example of Figure 1, SP X and SP Y share two peering links, respectively (X2, Y2) and (X3, Y3), and there are several disjoint intra-AS paths from X1 to X2 and X3. While BGP PIC would support any number of such destinations Z, our numerical examples will use 350000 such Z destinations.

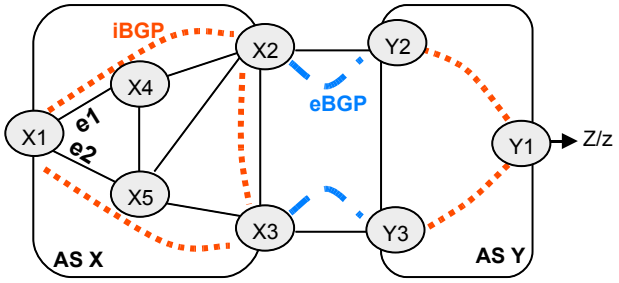

Figure 1: Reference environment

A core event impacts the IGP path to the preferred BGP next-hop. In theory, the speed of such convergence only depends on the IGP capability to detect the failure, flood it, recompute the shortest-path tree and implement the new alternate paths to the existing BGP next-hop. Such IGP convergence upon core event is O(200 msec)[1] with latest commercially-available products [9]. In practice, the loss of connectivity experienced by BGP-destined flows is classically larger by at least an order of magnitude for two reasons. First, the classical dataplane FIB organization consists in a fully resolved translation of the control-plane FIB : the dataplane FIB entry for a BGP destination immediately points to an adjacency [36]. In such a flat dataplane FIB organization, the duration of the loss of connectivity experienced by BGP-destined packets scales with the BGP table size as their dataplane FIB entries need to be updated to reflect the new adjacency as computed by the IGP. Second, the BGP control plane may react to the IGP convergence as the IGP metric to the BGP next-hop is part of the BGP decision process [29]. If this happens, BGP may have to issue modification requests to its related FIB entries. In the past, such modifications were implemented as "delete, add" that could lead to transient packet losses.

[1]Throughout this paper, we use the O(200 msec) notation to indicate a time that is approximately or below 200 msec.

An edge event leads to the loss of a preferred BGP next-hop. An edge event implies the absolute need to perform an immediate BGP next-hop change while a core event only requires an immediate IGP path change to the existing BGP next-hop. The reaction to an edge event depends on several factors: (1) failure detection, (2) alternate BGP next-hop discovery by BGP control-plane, and (3) installation of these alternate BGP nexthops in the dataplane FIB.

An edge event is locally detected in O(10 msec) by interface failure detection (e.g. thanks to SONET/SDH link service, Loss-of-Signal for back-to-back Ethernet links or with BFD) [9]. It is detected by the other nodes in O(200 msec) thanks to the IGP convergence. Upon a Border Router (BR) failure, the IGP neighbors of the node trigger an IGP convergence which leads all the routers in the network to delete the FIB entry to this BR [9]. In the remainder of this text, we will refer to IGP convergence detection upon edge events as it covers all cases.

The remaining two factors (alternate nexthop discovery and their per-prefix installation), are responsible for the slow convergence upon edge failures. The objective of this paper is to remove these factors from the data-plane convergence time.

Based on discussions with many network operators, the common requirement for BGP convergence upon core and edge failure is O(1000msec). The most stringent requirement is O(200msec).

3. FIB ARCHITECTURE

Historically, the FIB databases were "flattened": when translating the RIB content into the FIB content, the recursion is fully resolved such that any FIB entry is immediately linked to its outgoing adjacency [36]. In a flattened FIB database, all the entries are non-recursive.

In our previous example, this means that the FIB entry to Z points to the adjacency to X4 without any dependency on the FIB entry to X2. Upon packet reception, the destination address lookup matches the Z entry and the outgoing interface is immediately found. This organization was privileged in the past as it requires fewer memory accesses per packet lookup.

3.1 Hierarchical FIB database

In a hierarchical FIB architecture, the parent-child relationships are kept inside the FIB. There are thus recursive FIB entries which are children of parent non-recursive entries.

In our previous example, this means that the FIB entry to Z holds a pointer to the memory location of the FIB entry to X2. The FIB entry for X2 itself points to the adjacency for X4. This introduces a level of indirection. Upon packet reception, the destination address look up matches the Z entry which in turn points to X2's entry to finally find the outgoing interface. This organization trades off more memory accesses per packet look-up for better convergence, robustness

and scaling, as described later.

3.2 Generalization of the FIB hierarchy

We further generalize the hierarchical FIB structure by introducing shared **BGP Path List** and the notion of load balancing FIB entries. Equal-cost multipath (ECMP) is possible both at the IGP and at the BGP level. This ECMP decision is supported by load-balancing FIB entries.

For example, if the best IGP path to X2 is either via interfaces E1 or E2, then the FIB entry representing X2 would be made of two paths (E1, E2). At packet switching time, the lookup engine finds the FIB entry for X2, hashes on the flow parameters of the packet to derive a random but flow-deterministic index into either the first or the second path.

A BGP Path List (BGP PL) is defined as a set of best BGP next-hops. If the BGP router is configured with a unipath BGP policy, a path-list only contains one single BGP nexthop. With multipath BGP policy, a path-list may contain more than one BGP nexthop. A BGP PL is shared. All the BGP routes which share the same best BGP nexthops do share the same BGP PL.

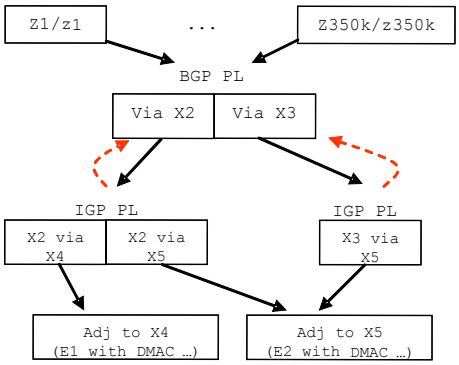

Figure 2: Generalized Hierarchical FIB

Figure 2 describes X1's hierarchical FIB in the scenario depicted by figure 1 assuming that X1 implements a BGP multipath policy for the 350000 routes learned from AS Y and that the IGP computes two equal-cost shortest paths to X2 (with respective outgoing interfaces E1 and E2) and one single equal-cost shortest path to X3 via outgoing interface E2.

X1's FIB is composed of 350000 terminal leaves. They all point to a shared FIB load balancing structure which represents the BGP PL (X2 or X3). This structure, in turn, points to two FIB structures, called IGP Path Lists (IGP PL), respectively representing the reachability to X2 and X3. The reachability to X2 is represented by a two-entry IGP Path-List while the reachability to X3 is represented by a one-entry IGP Path-List. Finally each entry of each IGP Path-List points to an adjacency.

A generalized hierarchical FIB organization requires several memory accesses per packet processing: one to find the longest-match terminal FIB leaf for the packet's destination address, one to find the BGP Path-List, one to find the IGP Path-List and finally one to find the adjacency. The exact algorithm used for the longest-match lookup is irrelevant to the discussion in this paper.

A hardware implementation of such a FIB organization for route lookups faces a fundamental challenge of multiple memory lookups per packet without introducing any compromise on the packet processing rate. However, it is possible to achieve the required processing rate when implemented with the current memory and ASIC technologies. Indeed, the commercially available carrier router that was used for our benchmark experiments supports several millions of FIB entries at 75 million packets per second processing rate.

