PROGRAMMABLE NETWORKING FABRICS FOR CORE AND EDGE

Highlights

Software based data center fabric that manages leaf/spine switches and controllers

- Autonomous Leaf/Spine/UPF switch management using a combination of centralized and distributed control functions
- Fabric Controller for:
  - Switch topology management of Leaf, Spine, UPF and controllers
  - Virtual Fabric creation, resource allocation and modification
  - External Management Interfaces

Programmable software-based data center networking fabric

- Industry standard P4 programming language
- Avoids vendor lock-in & eliminates the need to wait for silicon upgrades
- Allows for customer programmability
- Allows developers to develop new code and drive innovation

Fully Automated Fabric

- Advanced self-forming & self-discovery fabric
  - Automated discovery of incorrect network topology & cabling mistakes
- Zero-touch provisioning of the virtual networking and virtual components
  - Zero-touch fabric with minimal to no human intervention
- Provisions in minutes vs. hours/days with traditional solutions
- Automated software upgrades
- True network automation with autonomous leaf and spine switches
- Self-healing (automated remediation)

Scalable for distributed edge, central office, virtual central office, CORD, and branch office applications

- Native IPv6, multi POD, cloud and multi data center solutions
- Optimized solution for edge data centers running 4G and 5G applications
- Embedded 5G User Plane Function (UPF)

Integrated virtual network components

- vRouter, vSwitch and vGW (VXLAN)
- Reduced complexity
- Improved TCO

Optimized for virtual environments

- Provides support for both VM and container-based workloads

Continued

Software Defined Fabric™

The Software Defined Fabric™ is a fully automated, industrialized software solution for networking whiteboxes, and designed specifically with the vision of virtualizing the data center. Designed for hyperscale, as well as distributed data centers, the solution is targeted to clients in the Data Center Operator, Telco, Enterprise, Cloud, and Gaming industries. As a fully programmable fabric, it allows advanced networks functions to be deployed at scale with maximum performance and lowest cost. Software Defined Fabric brings forth a new paradigm in networking, leveraging automation, programmability, open networking standards and white box hardware. The solution offers the following key high-level benefits:

- A high performance and adaptable fabric architecture that provides a significant increase in throughput and power efficiency, and reduction in latency
- Distributed Kubernetes/OpenShift based control plane for better resiliency
- Future-proof; programmable control and data planes
- Operational consistency; uses same standard Linux distribution as that used by compute and storage nodes in the data center and is managed as a "server node" to enable unified OS updates of the data center
- Enhanced support for network "slicing", enabling allocation of physical resources into multiple autonomous and secure isolated domains (vFabrics)
- Optimized platform for VM and Container based technologies and workloads
- A native IPv6 based networking environment fully capable of supporting both IPv6 and IPv4 traffic
- Fully automated fabric with advanced self-forming and self-discovery capabilities, zero-touch provisioning of the virtual networking and virtual components with automated software updates

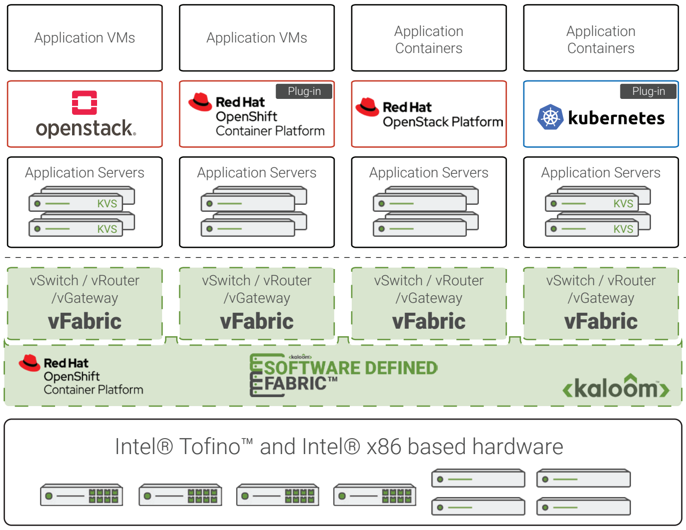

Figure 1: Software Defined Fabric - High Level Block Diagram

Advanced monitoring and segment analytics capabilities built-in

- Enables programmable in/out-of-band telemetry per flow/packet

Multivendor solution by design with no vendor lock-in

- Support for networking white boxes from multiple vendors such as Accton/Edgecore and Inventec
- Support for Point of Delivery (POD) modules from different vendors
- Easy integration into existing solutions

Fabric Virtualization

- Provides full support for network virtualization with native 5G network slicing
- Enables allocation of physical resources into multiple autonomous isolated network slices, called vfabrics
- Enables the assignment of vfabrics to different virtual DC operators
- Enables DC operators to faster provision new customers in software
- Provides full isolation between customers/tenants

Open Networking and APIs

- Standard Linux Based
  - No kernel patches
- Open APIs
  - NETCONF API and YANG modeling
- Orchestration agnostic
  - Plugins for OpenStack, K8s, and OpenShift
- No vendor lock-in
  - Networking white box friendly
- Open-source friendly
  - Contributing improvements upstream to Linux and K8s

