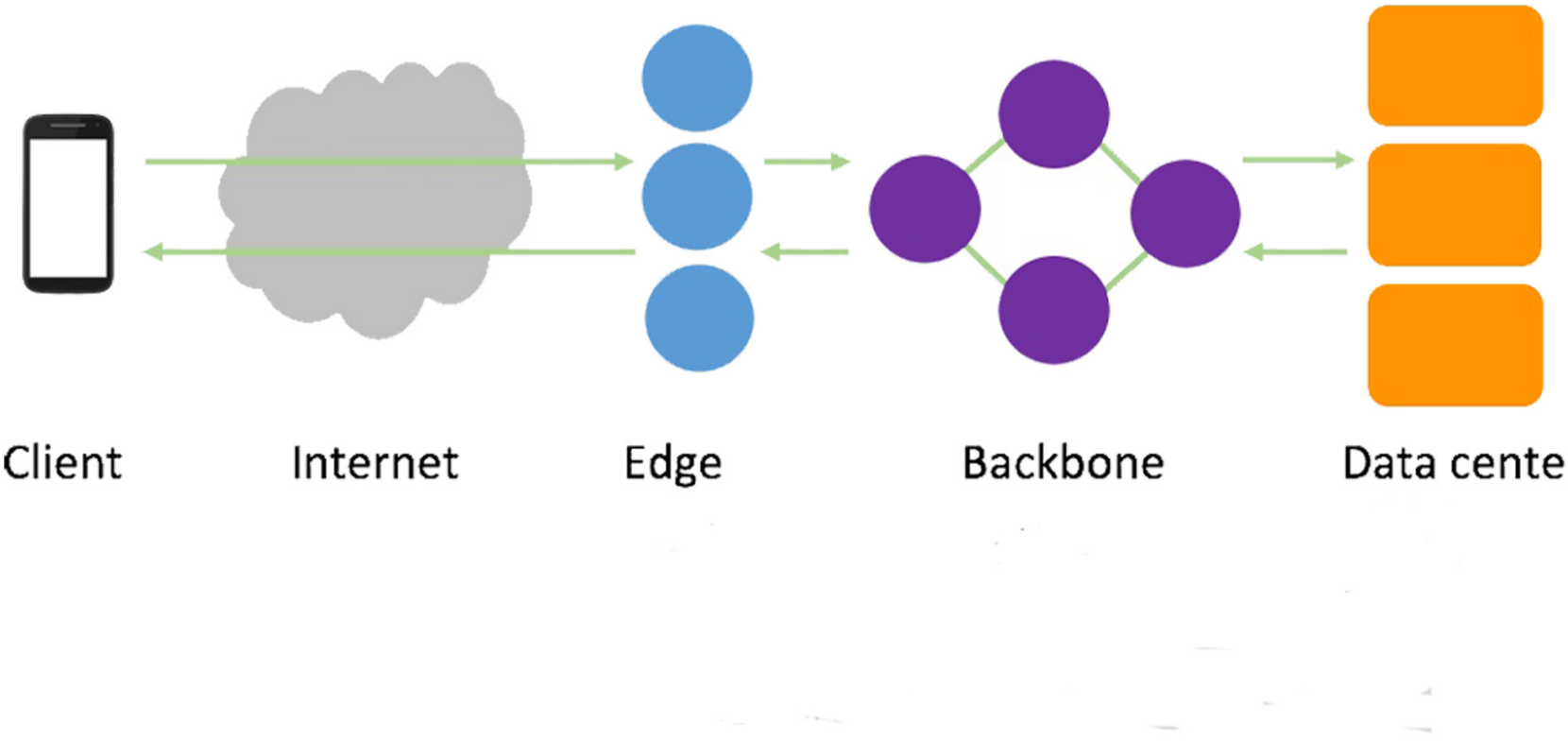
POSTED ON AUG 21, 2017 TO NETWORKING & TRAFFIC