3.3 Shared BGP Path-List and MPLS

In BGP/MPLS VPN [26] application, each BGP speaker allocates an MPLS label per prefix and includes it in the prefix advertisement messages. In our FIB organization, these labels are stored in the FIB terminal leaves to allow for BGP Path-List sharing. This is significant as it allows for prefix-independent convergence upon edge failures.

For example, assuming in Fig1 that VPN route Z1 (Z2) is learned via X2 with label $L12$ ($L22$) and via X3 with label $L13$ ($L23$), then a label array $(L12, L13)$ is stored in the terminal FIB leaf for Z1 and a label array $(L22, L23)$ is stored in the terminal FIB leaf for Z2.

At packet forwarding time, a packet destined to Z1 hits the FIB leaf to Z1 and the packet processing engine (PPE) stores a pointer to the label array $(L12, L13)$ in memory. The PPE follows the indirection to the BGP PL and picks one of the two BGP next-hops based on hashing. Each BGP nexthop contains an index called a path index. The PPE retrieves this path index and uses it to index into the label array to pick up the correct label. The same process applied on a packet destined to Z2 would result in the correct per-prefix per-path label despite the sharing of the BGP PL between Z1 and Z2 because the pointer to Z2's labels (L22, L23) would be stored in memory and the same path indices would work over Z2's label set as well.

Figure 3 illustrates this example as well as the support of LDP labels. Each path of each IGP Path-List points to a LabelInfo structure. The LabelInfo structure for Path "X2 via X4" holds the LDP label advertised by X4 for reachability to X2.

3.4 Backup BGP nexthop and BGP Path-List

We further generalize the BGP PL definition to contain two sets of BGP nexthops: an ECMP set of primary BGP nexthops and an ECMP set of backup BGP next-hops. The primary set follows the classical BGP decision process. If a unipath policy is configured, BGP will select a single preferred BGP next-hop based on the BGP decision process. If a multipath policy is configured, multiple BGP next-hops will form the primary set, again based on the conventional

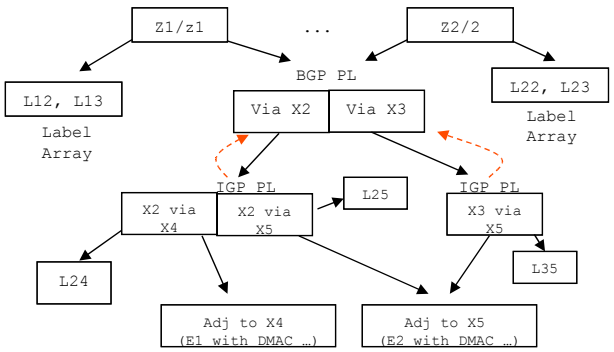

Figure 3: Hierarchical Generalized FIB with MPLS

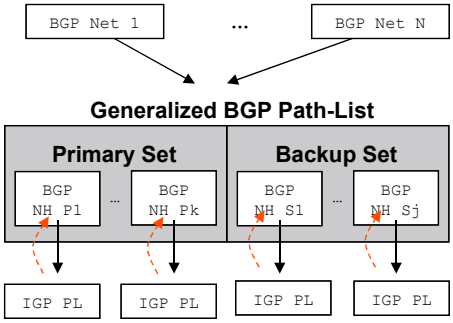

Figure 4: Generalized BGP Path-List

BGP decision process. The novelty consists in enabling BGP to compute backup next-hops and communicate them to RIB that will download the backup set to FIB.

While more refined algorithms would likely be used to select the backup set, a simple proposal consists in running the BGP decision process once to elect the primary set, then to exclude these paths from the table and running the decision process a second time to elect the backup set. Such second run could be performed asynchronously and at a lower priority to not delay the computation of the conventional primary set. In practice, some paths may fail at the same time, e.g. because they use a peering link that relies on the same optical device or end at the same router. In this case, the paths that could fail at the same time as those of the primary set should be removed from consideration for the backup selection. This can be achieved by first identifying those paths [14] and then configuring the BGP routers that learn those paths over eBGP sessions to mark them with one BGP extended community [27] per set of shared resources.

From a FIB organization viewpoint, such a generalized BGP PL now holds two sets of pointers to parent IGP PLs. At packet lookup time, as long as the primary set is not empty, the hashing decision is only done on the primary set. Once empty, the hashing decision is performed within the secondary set.

Figure 4 illustrates this organization with a BGP PL with K primary BGP nexthops and J backup nexthops. In practice, multipath BGP policies are rarely used by ISPs and the most frequent generalized BGP Path-List would have one primary BGP nexthop and one backup BGP next-hop.

3.5 Linked-List of BGP Path-List

The control-plane component of our proposed FIB architecture maintains linked lists of BGP PLs per dependent IGP PL. This is illustrated with red dotted arrows in figures 2 and 3. These linked lists are not present in the HW FIB.

These linked lists are essential for BGP PIC Edge. Indeed, upon IGP convergence, SW FIB will have to delete a parent IGP PL. Before doing so, the SW FIB will walk the list of depending BGP PLs and will modify them to only use valid BGP next-hops. Along this walk, SW FIB will update the BGP PLs in HW FIB and hence the packets will be rerouted in an amount of time which only depends on the IGP convergence and on the number of impacted BGP PLs, but not on the number of BGP prefixes.

BGP PIC Edge is an entirely automated behavior and supports any type of policy. When the FIB receives a request to add a new entry from the RIB, the FIB checks whether the related BGP PL already exists (it must have the same exact set of primary and secondary BGP nexthops). If not, it is created and the new entry is linked to it. If it exists, the new FIB entry is linked to the existing BGP PL. There is thus no dependency on how the BGP routes are learned and what policies are applied on peering links.

Specifically, figure 1 is a simplistic example to illustrate the functionality and one should not assume that BGP PIC Edge require parallel peering links with the same exact policy. The BGP PIC Edge mechanism automatically groups all the BGP entries sharing the same BGP PL.

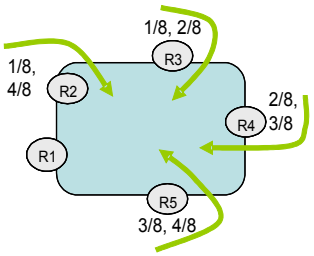

Figure 5: Automatic Mapping to BGP Path-Lists

This is illustrated in a more realistic example in figure 5. Assuming that `R1` knows at least two paths to any destination, that `R1` implements BGP PIC Edge and an hot potato policy, and that the IGP metrics are such that igp_{dist}(`R1`, `R2`)

$< igp_{dist}$(R1, R3) $< igp_{dist}$(R1, R5) $< igp_{dist}$ (R1, R4), we deduce that R1 automatically creates four BGP PLs. The BGP PL for 1.0.0.0/8 contains primary BGP nexthop (R2) and one backup BGP nexthop (R3). No other BGP destination shares this BGP PL. Upon loss of R2, R1 deletes the IGP PL to R2 and walks its list of depending BGP PLs. This list only contains two BGP PLs: {R2, R3} and {R2, R5}. If R3 would also announce 4.0.0.0/8, then 4.0.0.0/8 would share the same BGP PL with 1.0.0.0/8. In such case, upon loss of R2, although 3 BGP routes are impacted, only two BGP PLs need to be modified. We intuitively expect a very good sharing of BGP PLs. We use data from SP networks to confirm this in section 5.

