Routing Design for Large Scale Data Centers:
BGP is a better IGP!

Presented by: Parantap Lahiri, George Chen, Petr Lapukhov, Edet Nkposong, Dave Maltz, Robert Toomey, and Lihua Yuan

Global Networking Services Team, Global Foundation Services, Microsoft Corporation
Agenda

Problem Statement
What we started with
Why BGP over IGP
The new approach
Details and design choices
Problem Statement
Online Service DC Specifics

Server Perspective
  100’s thousands of servers
  10G NICs

Distributed Applications
  Aware of the network
  Explicit parallelism
  Example: Web Index computation

“Network as a computer” concept
Two types of traffic flows

Query
Background

Query
North/South
Scatter-gather

Background
East/West
Compute & Synchronize
Problem Statement

Build a topology providing significant amount of bisection bandwidth

The simpler the better

Design a scalable routing model for this topology

Single protocol
Simple behavior
Wide vendor support
What We Started With
Multiple definitions exist...

Has N stages (N=3,5,7..)
  Folded on diagram

Full bisection bandwidth if $M \geq N$

Natural link load-balancing
  ECMP Based
What we started with: Topology

3-Stage Clos

Leafs deployed in pairs

Oversubscribed at ToR layer

Layer 2 from servers up to the Leafs

MLAGs for bandwidth aggregation at L2
What we started with: Routing

BGP for routing

eBGP between Leaf and Spine devices

VLANs advertised into BGP on the Leafs

ECMP for load-sharing across multiple links
Why BGP over IGP
BGP Simplicity

Simpler protocol design concepts compared to IGPs

- Better vendor interoperability
- Less state-machines, data-structures etc

BGP allows for per-hop traffic engineering

Use for unequal-cost Anycast load-balancing solution
BGP Simplicity

Troubleshooting BGP is simpler
- BGP RIB structure is simpler compared to link-state LSDB
- Clear picture of what sent where (RIBIn, RIBOut)

Event propagation is more constrained in BGP
- E.g. link failures have limited propagation scope
- More stability due to reduced event “flooding” domains
Common arguments against BGP

What about configuration complexity – BGP neighbors, etc?
   Not a problem with automated configuration generation

What about convergence properties?
   Is not our primary goal anyways, few seconds are OK
   Practical convergence in less than a second
The New Approach
Limitations of BGP + L2 design

L2 issues

Broadcast storms
Hard to troubleshoot

MLAGs are proprietary

Single spine scales “up” only

MLAGs limit us to two Leafs per container

Bandwidths scales up, and not out
Topology for new deployments

Scaled-out Clos!

Think multiple parallel Clos topologies

Lower port density on switches

Horizontal scaling at every layer above ToR
Routing Design for Parallel Clos

BGP all the way down to the ToR (eBGP)
Separate BGP ASN per ToR
Benefits of new approach

No more L2 problems!

Bandwidth now scales out everywhere

No need to buy higher-radix boxes
Cheaper infrastructure

Uniform routing protocol

No interworking/redistributions etc

BGP AS_PATH visibility allows for easier troubleshooting
Details and Design Choices
BGP Specific: Features

Requires “BGP AS_PATH Multipath Relax”

- We rely on ECMP for routing
- Needed for Anycast prefixes

We use 16-bit Private BGP ASN’s ONLY

- Simplifies path hiding at WAN edge (remove private AS)
- Simplifies route-filtering at WAN edge (single regexp)

But we only have 1022 Private ASN’s…
BGP Specifics: Allow AS In

Reuse Private ASNs on the ToRs

Use of *Allow AS in* on ToR eBGP peerings

Effectively, ToR numbering is local to the container

Requires vendor support…
Message to the Vendors

There isn’t that many requirements…

Please implement uniform BGP features

  AS_PATH Multipath Relax
  Allow AS In
  Fast eBGP Fall-over
  Remove Private AS

There is more, but it’s a topics for separate discussion
Design Specifics: Default Routing

Don’t use “default route only” model

Don’t hide specific prefixes

Otherwise: Route Black-Holing on link failure!
Design Specifics: Route Summarization

Don’t summarize server subnets!

Summarizing P2P links is OK

Otherwise: Route Black-Holing on link failure!
Summary

BGP has been thought as slow and suitable for inter-domain routing only…

With modern implementations, it might as well work as IGP!

BGP is simple, allows for per-hop traffic engineering and supported by practically all vendors

This made it perfect choice for us!
Questions?