Steering oceans of content to the world



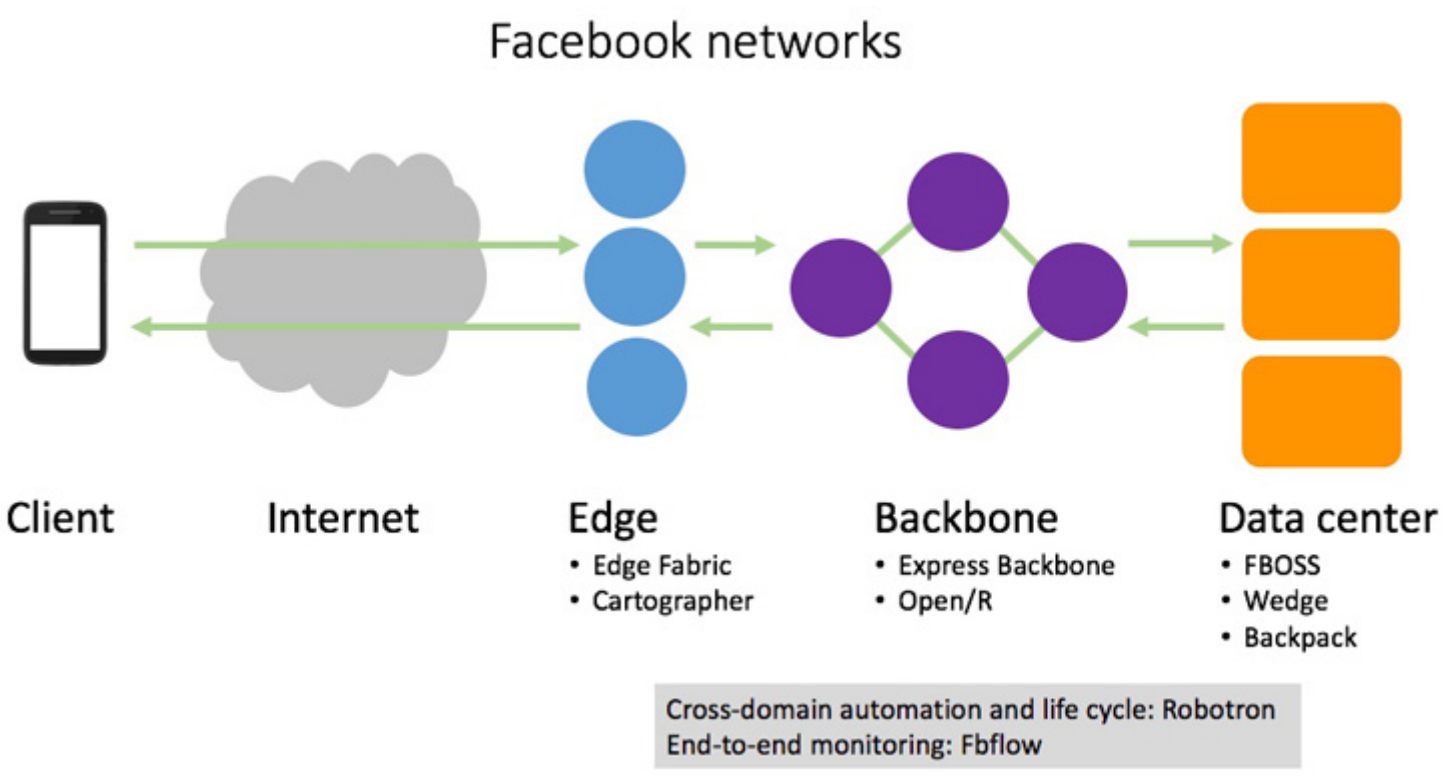
By Hyojeong Kim James Hongyi Zeng

A version of this post was originally published at research.fb.com.

The Facebook network is an integral part of Facebook's infrastructure. Every day, people and communities rely on Facebook's global network and infrastructure to share information, ranging from photos and videos to messages and News Feed, and expanding to more interactive and rich 360 and VR applications. Facebook has built out a global network of points of presence (PoPs), each connected to tens or hundreds of networks. In order to support Facebook's growing external demand and internal services, we embarked several years ago on building in-house hardware and software systems to scale our global network.