4. BGP PIC UPON CORE FAILURE

This section details and characterizes the convergence speed, scaling and robustness benefits of the hierarchical FIB organization upon core failures.

4.1 Convergence and Robustness

A hierarchical FIB design supports for BGP Prefix Independent Convergence upon Core failure: all the BGP destinations immediately benefit from an IGP convergence as the dataplane FIB entries representing the BGP destinations point to the FIB entries which represent the IGP routes to their BGP nexthop. After a core failure, the IGP converges in 0(200msec) [9] and the BGP-destined packets are immediately rerouted on the new IGP path.

To the contrary, a flattened dataplane FIB design significantly delays the dataplane convergence of BGP-destined traffic. Indeed, each dataplane FIB entry representing a BGP destination contains the fully resolved pointer to an adjacency (the BGP PLs and the IGP PLs present in the SW FIB organization are flattened when the HW FIB table is created and maintained) and hence all these hardware pointers need to be updated when the adjacency of their parent IGP path changes. This modification scales with the BGP table size and must occur upon any core event (link or node down or up, metric change). At 10μsec per modification, this may lengthen the loss of connectivity up to 3.5 seconds instead of O(200 msec). Besides this significant convergence benefit, our BGP PIC Core solution provides for much better robustness.

Assume in figure 1 that X1 prefers X2 to reach the 350000 Z destinations advertised by AS Y and that suddenly a core link flaps and impacts the best IGP path from X1 to X2.

Upon the down flap transition, the IGP convergence at X1 leads to the modification of the X2 FIB entry (e.g. use interface E2 instead of interface E1). A hierarchical FIB table is fully updated with this single operation while a flattened table requires 350000 additional FIB updates (each FIB entry depending upon the modified FIB entry must be re-resolved and updated with the new interface). Assuming 10μsec per recursion resolution, this flattening task drives the LC CPU to 100% usage during 3.5 seconds.

Let us assume that 100 milliseconds later, the link comes back up. In the hierarchical FIB design, the processors are idle and hence the event is processed immediately with one single FIB modification to restore the shortest IGP path to X2 (the 350000 BGP routes immediately use this new best IGP path). In the Flattened FIB case, this restoration cannot start before 3.4 seconds and will require another 3.5 seconds of continuous 100%-CPU usage. In total, the flattened FIB architecture would have lead to 7 seconds of 100%-usage CPU, an average loss of traffic of 1.75 second and a delayed use of the best IGP path (hence potential capacity congestion) for up to 7 seconds.

4.2 Requirement

The BGP PIC Core solution does not require BR-to-BR encapsulation. It does not require the availability of disjoint BGP Paths in the BGP table, RIB, or FIB. The BGP PIC Core solution does not require BGP PL in the dataplane. It only requires the presence of the IGP PL in the dataplane FIB organization. BGP PIC core does not increase the feasibility or magnitude of such loops and does not create the possibility of other loops.

4.3 Characterization

The lab setup reflects the topology of figure 1. We use routers with 10 Gbps Ethernet interfaces. Link X1-X4 is instrumented to fail the link on demand. This failure is detected by X1 and X4 as a normal failure detection. An ISIS emulator is connected to X4 and inserts an additional topology of 1000 nodes and 5000 prefixes. This corresponds to the size of the IGP in a large Tier-1 ISP [9]. The X1-X4 link failure impacts this complete topology and hence X1's IGP convergence scales with a 1000-node real topology and 5000 prefixes. This is a worst case scenario from the IGP's viewpoint. X1 learns about 350000 BGP routes with 2 paths (via X2 and via X3). X1's BGP policy consists in preferring X2 over X3. We send traffic from a packet generator on the left of X1 towards the BGP destinations advertised by the right AS. 15 streams are scattered evenly through the 350000 BGP destinations. X1, the unit under test (UUT), is a commercially available product. X1 can be loaded with two different softwares with or without BGP PIC Core support. All the other nodes run the BGP PIC Core enabled SW.

Each of the 15 streams transmits one packet every millisecond. In addition, a one million packets per second background stream is sent through the UUT. For each of the 15 test streams, we measure the number of packets lost when the link X1-X4 is failed. This is indicative of the dataplane convergence time for this specific prefix. We keep measuring for 5 minutes to record any loss induced by the later BGP control-plane convergence. Indeed, after the failure, X3 becomes the closest exit for X1 and hence the BGP control-plane convergence process will send 350000 modification requests to FIB.

Figure 6 plots and contrasts the collected results with and

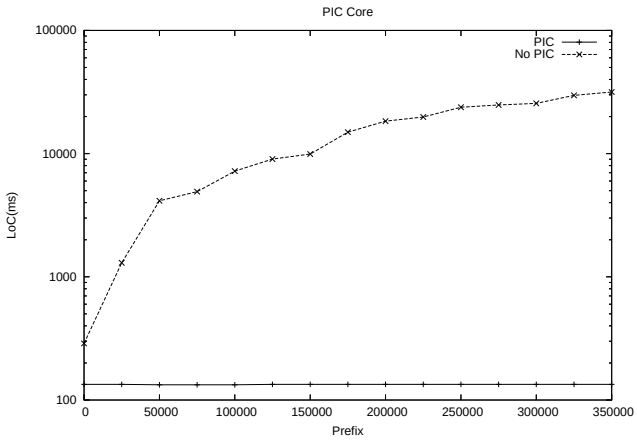

Figure 6: BGP PIC Core

without BGP PIC Core support on X1. The X axis represents the 350000 BGP prefixes. The Y axis reports the measured convergence time in millisecond in logarithmic scale for each BGP prefix.

When the UUT runs the software enabling BGP PIC Core: the IGP convergence leads to a FIB modification of the route to X2 134 msec after the failure. All the FIB entries to the 350000 BGP destinations resolve upon the FIB entry to X2 and hence loss ends after 134 msec. This is confirmed by the measurements: all measured flows across the 350000 BGP destinations show 134 msec packet loss. After the IGP convergence, BGP on X1 detects that X3 is the closest exit and hence will update, one by one, the 350000 BGP-derived FIB entries to resolve via the X3 FIB entry instead of the X2 FIB entry. Our experiments confirm that this modification, although very long, is hitless for the traffic.

When the UUT does not run the SW enabling BGP PIC Core: the FIB entry for X2 is updated 288 msec[2] after the failure. All the flattened FIB entries to the 350000 BGP destinations need to be updated one by one with the new path to X2. This is confirmed by the measurements: there are 30 seconds of difference between the loss reported by the first and last streams. The later BGP-induced modification of the 350000 destinations from X2 to the now shorter-exit X3 is handled by the same code as in the BGP-PIC-enabled SW and is known to be lossless. We confirmed this in the lab by artificially delaying the BGP control plane convergence by 60 seconds. The measured loss of packets all happened during the first 30 seconds which confirm that the later BGP convergence was handled by the FIB in a lossless manner.

5. BGP PIC UPON EDGE FAILURE

[2]This SW does not contain all the IGP convergence optimizations available in the BGP-PIC-Core-enabled SW and hence the IGP convergence is slightly slower. This difference has no consequence on the analysis conducted in this paper.

This section details and characterizes the convergence speed and scaling benefits of BGP PIC Edge solution.

