Abstract

A design for a fabric of switches to connect up to one million servers in a data center is described. At that scale, it is impractical for every switch to maintain knowledge about every other switch and every other link in the fabric. Aggregation of routes is an excellent way to scale such a fabric. However, aggregation presents some problems under link failures or switch failures. This design solves those problems.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

[RFC7938] defines a massive scale data center as one that contains over one hundred thousand servers. It describes the advantages of using BGP as a routing protocol in a Clos switching fabric that connects these servers. It laments the need to announce all routes individually, because of the problems associated with route aggregation. A fabric design that scales to one million servers is considered enough for the foreseeable future and is the design goal of this document. Of course, the design should also work for smaller fabrics.

A switch fabric to connect one million servers will consist of between 35000 and 130000 switches and 1.5 million to 8 million links, depending on how redundantly the servers are connected to the fabric and the level of oversubscription in the fabric. A switch that needs to store, send and operate on hundreds of routes is clearly cheaper than one that needs to store, send and operate on millions of links.
A switch running BGP and aggregating its routes needs to send only one route. In the ideal case, each switch receives just one route from each of its neighbors. For each link or a neighbor that fails, the switch should send just one extra route. No single link failure needs to be known by every switch in the fabric and some switch failures do not need to be known by every switch either. The routes that advertise these failures should only propagate to those switches that need to know about them. During normal operation, the number of failures are few, so the number of advertisements are few.

A route that advertises a failure is called a negative route. Negative routes are not a new idea, but they are unpopular, because they cause a number of problems. This document solves the problems.

2. Solution Overview

In a Clos network all northbound links can reach all destinations and there is typically only one or very few southbound links to reach any specific destination. Therefore, traffic from source to destination is spread to all available northbound links, reaches all the spines and then concentrates southbound towards its destination. When a link fails, then a spine will lose connectivity to some southbound destinations. That means any northbound link to that spine also loses connectivity to the same destinations.

When the fabric is fully connected with no failed links, then the forwarding tables in the switches can simply contain multipath aggregate routes to all the northbound links. Each of the multipath routes is the same, so traffic is spread out smoothly among these routes. As soon as a link fails, the forwarding tables must exclude the resultant unreachable destinations from some of the northbound links. The way to do that is to add specific routes for the failed destinations to point at the remaining links that can reach those destinations. Since traffic will always prefer specific routes to aggregate routes, the traffic to the failed destinations will no longer take the aggregate routes.

Two methods to create these specific routes are described. One way is to send a negative route from the point where the failure is detected. Receivers use the negative route to punch holes out of the aggregate routes and create the specific routes by subtracting the negative route from the aggregates. This method is described starting at section 4. The other method creates the specific routes at the point of the failure and announces them in BGP. This method is described starting at section 9.
3. Problems with negative routes

- Massive failures can cause lots of negative routes and overwhelm the switches.

- In order for a switch to know what has failed, it must know what is supposed to be up. For it to know this requires either an error prone algorithm or an error prone configuration.

- During certain network events that cause multiple routes to be sent and/or withdrawn, the messages may race each other and cause transient loss of connectivity to paths that were otherwise unaffected by the event. This occurs in link state routing protocols as well.

- Computation of forwarding table entries may consume a lot of CPU time in pathological cases. However, even in pathological cases, this is still much less CPU time than it takes to compute an SPF in a million links.

4. Use of a negative route in BGP

Three new BGP well known communities are defined:

- Hole-Punch: A route with this community can punch a hole out of another route with a shorter netmask that covers the address space of this route.

- Punch-Accept: A route with this community can have holes punched out of it by hole punch routes.

- Do-not-Aggregate; Do not aggregate this route.

A fabric switch will aggregate routes learnt from neighbors to its south. It must know all the routes that are expected to complete the aggregate. It will announce the aggregate with the Punch-Accept community. If any of the routes that are expected to complete the aggregate are missing, then it will announce those missing routes with the Hole-Punch and Do-not-Aggregate communities along with the aggregate route.

A receiver of a route with the Hole-Punch community will give it a lower than normal local preference and will search the BGP table for other routes with the following properties:

- a shorter netmask than this route,

- covers the address space of this route,
has the Punch-Accept community,

- is installed in the Routing Table.

This is the candidate set. Then, it will remove any routes that have a shorter netmask than the route with the longest netmask in the set. The final candidate set of routes will all have the same prefix. For each route in the candidate set, BGP will create a new route with the same prefix as the Hole-punch route and the same attributes as the Punch-Accept route. This new route is called a chad route. If a route has an MPLS label, then the label is considered part of the attributes, not part of the prefix.

Chad routes will take part in bestpath and multipath selection. If a chad route becomes a bestpath or a multipath, it will be installed in the Routing Table. However, chad routes are not advertised by default. That means if a chad route is bestpath and other routes exist for the same prefix, then no route is advertised for that prefix.

If a chad route has the same nexthop (and MPLS label, if labels are used) as a hole-punch route of the same prefix, then the chad route becomes hidden. Hidden means that it cannot take part in route selection.

5. Implementation Notes to Reduce CPU Time Consumption

This section is not normative.

When a Punch-Accept route is received, BGP needs to scan a subtree of the BGP prefix table rooted at the prefix of the Punch-Accept route to look for Hole-Punch routes that might create chad routes from it. That subtree could be large. To reduce the number of routes to scan, a separate prefix table is created to store copies of the Hole-Punch routes. The number of Hole-Punch routes is expected to be much smaller than the total number of routes. That makes the scan much quicker. The Hole-Punch routes must additionally be stored in the regular BGP route table.

