



August 24, 2016

Robotron: Top-down Network Management at Facebook Scale

By: James Hongyi Zeng, Starsky H.Y. Wong, Xiaozheng Tie, Yu-Wei Eric Sung



Robotron: Top-down Network Management...

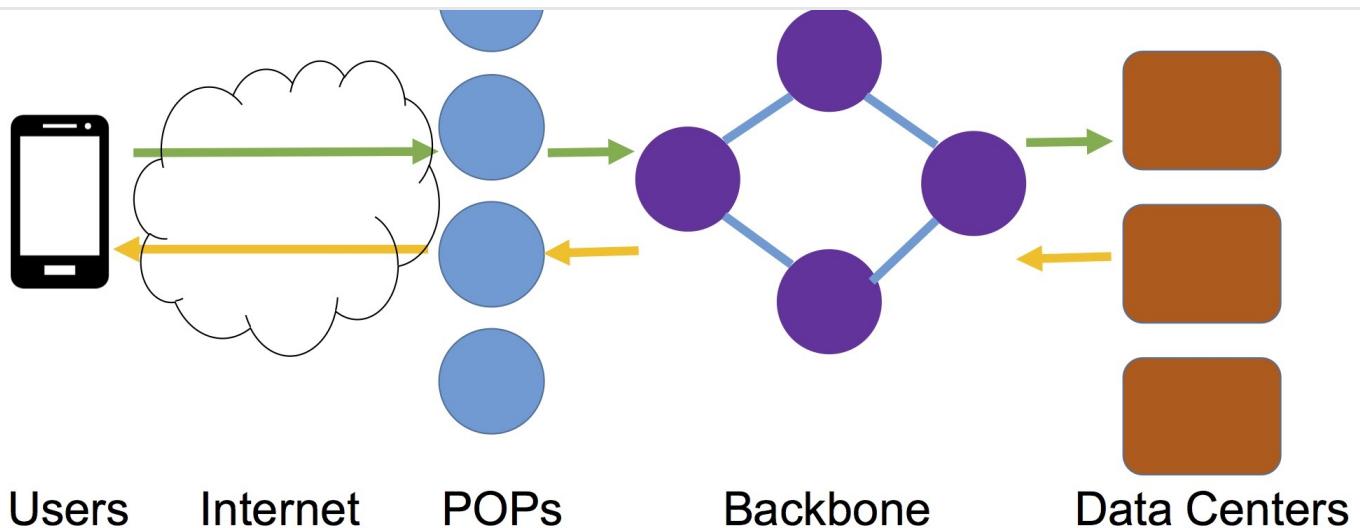
Posted by Facebook Research

2,121 Views

Managing a healthy and sustainable network is difficult. However, little is understood about the networking management practices outside the network engineering community. We developed a state-of-the art system named Robotron to manage tens of thousands of network devices connecting hundreds of thousands of servers globally at Facebook. This week, we are presenting an overview of the system in the paper [Robotron: Top-down Network Management at Facebook Scale at SIGCOMM 2016](#) in Florianópolis, Brazil.

The Robotron system is designed to manage a massive production network in a top-down fashion. The goal, reducing effort and errors on management tasks by minimizing direct human interaction with network devices, is easier said than done. Managing a large, dynamic, and heavily utilized network is challenging. Every day, network engineers perform numerous diverse tasks such as circuit turn-up and migration, device provisioning, OS upgrade, access control list modification, tuning of protocol behavior, monitoring of network events and statistics, etc.

Our paper also sheds some light on key learnings the team discovered while designing, implementing and operating the Robotron system which has managed Facebook's production network of data centers, global backbone, and edge point of presence (POPs) over the last eight years. User requests enter Facebook's network via nearby edge POPs. Any request unable to be fulfilled by our POPs is routed through our backbone to our data centers (DCs), at which point the request is processed and the response is sent back to the user. By sharing our experiences with Robotron, we hope to motivate more future research in the area of network management.