5.1 Convergence

To anticipate an edge failure, the BGP control plane ensures that any BR knows at least two paths to any BGP destination (see section 6). It computes a primary set of BGP nexthops and a set of secondary BGP nexthops. These two sets are communicated to the FIB. The FIB SW automatically creates and shares BGP PL between these BGP destinations. Each IGP PL contains a linked list of BGP PLs that depend on it.

Upon edge failure, the IGP convergence concludes with a deletion of an IGP PL. Then the SW FIB walks the list of dependent BGP PLs and disables the failed path. The related BGP PLs in the HW FIB table are updated. Two different cases need to be discussed. If the router was using a multipath policy (e.g. in figure 1 X1's FIB contains paths via both X2 and X3), then the update of the BGP PL invalidates one path but the others remain active and can still be used to forward packets. If the router was using a unipath policy (e.g. in figure 1 X1's primary set contains only the BGP PL via X2), then the update of the BGP PL causes the primary set to become empty and the backup set is now used to forward the packets (e.g. X3 in figure 1). BGP-destined packets are rerouted via alternate viable BGP nexthops. The loss of dataplane connectivity experienced by these packets scales with the IGP convergence time and the number of impacted BGP PLs. Thanks to the hierarchical FIB, it does not scale with the number of impacted BGP prefixes. Assuming an O(200 msec) IGP convergence [9] and up to approximately 100 impacted BGP PLs per IGP PL, O(200 msec) BGP dataplane convergence upon any edge failure can be achieved.

5.2 Requirement

On top of the BGP PIC Core requirement, BGP PIC Edge requires: (1) BGP PL support with primary and backup sets of BGP nexthops, (2) each IGP PL must maintain a linked list of dependent BGP PLs, (3) BR-to-BR encapsulation, (4) an iBGP design allowing routers to learn the disjoint BGP paths available at the boundary of the AS.

Since the BGP FIB is updated before the BGP control-plane convergence that will follow the edge failure, we must ensure that this does not cause forwarding loops. In the multipath scenario, the activation of BGP PIC edge cannot create any loop as the FIB modification simply consists in discarding no-longer-valid paths. In the unipath scenario, the utilization of BR-to-BR encapsulation ensures that no transient loops will occur. Consider an ingress BR (e.g. X1 in figure 1) that updates its hierarchical FIB after the failure of a primary egress BR (e.g. X2 in figure 1). After the update, the interdomain packets affected by the failure are sent encapsulated to an alternate egress BR (e.g. X3 in figure 1). Since these packets are encapsulated, core routers (e.g. X4 in figure 1) do not use their BGP FIB to forward them. Thus,

they cannot participate in a BGP-PIC induced loop. Furthermore, the BR-to-BR encapsulation (MPLS, IP) is deployed in such a way that the encapsulated packet contains a label that allows the egress router to forward the received packets over a given external peering link without consulting its BGP FIB. Thus, the egress BR cannot create a loop by forwarding the received encapsulated packets to another BR of its AS.

It is interesting to highlight the analogy between the generalized hierarchical dataplane FIB design and the dataplane support for BR-to-BR encapsulation. Historically both were not supported due to technology and cost limitations and this lead to significant issues throughout any BGP-based deployment. Flattened FIB lead to slower convergence, worse scaling and robustness. Pervasive BGP deployment lead to BGP-induced loops [28], worse scaling and robustness (core routers need to run BGP) and less functionality (e.g. no ability for traffic engineered exits). Most of the commercially-available SP products in 2008 are able to support BR-to-BR encapsulation (MPLS, IP) at line rate. The commercially-available carrier router benchmarked in this paper supports the generalized hierarchical FIB with millions of entries at 75Mpps.

5.3 Characterization

As explained above, one of the scaling factors of BGP PIC edge is the number of BGP PLs that need to be maintained by the routers and updated upon an edge failure. To evaluate this factor, we analyzed the BGP routing tables of five route reflectors from a large Tier-1 ISP. These route reflectors receive a much larger number of paths than a normal border router. Their RIB is thus much larger than the one of border routers.

The number of BGP PLs does not directly depend on the number of BGP nexthops used, but on the number of paths per destination from the RIB that are placed in the hierarchical FIB. For fast data-plane convergence, a router will be typically configured with a unipath policy that installs in its FIB a primary set containing its best path and a backup set containing one alternate path. This is the most common scenario from a deployment viewpoint. In theory, in a network containing n BGP nexthops, there could be up to $n \times (n-1)$ BGP PLs in such a scenario. However, in practice only a small fraction of all pairs of BGP nexthop will appear as a BGP PL.

For each of the five RRs, the total number of BGP PL is between 423 and 645. We performed the same analysis based on one BGP router of the European research network GEANT. The number of BGP PL for this router is only 54.

Upon the loss of a specific IGP PL, BGP PIC Edge walks the linked list of its associated BGP PL. The scaling factor in case of edge Failure is the number of these associated BGP PL. For the Tier-1 ISP, we computed the number of BGP PL impacted by the failure of each BGP nexthop. The number of BGP PL impacted by each edge failure is usually much smaller than 10, with the percentiles 10, 50, 90 and 100 being respectively 0, 2, 5 and 85 impacted BGP PL. In the case of GEANT, most of the failures impact 1 or 2 BGP PLs, with a maximum of 5.

For the next sections, we take the worst case, i.e. O(100), to represent the number of BGP PL impacted by a failure.

The experimental setup of figure 1 is reused with the following modifications: (1) `X1`'s BGP policy is changed to select both `X2` and `X3` (BGP multipath policy) as the nexthops for all BGP destinations, (2) `X2` is failed instead of a core node within `X`'s network, (3) we increase the number of measurement flows to match the increase of the number of BGP prefixes, (4) the source and destination addresses of these flows are chosen to ensure that, prior to the failure of `X2`, these flows were forwarded via `X2` instead of `X3`, (5) LDP is configured on all devices to use BR-to-BR encapsulation.

Figure 7 plots and contrasts the collected results with and without BGP PIC Edge support on `X1` and for 250000 and 500000 BGP prefixes advertised by the right AS. The X axis represents the BGP prefixes. The Y axis reports the measured loss duration (convergence) in millisecond in log scale for each BGP prefix.

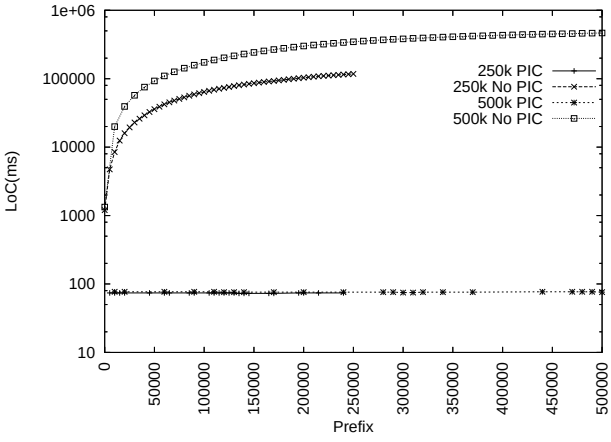

Figure 7: BGP PIC Edge

When the UUT runs the software enabling BGP PIC Edge: the FIB entry for `X2` is deleted 74 (77) msec after the failure with 250000 (500000) BGP destinations. This time only depends on the IGP convergence speed and is independent of the BGP table size[3]. As soon as the FIB entry to `X2` is deleted, `X1`'s FIB SW walks the list of dependent BGP PLs and modifies them to only use the remaining BGP nexthop (`X3`). This operation depends on the number of BGP PLs and not on the number of BGP prefixes. As soon as the BGP PL is modified, all BGP-derived FIB entries resolving

[3]The very small IGP convergence difference between the two occurrences of the same event (2msec) is a benefit of the software and hardware optimization for IGP convergence [9].

through that BGP PL are rerouted losslessly. This is confirmed by the experiment. Irrespective of the number of BGP prefixes advertised (250000 or 500000), all the 24 streams destined to addresses evenly scattered across these prefixes report the same loss of 80 msec. After the IGP convergence, BGP on X1 detects that X2 is now an invalid BGP next hop and updates the best path for all the impacted routes. This very long BGP-induced convergence will modify each FIB entry to point to a BGP PL containing a single BGP next-hop (X3) instead of the modified BGP PL (X2 disabled, X3). While very long, this BGP convergence is lossless as confirmed by the experiment.