6. Smooth Startup and Avoidance of Too Many Negative Routes

When several switches of a data center fabric start up at the same time, many negative routes can be transiently created before the whole system is up.

When the BGP process starts, it will typically start in receive-only mode for some time, then perform route selection and send out its own updates. To ensure a smooth startup of the data center when many
nodes start at the same time, the startup sequence is modified as follows.

- All BGP speakers SHOULD send EOR after sending all routes after the BGP session becomes established.

- When all southbound configured BGP neighbors have sent their EOR, the BGP speaker will perform route selection and send all updates to the northbound neighbors and then send EOR. If some southbound neighbors cannot establish, a timer will be used to prevent waiting forever.

- After the previous step completes, when all northbound configured BGP neighbors have sent their EOR, the BGP speaker will perform route selection and send all updates to the southbound neighbors and then send EOR. If some northbound neighbors cannot establish, a timer will be used to prevent waiting forever.

- If the number of received negative routes causes too many forwarding entries, then BGP can look for aggregate routes that are accompanied by many hole-punch routes and invalidate some of the aggregate routes and their accompanying hole-punch routes. If the number of received negative routes is too many to hold in the BGP table, then BGP can shut down neighbor sessions that are sending the most negative routes.

7. Avoidance of Transients

If one event were to cause both an aggregate and a hole-punch route to be announced at the same time, but the hole-punch route were to arrive late, a transient could result. The following rules prevent that.

- It is common practice for aggregate routes to be withdrawn when no components of the aggregate exist. Hole-Punch routes need to always be announced, even if the aggregate is not.

- After a BGP session establishes, no routes that are received from it should be installed in the RIB until the EOR is received from that session.

- If overlapping hole-punch routes need to be updated and withdrawn, then the updates must be sent before the withdraws.

- If overlapping hole-punch and punch-accept routes need to be updated, then the hole-punch routes must be updated first.
If overlapping hole-punch and punch-accept routes need to be withdrawn, then the punch-accept routes must be withdrawn first.

8. Configuration

All the BGP sessions need to be configured on each switch. The BGP sessions need to be configured as northbound or southbound. The routes that are expected to complete an aggregate route must be configured.

A companion document describes a protocol that can discover and configure the entire fabric. If that companion document is used, then no IP addresses or tier designations or any other location dependent configuration is required on the switches.

9. South Triggered Automatic Disaggregation (STAD)

In this method, a node that is south of a failed link or node announces its prefix(es) along alternative links with a hint to trigger automatic disaggregation or inhibit their suppression on upstream tier nodes. These disaggregated or unsuppressed routes traverse along redundant paths and disjoint planes to switches in other clusters in the topology where they are used in forwarding.

The hint is in the form of a well known BGP community. A few new well known communities are used in this scheme.

- Do-not-Aggregate : Do not aggregate this route
- Tier : An Extended Community identifying the tier of the originated route
- Dis-Aggregate : Triggers announcement of more specific routes at receiving node

The techniques in this draft assume a CLOS topology of the form described in [Figure 3] of RFC7938 where an access switch such as a TOR forms the lowest tier and is connected to multiple northbound upper tier switches; which in turn are connected to multiple upper tier switches, forming disjoint planes across the topology with fan-outs.

A figure illustrating the topology will be added in a subsequent version.

Upon a link failure, the node south of the failure announces its prefix to it’s other northbound BGP sessions with the Do-Not-Advertise community.
A higher tier node that receives a route with a Do-not-Aggregate community will not suppress this route when there is a local covering aggregate, but will propagate it further as is.

This procedure enables the more specific route to reach the appropriate tier switches in other clusters where the topology fans out on multiple northbound links. The received paths for the more specific prefix form a multipath excluding the links which would lead to failed paths in the topology.

A route that is advertised with the Do-Not-Aggregate community as per this section will also add a Tier extended community. If this extended community is present, then the Do-Not-Aggregate community is only applicable at tiers that are more north than the tier indicated in the extended community.

The Tier Extended Community ensures that the unsuppressed specific routes do not propagate further beyond the corresponding fan-out points in the other clusters.

If all the northbound links or BGP sessions at a node have failed, then the node will announce its southbound route with the Dis-Aggregate community. This signals all its south-side nodes to advertise their north-bound routes with the Do-Not-Aggregate community along the other north-bound links.

These techniques are applicable to any tier in the topology.

At the lowest tier, if there are servers that are attached to more than one fabric switch (eg. TOR), then the host routes (or configured more-specific routes) for the server are not aggregated by the TOR to its connected upper tier switches. In this case, these routes are aggregated by the upper-tier switches towards the rest of the topology.

10. Configuration for STAD

Each switch has the notion of northbound and southbound sessions or links. In addition, it is assigned to a tier in the hierarchy. The switch uses this configuration to drive the procedures described in the section above. A switch at the lowest tier (eg. a TOR) will have server subnet prefixes configured. Switches at higher tiers have aggregates configured.
11. Security Considerations

TBD

12. IANA Considerations

TBD

13. Acknowledgements

14. References

14.1. Normative References


14.2. Informative References


Authors’ Addresses

Jakob Heitz
Cisco
170 West Tasman Drive
San Jose, CA, CA 95134
USA

Email: jheitz@cisco.com

Dhananjaya Rao
Cisco
170 West Tasman Drive
San Jose, CA, CA 95134
USA

Email: dhrao@cisco.com