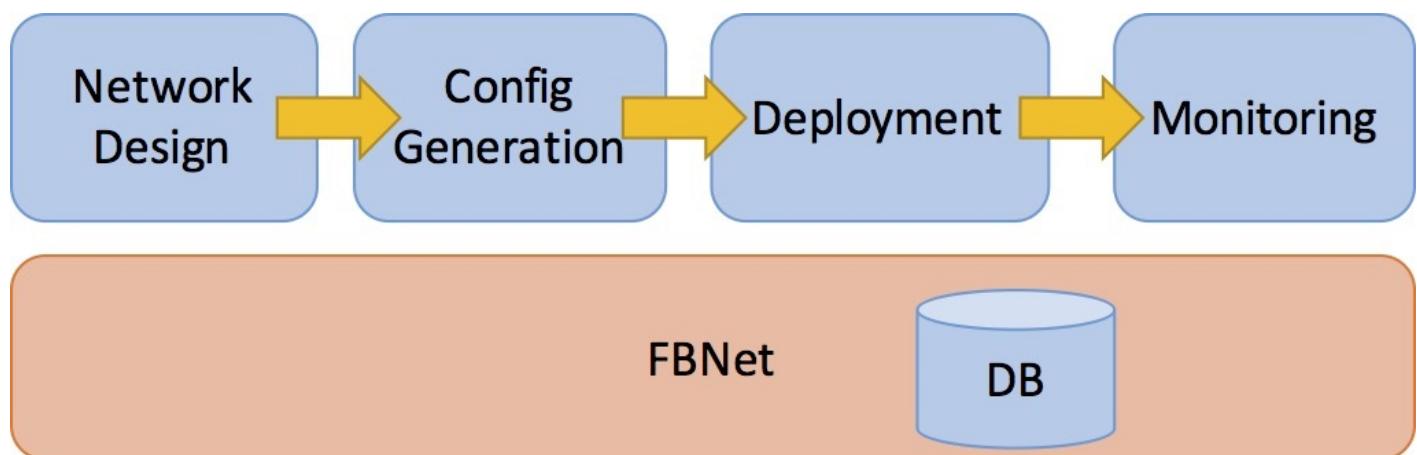
Motivation and approach

Network engineers highly value judicious network management for several reasons. First, a properly configured network is a prerequisite to higher-level network functions. For example, routing protocols may not function correctly if an underlying circuit is not provisioned as planned. Second, since network management tasks naturally involve human interactions, they are highly risky and can cause high-profile incidents. And finally, agile network management enables the network to evolve quickly, e.g., adding new devices or upgrading capacity, to support fast changing application needs. However, the field of network management is traditionally considered too *operational* and lacks published principles. While developing the Robotron system, the team focused on addressing challenges in several key network management areas: distributed configurations, handling multiple domains, versioning, dependencies, and vendor differences.

The system's top-down approach translates human intentions into a set of distributed, heterogeneous configurations. Beyond configuration generation, Robotron also deploys and monitors configurations to ensure the actual state of the network does not deviate from design. In addition to describing Robotron's design and implementation in detail, the paper also highlights Robotron's usage statistics to shed light into the operations of Facebook's production network.

Design

As with many companies, Facebook relied heavily on manual configuration and ad-hoc scripts to manage its network in its early days. In 2008, we developed FBNet, an object store to model high-level operator intent. Since then, FBNet and the suite of network management software built



Using FBNet as the foundation, Robotron covers multiple stages of the network management life cycle: network design, config generation, deployment, and monitoring.

FBNet: FBNet is the central repository for information, implemented as an object store, where each network component is modeled as an object. Object data and associations are represented by attributes. For example, a point-to-point circuit is associated with two interfaces. The circuit and interfaces are all objects connected via the circuit's attributes. FBNet serves as the *single source of truth* for network component state, used in the life cycle stages described below.