When the UUT does not run the software enabling BGP PIC Edge: the FIB entry for X2 is deleted 79 (77) msec after the failure with 250000 (500000) BGP destinations. Without BGP PIC Edge, 50% of the traffic is dropped until the BGP induced convergence occurs (in our case, 100% of the 24 streams since we choose the flows so that they are all forwarded via X2). The BGP convergence starts very quickly thanks to next-hop tracking service of the RIB process. However, it lasts very long as for each impacted BGP prefix, the BGP code must invalidate the path, select a new bestpath, update the RIB and the FIB and generate the BGP update/withdraws and send them to peers. The BGP control-plane convergence takes 118 seconds for 250000 routes in this experiment, or roughly 500 μsec per route.

BGP PIC Edge is a fundamental element of modern router architectures. It offers an entirely automated dataplane protection mechanism which hides the very long control-plane driven BGP convergence when tens of thousands of routes are impacted. In this experiment, a customer of a VPN or Internet service not benefiting from PIC Edge would experience loss of connectivity of 118 seconds instead of 80 msec with PIC Edge support.

We complete our analysis of BGP PIC Edge with a characterization of the duration required to walk and modify one BGP PL.

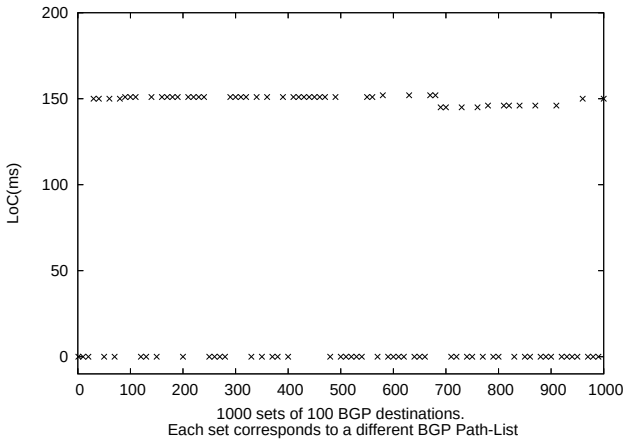

Figure 8: BGP PIC Edge and the number of BGP PLs

We reuse the same testbed with 100000 BGP routes advertised by the right AS. We configure X2 with 1000 loopbacks, each with a different IP address. A policy is applied on X2's outbound iBGP session to X1 to change the BGP next-hop of advertised routes. This policy divides the 100000 routes received from the right AS in 1000 sets. Each set is associated with a different BGP next-hop (one of the 1000 loopbacks of X2). One tenth of these sets are monitored by a dedicated stream. We thus have 100 streams destined to 100 BGP destinations scattered evenly across the 1000 sets of BGP routes. Each set is multipath load balanced between one loopback of X2 and X3. X3 is failed and figure 8 reports the measured loss for each stream. The X axis reports the 1000 sets of 100 BGP destinations advertised by the right AS. The Y axis reports the measured loss for each monitored stream.

With BGP PIC Edge, X1 automatically organizes its hierarchical FIB by creating 1000 different BGP PL. 100 unique BGP prefixes share each such path-list. Each BGP PL contains two entries, thus creating a multipath structure: the first entry is a loopback on X2 and the second entry is X3.

Upon deletion of the FIB entry to X3 (depending only on IGP convergence and occurring 145 msec after X3's failure in this experiment), X1's FIB walks the list of 1000 dependent BGP PLs and modifies them one by one to only use the remaining valid multipath entry (one of the loopbacks of X2).

In this experiment, we did not hand-pick the source and destination addresses. 52 of the 100 streams report a zero loss, indicating that these streams were load-balanced via X2 before the failure of X3. The 48 other streams report a loss duration ranging from 145 to 152 msec. It thus takes 7 msec to back-walk and modify the 1000 BGP PLs impacted upon the deletion of the FIB entry to X3. This confirms that BGP PIC Edge is independent of the number of impacted BGP prefixes and only scales with the number of impacted BGP PLs by an order of magnitude of 7μsec per impacted BGP PL.

6. DISTRIBUTING ALTERNATE BGP PATHS

The FIB organization described in the previous sections is sufficient to allow the BGP routers of an AS to quickly recover from edge failures provided that each BGP router knows at least two paths to reach each destination prefix. In this section, we discuss different techniques that allow BGP routers to learn these paths.

6.1 Best external advertisement

Quite often, service providers employ routing policies that cause a BR to choose a path received over an iBGP session (that of another BR) as the bestpath for a prefix even if it has an eBGP learnt path. Known popularly as active-backup topology, this is done to define one exit or egress point for the prefix in the AS and use the other(s) as backups if the primary link or eBGP peering were to go away. These

policies are equally applicable for both Internet access and MPLS VPN networks. Moreover, in MPLS VPN networks, the topology is usually regulated by the VPN customer by marking the route advertisements with a special community.

The policy, though beneficial, causes the BR to hide to the AS the paths that it learned over its eBGP sessions since it does not advertise any path for such prefixes. To cope with this, some routers have been modified to advertise one externally learned path, called as the best-external path. The best-external behavior causes the BGP selection process to select two paths to every destination: (a) best path that is selected from the complete set of routes known to that destination, (b) best-external path that is selected from the set of routes received from its external peers. BGP advertises to external peers the best path. Instead of withdrawing the best path from its internal peers when it selects an iBGP path as the best path, BGP advertises the best external path to the internal peers.

This feature is an essential component of PIC edge for both Internet access and MPLS VPN scenarios since it allows the availability of alternate paths in the network in the active-backup topology.

6.2 BGP/MPLS VPNs

In a network providing BGP/MPLS VPNs services, providing at least two paths to reach each VPN prefix is easily solved. Indeed, when a Provider Edge (PE) router advertises inside its AS a prefix learned from a Customer Edge (CE) router, it combines the prefix with a route distinguisher (RD) [26]. Important sites are multi-homed with two different CEs attached to different PEs while smaller sites are usually single-homed. By using distinct RDs for each CE of multi-homed sites, a network operator can easily ensure that all prefixes advertised by this site will be learned by all PE routers that serve this VPN. This design technique, called unique RD, is the most common one. Thus all PE routers will know all alternate paths to reach each VPN prefix.

The active-backup topology still needs to be addressed since the BR will not advertise a route learnt from a CE if it chooses an iBGP path as the best path. Thus the BRs need best-external advertisement support. Since the BR attaches a unique RD to the best-external advertisements, the route reflectors transparently reflect all such prefixes, thus making alternate paths available at the other BRs.