Some of the systems we have developed focus on particular domains in the network (data center, backbone, edge), while some span across domains and tackle specific stage of the network lifecycle:

- In the data center — FBOSS, Wedge, and Backpack are the hardware and software that power the open switches running in our data centers, a key step toward a disaggregated network.
- In the backbone — Express Backbone and Open/R make up our SD-WAN, a dedicated private backbone network that leverages centralized traffic engineering in a global controller coupled with a new distributed routing system.
- Cross-domain automation/life cycle — Robotron is a system that handles every part of the network life cycle across all network domains, from data center to backbone and edge.
- End-to-end monitoring — Host-based system fbflow provides end-to-end flow monitoring.



In the edge of the Facebook network, we built a system called Edge Fabric to make our edge networking faster, more efficient, and more flexible. This completes the overall picture of Facebook's in-house software-centric networking stack, across all domains of the Facebook network. Edge Fabric has been in production for the past several years, steering egress user traffic from our PoPs around the world.

Our edge networking team needed to take into account the complex connectivity options in each PoP, the preference of each network, and application-level performance. We soon found that traditional routing protocols could not address our needs. Before we dive into the system details, let's first set the context.

Challenges

When somebody visits Facebook, our global load-balancing system, Cartographer, assigns them to PoPs via DNS. After arriving at a PoP, user requests either terminate there or are relayed back to our data centers for further processing. Responses are sent back to users via the same PoP.

At each PoP, Facebook can have diverse routing options to reach a particular person. The most common options are:

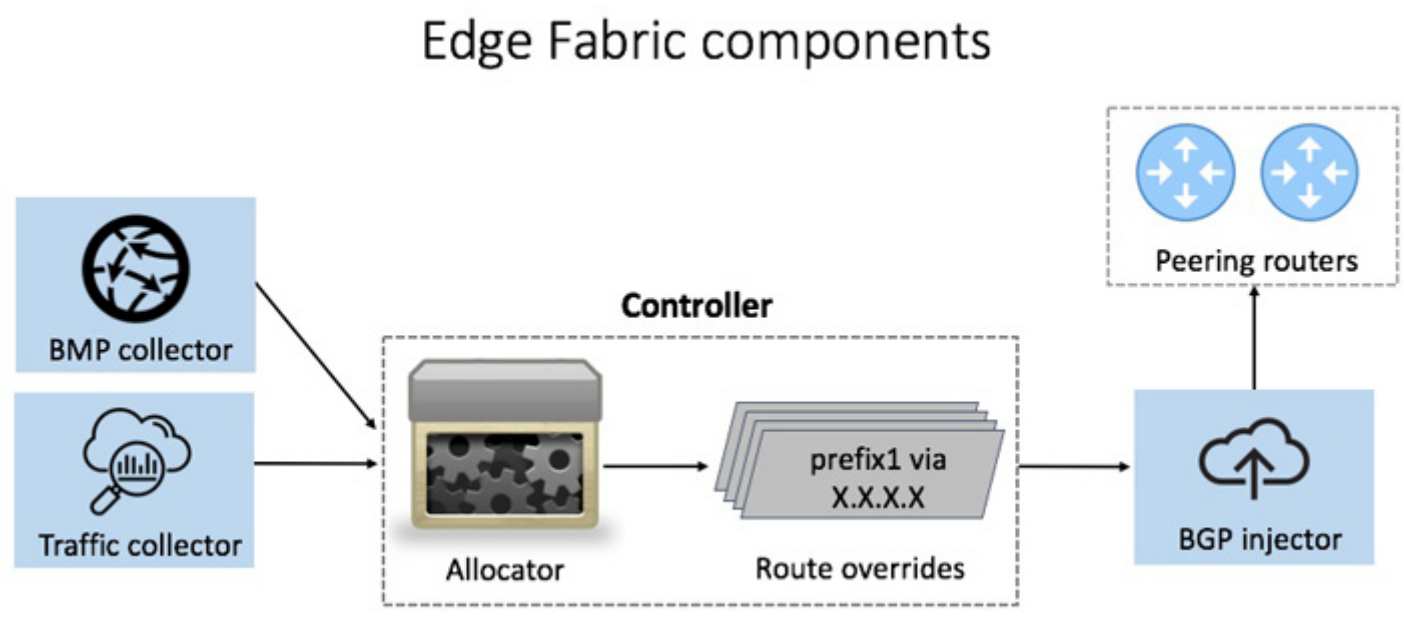
- Private peering — a dedicated link to the end user's network.
- Public peering — a shared fabric of an internet exchange point to the end user's network.
- Transit provider — another network that promises to deliver the content to the end user's network eventually.