Network Design: The first stage of the management life cycle is translating the high-level network design from engineers into changes to FBNet objects. For example, when designing a cluster, an engineer must provide high-level topology information, e.g., number of racks per cluster, number of uplinks per top-of-rack switch, etc. Robotron realizes the design in FBNet by creating top-of-rack switch, circuit, interface, and IP address objects for the cluster.

Config Generation: After FBNet objects are populated, the config generation stage builds vendor-specific device configs based on object states. Config generation is highly vendor- and model-dependent. A set of template configs, which are extended as new types of devices are put into production, enables FBNet to provide the object states necessary for each build.

Deployment: Once device configs are generated, the next stage is to deploy them to network devices. Correct and safe multi-device deployment can be challenging. Many design changes affect multiple heterogeneous devices. To reduce the risk of severe network disruptions, changes are deployed in small phases before reaching all devices.



active network. For example, all production circuits are monitored to ensure they are up and passing traffic.

Evolution

Robotron's design has evolved significantly since 2008. Perhaps counter-intuitively, it did not start out as a top-down solution. Instead, its initial focus was on gaining visibility into the health of the network through active and passive monitoring systems. FBNet was created to track basic information about network devices such as loopback IPs and store raw data periodically discovered from network devices. However, per-device data was too low-level, vendor-specific, and sometimes required piecing multiple data together to construct meaningful information, making it extremely difficult to consume. As a result, basic models were created in FBNet to store a normalized, vendor-agnostic view of the actual network state constructed from the raw data. Ad-hoc audits could then be easily written against the models to look for design violations, misconfigurations, hardware failures, etc.

With basic monitoring in place, the team started tackling the other stages of the network management lifecycle and quickly encountered two main challenges based on user feedback. First, deployment of config updates (e.g., changes to routing or security policies) to a large number of devices was still manual and required logging into each device and copying and pasting configs. To address this, a deployment solution was developed to enable a scalable and safe config rollout. Second, many backbone circuits needed to be turned up to meet the growing inter-DC traffic demand. However, provisioning a circuit was a time-consuming and error-prone process, involving finding unused point-to-point IPs manually and configuring them on both circuit endpoints. Not only was the team unable to grow the network capacity fast enough, many circuits were misconfigured with conflicting IPs. To automate such design changes, the team introduced new models to FBNet in which IPs and circuits were allocated using design tools based on predefined rules, and relevant config snippets were generated for deployment. Over time the suite of design tools was developed to cover different use cases, and additional templates were added for different vendors to generate vendor-specific device configs.

Looking ahead

Robotron incorporates many experiences learned from operating Facebook's production network. Without these, we would have been unable to achieve the level of success with the system we have today. The Robotron paper not only outlines the approaches we took, but also points out many of

facebook research



operational efficiencies across network operators. To facilitate this, we are in the process of open sourcing part of Robotron, e.g., `fbpush`. If you are excited about working on Robotron, apply to our Software Engineer, Network position!

Acknowledgment

Thanks to many members in NetEng, Edge and Network Services, and Net Systems teams at Facebook for making Robotron a reality.

Areas:



Systems & Networking

Like

Share

Related Content

Blog

Announcing the winners of the Networking Systems request for proposals

September 25, 2019



Blog

Facebook announces new research awards at AI Systems Faculty Summit

September 24, 2019



Blog

Registration now open for the 2019 Probability and Programming Research Workshop, Big Code Summit, and PLEMM

July 16, 2019



Blog

Facebook hosts 2019 Networking and Communications Faculty Summit

<https://research.fb.com/robotron-top-down-network-management-at-facebook-scale/>



facebook research



Follow Facebook Research



Visit Our Other Blogs

code.fb.com

facebook.ai

oculus.com/blog

[RSS Feed](#)

[About](#)

[Careers](#)

[Privacy](#)

[Cookies](#)

[Terms](#)

[Help](#)

Facebook © 2019