In summary, with a unique RD design and best external advertisement, alternate paths can be guaranteed for MPLS VPN networks.

6.3 Internet access

When considering Internet access services, the problem is slightly different. In this section, we first show that alternate paths are available at the boundary of the AS. Second, we briefly describe why the border routers don't always receive the alternate paths and discuss several techniques that allow to distribute these paths to the BR's.

6.3.1 Availability of alternate paths

First, we note that for most prefixes, an AS learns several paths towards each prefix, possibly with different BGP attributes. Indeed, most ISPs are connected via multiple peering links to their peers [8] and having multiple physically disjoint links is often a requirement in peering agreements. Furthermore, most customer networks maintain multiple links to their providers.

To quantify this availability of multiple paths, we analyzed two real ISP networks : GEANT and the Tier-1 ISP considered earlier. For each network, we built a C-BGP model [23] using the network topology, BGP configurations, and BGP routing table dumps. Figure 9 shows the number of paths per prefix for both ASes. To plot the two networks on the same figure, the X-axis shows the cumulative number of prefixes, and the Y-axis the number of paths per prefix. We see that half of the prefixes are reachable via more than 5 paths in both ASes. Our C-BGP model shows that 2,5% (resp. 5%) of the prefixes are reachable via only one single path in GEANT (resp. the Tier-1 ISP). Our model underestimates the path diversity in this case because it was fed with the BGP routes learned by one router (of the iBGP full-mesh) in the case of GEANT and five top-level route reflectors for the Tier-1 ISP. In GEANT, the prefixes reachable via only one path are the prefixes advertised by the GEANT's customers that are attached to a single router. Most of these customer prefixes are also reachable via the commercial providers that peer with GEANT, but these paths are less preferred than the customer paths and thus they are not advertised in GEANT's iBGP full mesh. We also checked manually most of the prefixes for which no alternate was found in the C-BGP model of the Tier-1 ISP. Most of these prefixes were learned from dual-attached peers but the BGP filters on one of these links were configured with a low `local-pref` value. It is very likely that alternate paths could have been found in the ADJ-RIB-INs of the routers attached to those links.

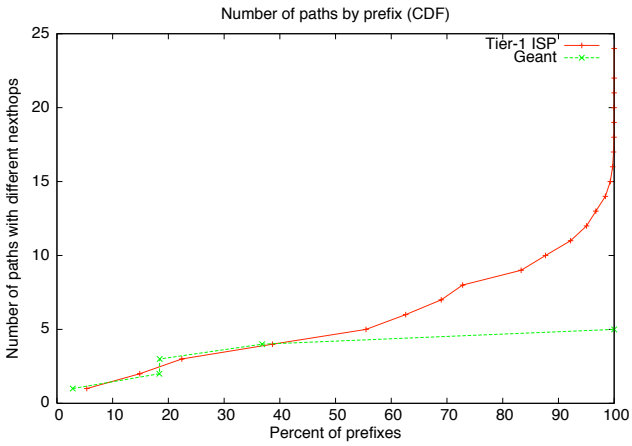

Figure 9: Number of paths learned for each prefix

This analysis shows that path diversity is available at the borders of the network, and can be used by BGP PIC EDGE to quickly recover from edge failures. However, even if the diversity is available at the borders of the AS, this does not mean that all routers will have alternate paths for all prefixes for which diversity exists in the network [30]. By using the C-BGP model of the Tier-1 ISP, we were able to reproduce the distribution of the BGP routes on all BRs. Then, we measured on the RIB of each modeled BR the number of prefixes for which there are at least two different paths. Figure 10 shows on the X-axis the percentage of routers in the Tier-1 ISP while the Y-axis shows the cumulative number of prefixes that have been learned via two or more different paths by the router on the X-axis. The figure shows that 80% of the routers have learned alternate paths for less than 50% of the prefixes. This should be compared with the fact that 95% of the prefixes have been learned via two or more paths by the AS. This lack of diversity is because most routers are clients of two route reflectors and for many prefixes, these two RRs advertise the same best path and hide the alternate path. The routers having diversity above 80% are the top-level Route Reflectors.

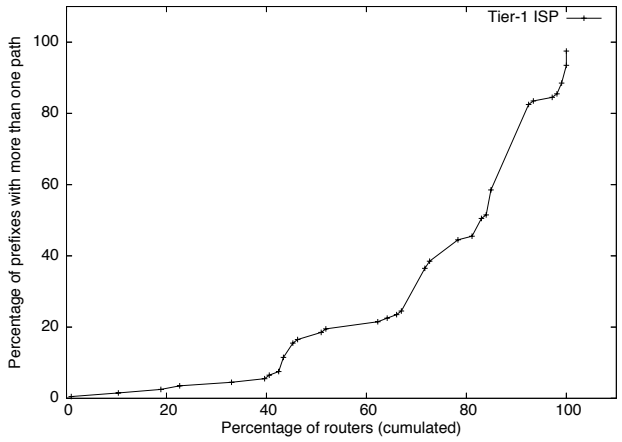

Figure 10: Diversity per router

6.3.2 Solutions to distribute alternate paths

Several solutions have been proposed to distribute more iBGP paths inside an AS. As we discussed in section 6.1, when a router selects a path received from an iBGP session as the best path for a prefix, it does not advertise any path for that prefix. To increase the path diversity in these scenarios, the routers should be modified to advertise the best-external path. This is a base requirement that the alternatives described below rely upon.

The simplest and widely used solution to ensure that all routers learn all those paths is to configure a full mesh of iBGP sessions. This iBGP organization is often used in small networks. For example, both GEANT and ABILENE use a full mesh of iBGP sessions. Of course, a drawback of the iBGP full mesh is that $n \times (n-1)/2$ iBGP sessions are used in a network containing n BRs. Thus, this solution is mainly targeted at small to medium networks.

An alternative is to consider Internet IPv4 routes as belonging to a VPN and configure a distinct RD for this VPN at each BR with eBGP sessions for Internet peering. This solution allows all routers to receive, through the existing iBGP sessions and without changing BGP, all paths towards each destination prefix, even in the presence of route reflectors, as analyzed in section 6.2. This meets our requirement for learning alternate paths, but increases the memory consumption. For example, in the Tier-1 ISP that we considered, the RIB of each router would need to store on average six paths for each prefix.

A different solution that has been discussed within the IETF is to change BGP to allow several paths towards the same prefix to be advertised over one BGP session. This extends the base BGP protocol that allows for only the best path to be advertised. The extension proposed in [12] was initially motivated by load balancing. The extension proposed in [32], known popularly as add-path, was initially targeted at solving the MED oscillation problem and received more support than [12] within IETF. The add-path proposal extends the NLRI format to include a path identifier, thus supporting more than one path to be advertised, without implicitly replacing the first advertisement of the prefix with the next. For the PIC requirement, this extension will allow the RRs to advertise both the best path and one or more alternate paths to the BRs, increasing the path diversity in the network. Despite discussions at the IETF, neither of these proposed extensions have been implemented. First, they present a set of challenges for the implementation. Second, they require a network-wide migration of all BRs and RRs to support the extensions.

An alternate mechanism that is easily deployable is to introduce dedicated route reflectors to advertise the alternate paths. These can be installed in addition to the set of primary RRs in a cluster. They select the most disjoint path from the best path as the alternate path and advertise that to all iBGP peers instead of the best path, thus increasing the path diversity.