Normally, that order is the preference of most networks. Such preference is configured statically and controlled by Border Gateway Protocol (BGP), a protocol between networks for exchanging routing information.

We found that BGP on its own was unable to easily address a couple of important limitations:

- **Peering capacity is limited, but BGP is not capacity-aware.** A link's capacity may not be able to deliver all traffic that Facebook would like to send over it. Rapid growth in demand can quickly overwhelm the capacity of an existing interconnection. Short-term spikes in demand (perhaps due to an event or a holiday) or reductions in capacity due to failures can cause volatility in demand and available capacity. In any of these cases, BGP's decision won't change, as it is not capacity-aware.
- **BGP decisions can hurt performance.** Facebook's BGP policy favors paths that are likely to optimize network traffic performance. However, because BGP itself is not performance-aware, this policy relies on attributes such as path length that serve as imperfect heuristics for proximity and performance. Our paper gives several examples of where these heuristics can go wrong.

System architecture



In 2013, we started to develop Edge Fabric to overcome these challenges. Edge Fabric consists of several loosely coupled microservices.

Data collection

Edge Fabric first needs to know all the routes from a PoP to a destination, and which routes traffic will traverse if it does not intervene. In addition, it needs to know the volume of traffic per destination prefix and the capacity of egress interfaces in the PoP. We built a BMP collector (BGP monitoring protocol) and traffic collector to assemble this information.

Controller

The allocator projects what the utilization of all egress interfaces in the PoP would be if no overrides are injected, assigning each prefix to its preferred route(s) from its emulated BGP path selection. The BGP best path computation process may return multiple equivalent routes.

Every 30 seconds, the allocator generates overrides to shift traffic away from interfaces that it projects will be overloaded. For each overloaded interface, the allocator identifies the prefixes projected to traverse the interface and, for each prefix, the available alternate routes. It then identifies the single ⟨prefix, alternate route⟩ pairing that it prefers to shift first from the interface, applying the several rules in order until a single preference emerges.

Once a pairing has been selected, the allocator records the decision and updates its projections, removing the prefix's traffic from the original interfaces and placing all of the PoP's traffic for the prefix onto the interface of the selected alternate route.

Enacting overrides

In each round, the allocator generates a set of BGP updates for Edge Fabric's overrides and assigns each update a very high local preference. The allocator passes the BGP updates to the BGP Injector service, which maintains a BGP connection with every router in the PoP and enacts the overrides by announcing the BGP updates to the target routers. The injector service then withdraws any overrides that are no longer valid in the current allocation.

These microservices together make up the Edge Fabric system, and the effect is that Edge Fabric is able to augment BGP with measurement and control mechanisms to overcome BGP's lack of congestion awareness and performance awareness. We designed it to be simple and scalable, taking advantage of centralized control and existing support in vendor software and hardware.

Our new paper has more details about these microservices, as well as how we incorporate application-level performance metrics such as bandwidth and latency into the decision process. We also share a brief history of system evolution as well as our operation experience.

Looking ahead

Today's internet traffic is generally managed by a small number of big content providers. How they interact with other networks largely shapes inter-domain routing around the world.

BGP will be the internet's inter-domain routing standard for the foreseeable future. By sharing our experience engineering our egress traffic, including a detailed look at the opportunities and challenges presented by the internet connectivity of today's large content providers, we hope that the limitations of BGP can be better understood and the experience of everybody on the internet can be improved.

Edge Fabric complements the systems Facebook has previously shared, as it goes into details of the edge of our network, in addition to our backbone, data center networks, and management systems. Overall, Facebook has taken a software-centric approach to the future of our network, and we look forward to continuing to share and discuss these network systems with both industry and academia.