In a nutshell, our analysis demonstrates that path diversity is present for Internet routes at the borders of the SP network. We further expound on the solution space for distributing these alternate paths in addition to the best path to each border router, satisfying the requirement for PIC edge.

7. PREFIX-DEPENDENT CONTROL-PLANE CONVERGENCE

In this section, we explain how the BGP control Plane reacts to the failure events and how it reconciles with the BGP-PIC-modified FIB entries present on the LCs.

7.1 Nexthop Tracking

BGP depends on the status of IGP routes in the RIB for

validating its BGP next-hops and for choosing between paths (shortest exit rule in the best path algorithm). When BGP installs prefixes in its routing table, it also registers the corresponding next-hops with the RIB to be notified when the IGP route to those next-hops get modified. Upon notification, BGP reruns best path selection for the depending BGP prefixes and, if required, updates RIB and peers.

7.2 Core Failure

Upon core failure, the IGP modifies its shortest path to a BGP next-hop and updates the RIB accordingly. The RIB update causes two parallel reaction chains. On one end, RIB will send FIB updates to the LC which will cause the data plane switch-over on the alternate path and the end of dataplane connectivity loss for both the IGP-destined traffic and all the BGP-dependent-destined traffic (BGP PIC Core). It is important to highlight that this reaction chain is the fastest and hence the most important. On the other end, RIB notifies BGP which triggers a BGP control-plane convergence operation. This control-plane operation could be extremely long as it is prefix dependent. Indeed, BGP needs to recompute best-path for all the BGP prefixes depending on the modified BGP next-hop. If due to the longer IGP path, another BGP next-hop becomes preferred, then the number of RIB and FIB updates will scale with the number of impacted BGP routes. Based on our characterization of BGP prefix dependent convergence, a rough rule of thumb of 500μsec per (best path, RIB update, BGP update message) leads for example to a 100-second BGP convergence for 200k depending prefixes. This potentially very slow BGP convergence does not need to occur immediately after the failure thanks to the BGP PIC Core behavior.

An implementation can easily enforce that the slow non-urgent BGP convergence be delayed after the IGP convergence (and the enabling of BGP PIC Core) by introducing a delay between the RIB notification of BGP next-hop change and the start of the BGP control-plane convergence. A few seconds should be long enough. More specifically, we note that this delay may and should be inserted when the RIB notification is of “modify” type: i.e. the BGP next-hop is still reachable, it is just using an alternate path, and hence BGP convergence is not urgently required.

Furthermore, the implementation must update the FIB entries atomically: if a FIB entry representing a BGP prefix needs to use BGP next-hop2 instead of BGP next-hop1, the implementation should keep the existing FIB entry for this BGP route and simply modify the pointer "to BGP next-hop1" by a pointer "to BGP next-hop2". It is indeed very important that, if the policy leads to a new BGP next-hop selection (the previous one is no longer the closest after convergence), the modifications sent to the LC FIB are hitless (switching from a valid forwarding entry to a valid forwarding entry should be lossless). Our lab characterization validated this behavior.

7.3 Edge Failure

Upon edge failure, the IGP convergence is extremely simple from a topology viewpoint (maximum efficiency of incremental shortest-path tree computation as we delete a terminal node from the tree) and from a RIB update time (by definition, an edge failure is limited to a few IGP prefixes and hence the RIB update time is negligible). For that reason, the two reaction chains, while still in parallel from a theoretical viewpoint, are serialized in practice. First, the RIB delete triggers a FIB delete to the LC which triggers the modification of the shared dependent BGP PLs (i.e. BGP PIC Edge). Second, the RIB delete triggers BGP control-plane convergence. In this case, the notification is a “delete” and hence the BGP implementation may automatically differentiate it from a modification (core scenario). BGP knows that the IGP convergence is extremely quick and basically done by the time BGP gets hold of the delete information, BGP may start its control-plane convergence immediately. This control-plane convergence will be slow but the end customers will not notice anything as the dataplane convergence already occurred thanks to BGP PIC Edge. Once again, hitless modification of the FIB entries ensure that the control-plane BGP driven convergence does not incur any loss of connectivity. This was validated by our measurements.

8. RELATED WORK

Many papers have studied the convergence of the BGP protocol [17, 10, 18, 4] but few have implemented solutions to improve the dataplane convergence. As Bush et al. have pointed out [5], the dataplane convergence is much more important in operational networks than the control-plane convergence.

Internet measurements have shown the slow control-plane convergence of BGP [17, 18] and more recently their negative impact on the data plane performance [33, 16]. We expect that the solutions proposed in this paper, once widely deployed, will reduce the impact of link failures on the dataplane and may also reduce the BGP churn. Several BGP extensions have been proposed to improve the BGP control-plane convergence, such as Ghost flushing by Bremler et al. [4], RCN by Pei et al. [21] or EPIC proposed by Chandrashekar et al. in [6]. These extensions aim at reducing the BGP control-plane convergence time by tuning its parameters or reducing the amount of path-exploration. These extensions affect the interdomain control-plane convergence. They are complementary to our solution.

Kushman et al. proposed in [16] to pre-compute interdomain failover paths around a peering link by adding information in BGP update messages. Upon failure, routers update their FIB to use the failover interdomain path. Another interdomain routing protocol that allows the propagation of alternative interdomain paths was proposed by Xu et al. in [35]. We took a completely different approach. First, our measurements show that alternate paths can be found at the borders of the AS. Thus, propagating alternative inter-

domain paths is not required. Second, our hierarchical FIB solves the key bottleneck which is actually the updates of the FIB.

Inside a single AS, Texeira et al. [29] have studied the impact of intradomain failures on BGP. Their measurements showed that the BGP convergence could be significantly delayed compared to the IGP convergence. We have shown and evaluated in this paper how BGP can efficiently track the IGP changes. Furthermore, our measurements confirm that upon changes our hierarchical FIB can be updated without risking any transient loss of connectivity.

Feamster proposed in [7] to develop a Routing Control Platform. With this approach, a centralized server would know all available paths and could compute both the primary and the secondary paths for each prefix on each router. A distributed implementation of the RCP was recently proposed in [31]. Using RCPs could be an alternative to the solutions discussed in section 6.

The solution which is closest to ours is the one described in [3]. It relies on a new FIB organization and on the utilization of tunnels between border routers to protect BGP peering links from failures. This solution only allows to protect from link failures, it does not protect from node failures. Furthermore, this solution also requires consistent BGP policies across peering links while this is not required by BGP PIC edge. This is an essential benefit of the automatic hierarchical FIB organization.

9. CONCLUSION

We proposed BGP prefix independent convergence, a novel component of a router supporting BGP traffic, that achieves sub-second convergence during network failures. The solution consists of a generalized hierarchical organization of the dataplane FIB table. By creating levels of indirection in the FIB table (BGP destinations to shared BGP path-lists to IGP path-lists and finally to the interface adjacencies), we limit the exploration of data structures in the FIB organization during a failure event, thus achieving convergence in a prefix independent manner.

Upon any intra-AS failure (core), the IGP convergence leads to modifications of IGP path-lists and hence all the dependent BGP destinations immediately converge. Upon any inter-AS failure (edge), the IGP convergence leads to the deletion of an IGP path-list which triggers the immediate modification of their few shared dependent BGP path-lists. In both of these events, the solution does not require exploration of BGP prefixes – thus the time for convergence remains constant as the number of BGP destinations increases over time. Our theoretical analysis and experimental characterization confirm this behavior.

For edge failures, our solution assumes encapsulation from border router to border router. This is not a constraint as the majority of SP networks run this model with either MPLS or IP encapsulation. The solution also requires the availability of at least two paths to a BGP destination. Through analysis of SP data, we explained why such destinations will have disjoint paths at the level of an AS. We also reviewed several techniques that enable the border routers to learn about these disjoint paths.

Our experimental analysis of the BGP PIC solution confirms its fundamental benefits in terms of convergence speed ($\sim$ 200 milliseconds instead of $\sim$ 2 minutes), scaling, and robustness.

Acknowledgments

We would like to thank our lead customers and the entire BGP PIC team at Cisco Systems: Kris Michielsen, Arunabha Saha, Dheerendra Talur, Karthik Subramanian, Ahmed Bashandy, Sunil Khaunte, Neil Jarvis.

We would also like to thank Bruno Quoitin for many suggestions and comments and Benoit Fondeviole, Vincent Gillet, Bruno Decraene and Sebastien Tandel for their help in collecting information and building the model of the Tier-1 ISP.

10. REFERENCES

[1] A. Basu, C. Luke Ong, A. Rasala, F. Bruce Shepherd, and G. Wilfong. Route oscillations in I-BGP with route reflection. In *SIGCOMM'02*, Pittsburgh, PA, USA, August 2002.

[2] T. Bates, Y. Rekhter, R. Chandra, and D. Katz. Multiprotocol Extensions for BGP-4. Internet Engineering Task Force, RFC2858, June 2000.

[3] O. Bonaventure, C. Filsfils, and P. Francois. Achieving sub-50 milliseconds recovery upon BGP peering link failures. In *Co-Next 2005*, Toulouse, France, Oct. 2005.

[4] A. Bremler-Barr, Y. Afek, and S. Schwarz. Improved BGP convergence via ghost flushing. In *IEEE Infocom*, March 2003.

[5] R. Bush, T. Griffin, Z. Morley Mao, E. Purpus, and S. Dtutsbach. Happy packets : Some initial results. Presented at NANOG, May 2004.

[6] J. Chandrashekar, Z. Duan, Z. Zhang, and J. Krasky. Limiting path exploration in BGP. In *IEEE INFOCOM*, Miami, Florida, March 2005.

[7] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *Future Directions in Network Architecture*, August 2004.

[8] N. Feamster, Z. Mao, and J. Rexford. BorderGuard: Detecting Cold Potatoes from Peers. In *ACM Internet Measurement Conference*, Taormina, Italy, October 2004.

[9] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure. Achieving sub-second IGP convergence in large IP networks. *SIGCOMM Comput. Commun. Rev.*, 35(3):35–44, 2005.

[10] T. Griffin and B. Presmore. An experimental analysis of BGP convergence time. In *ICNP 2001*, pages 53–61. IEEE Computer Society, November 2001.

[11] T. Griffin and G. Wilfong. On the correctnes of iBGP configuration. In *SIGCOMM'02*, pages 17–29, Pittsburgh, PA, USA, August 2002.

[12] J. Halpern, M. Bhatia, and P. Jakma. Advertising Equal Cost Multi-Path routes in BGP. Internet draft, draft-bhatia-ecmp-routes-in-bgp-02.txt, work in progress, February 2006.

[13] K. Kompella and Y. Rekhter. Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling. RFC 4761 (Proposed Standard), January 2007.

[14] R. Rao Kompella, J. Yates, A. Greenberg, and A. Snoeren. IP fault localization via risk modeling. In *USENIX NSDI'05*, Boston, USA, May 2005.

[15] N. Kushman, S. Kandula, and D. Katabi. Can you hear me now?!: it must be BGP. *SIGCOMM Comput. Commun. Rev.*, 37(2):75–84, 2007.

[16] N. Kushman, S. Kandula, D. Katabi, and B. Maggs. R-BGP: Staying Connected in a Connected World. In *4th USENIX NSDI*, Cambridge, MA, April 2007.

[17] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. An experimental study of internet routing convergence. In *SIGCOMM 2000*, August 2000.

[18] Z. M. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates internet routing convergence. In *ACM SIGCOMM'2002*, 2002.

[19] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *IEEE Infocom2004*, Hong Kong, March 2004.

[20] D. Meyer, L. Zhang, and K. Fall. Report from the IAB Workshop on Routing and Addressing. Internet draft, draft-iab-raws-report-01.txt, work in progress, February 2007.

[21] D. Pei, M. Azuma, N. Nguyen, J. Chen, D. Massey, and L. Zhang. BGP-RCN: Improving BGP convergence through Root Cause Notification. *Computer Networks*, 48(2):175–194, June 2005. 2005.

[22] D. Pei and J. Van der Merwe. BGP convergence in Virtual Private Networks. In *Internet Measurement Conference*, Rio de Janeiro, Brazil, October 2006.

[23] B. Quoitin and S. Uhlig. Modeling the routing of an Autonomous System with C-BGP. *IEEE Network*, 19(6), November 2005.

[24] Y. Rekhter and P. Gross. Application of the Border Gateway Protocol in the Internet. RFC 1655 (Proposed Standard), July 1994. Obsoleted by RFC 1772.

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

[26] E. Rosen and Y. Rekhter. BGP/MPLS IP Virtual Private Networks (VPNs). RFC 4364 (Proposed Standard), February 2006. Updated by RFCs 4577, 4684.

[27] S. Sangli, D. Tappan, and Y. Rekhter. BGP Extended Communities Attribute. RFC 4360 (Proposed Standard), February 2006.

[28] A. Sridharan, S. Moon, and C. Diot. On the correlation between route dynamics and routing loops. In *Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement*, pages 285–294. ACM Press, 2003.

[29] R. Texeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *SIGMETRICS/Performance'04*, New York, NY, USA, June 2004.

[30] S. Uhlig and S. Tandel. Quantifying the impact of route-reflection on BGP routes diversity inside a Tier-1 network. In *IFIP Networking 2006*, Coimbra, Portugal, May 2006.

[31] Patrick Verkaik, Dan Pei, Tom Scholl, Aman Shaikh, Alex C. Snoeren, and Jacobus E. van der Merwe. Wresting Control from BGP: Scalable Fine-grained Route Control. In *USENIX Annual Technical Conference 2007*, Santa Clara, USA, June 2007.

[32] D. Walton, A. Retana, and E. Chen. Advertisement of Multiple Paths in BGP. Internet draft, draft-walton-bgp-add-paths-05.txt, work in progress, August 2006.

[33] F. Wang, Z. Mao, J. Wang, L. Gao, and R. Bush. A measurement study on the impact of routing events on end-to-end Internet path performance. In *ACM SIGCOMM*, pages 375–387, Pisa, Italy, September 2006.

[34] D. Watson, F. Jahanian, and C. Labovitz. Experiences with monitoring OSPF on a regional service provider network. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 204. IEEE Computer Society, 2003.

[35] W. Xu and J. Rexford. Miro: multi-path interdomain routing. In *SIGCOMM '06*, pages 171–182, New York, NY, USA, 2006. ACM.

[36] Alex Zinin. *Cisco IP Routing: Packet Forwarding and Intra-domain Routing Protocols*. Addison Wesley Professional, 2002.