TAGS: INFRA

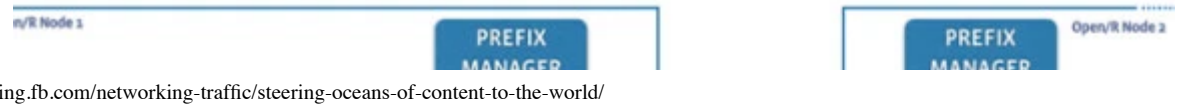
Related Posts

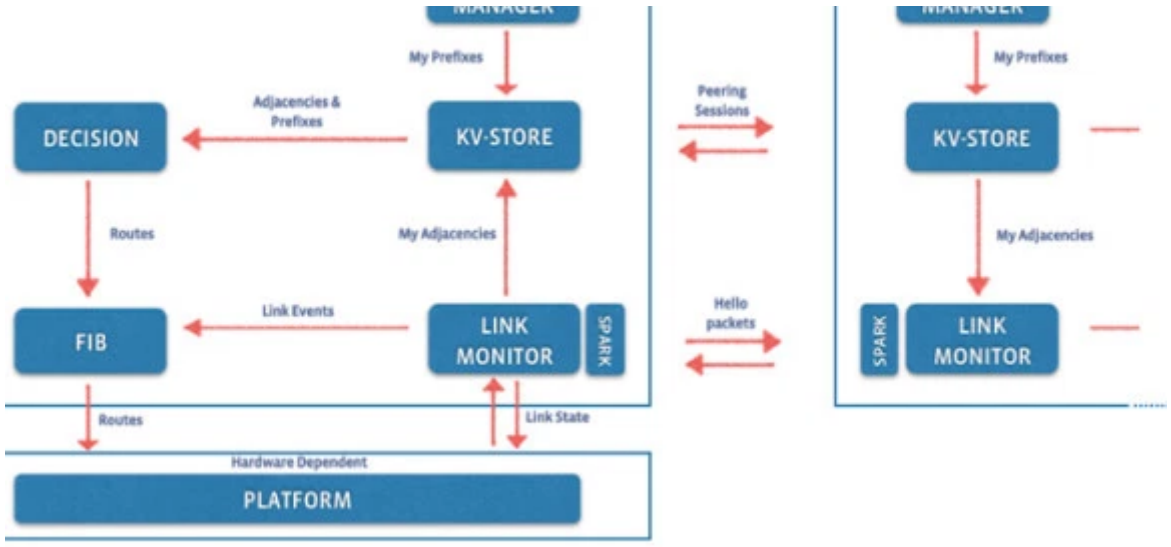
Mar 19, 2018

Performance @Scale 2018 recap

Sep 25, 2019

Networking @Scale 2019 recap





Nov 15, 2017

Open/R: Open routing for modern networks

Related Positions

Software Engineer (AI)

PARIS, FRANCE

System Test Engineering Architect, Portal

MENLO PARK, US

Solutions Engineer

MENLO PARK, US

Software Engineer, Ads Ranking

SEATTLE, US

Software Engineer, Machine Learning (IGTV)

SAN FRANCISCO, US

See All Jobs

Join Our Engineering Community

Available Positions

Software Engineer (AI)
PARIS, FRANCE
System Test Engineering Architect, Portal
MENLO PARK, US
Solutions Engineer
MENLO PARK, US
Software Engineer, Ads Ranking
SEATTLE, US
Software Engineer, Machine Learning (IGTV)
SAN FRANCISCO, US

See All Jobs

Stay Connected

Facebook Engineering

Like

@fbOpenSource

Follow

Open Source

Facebook believes in building community through open source technology. Explore our latest projects in Artificial Intelligence, Data Infrastructure, Development Tools, Front End, Languages, Platforms, Security, Virtual Reality, and more.

ANDROID

iOS

WEB

BACKEND

HARDWARE

Learn More

facebook research

Facebook Research

Like

Facebook Developers

Like

RSS

Subscribe