Significant x86 resource utilization improvements

- Delivers power efficiencies and savings
- Reduces number of CPU cores used for networking and frees them up for payload tasks
- Delivers lower latency and higher performance by offloading and improving NIC functionality into the fabric with the Kaloom Virtual Switch

Lower Latency

- Provides significantly lower server and NIC latency when using the Kaloom Virtual Switch
- Improves virtual end to end latency with advanced service chaining capabilities

Standard Linux (Red Hat CoreOS)

- No hacked and/or out-of-date Linux kernels which guarantees faster updates and security fixes
- Allows standardizing on the same OS for compute, storage and networking
- Leverages security-enhanced Linux (SELinux), control groups (cgroups), and kernel namespaces to provide military-grade security
- Lightweight OS that provides the flexible & modular capabilities of Linux containers
- Standard upstream kernels without any patches
- Same s/w orchestration model and management tools across networking, compute and storage

For more detailed information please refer to *https://www.kaloom.com/product-collateral*

Software Defined Fabric is built on a 3-stage folded CLOS and fully non-blocking topology (see figure below) also referred to as a Leaf and Spine architecture where "Spine" switches represents the middle stage of the CLOS topology and "Leaf" switches represent the input and output stages.

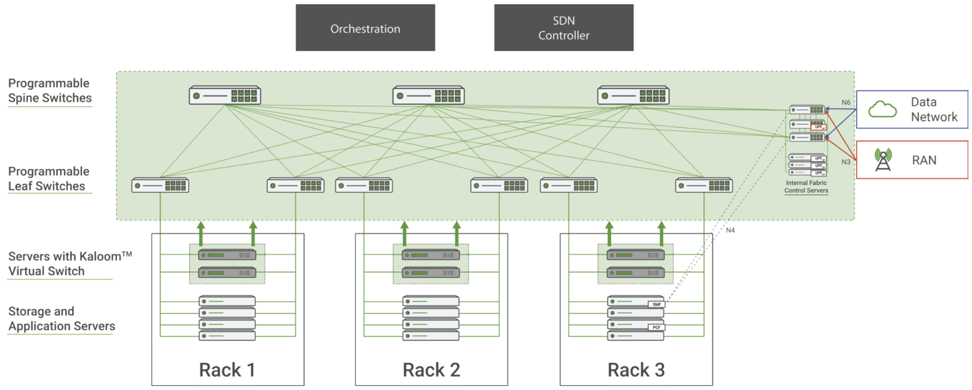

Figure 2: Software Defined Fabric™ Topology

Figure 2 above illustrates several application servers connected to the leaf switches, with each server being connected to two separate leaf switches for redundancy. Leaf switches may have multiple paths to every server and connect to all the spine switches for maximum connectivity options. The Software Defined Fabric is controlled by a cluster of Controllers together with distributed control plane functionality in each network node, to create a distributed Kubernetes based control plane for the fabric. The controllers have one serving as the Master Controller to control quorum and the remaining nodes used to deploy networking services (vRouters, etc.).

Figure 3 below depicts the physical view of a typical Pod deploying the Software Defined Fabric as a networking solution and contains the following components:

Spine Switches – Connects all the leaf switches. Each leaf switch is connected to all spine switches in the upper layer.

UPF Switches – Provides the 4G and 5G User Plane Functionality in a terabit capable switch. 3GPP standards compliant it is ready to be deployed in multi-vendor environments.

Edge Leaf Switches – Provide the interface to exit the Software Defined Fabric and forward the user data traffic towards the Wide Area Network (WAN). They can be co-located with other Leaf Switches.

Application Servers – Connect to the Software Defined Fabric leaf nodes and run various data center workloads.

Fabric Controller – Runs in controller servers, dual-purpose leaf-controller or UPF-controller switches to manageand control all the nodes of the Software Defined Fabric

Leaf Switches – Connects the application servers via one or multiple paths.

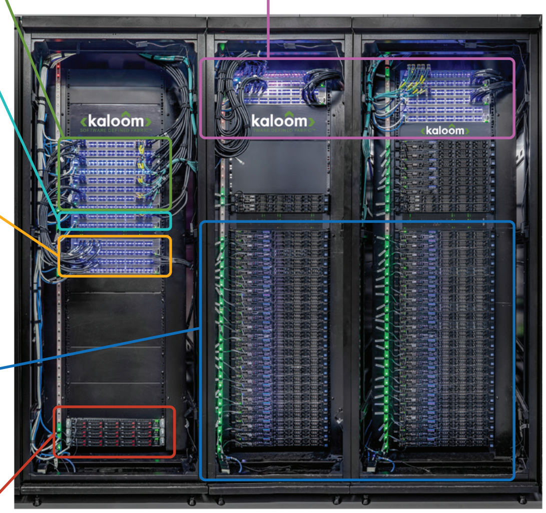

Figure 3: Software Defined Fabric Physical View

In the application server racks, each end-user server used for storage or computing applications is connected to two separate leaf switches for redundancy. The Pod can be expanded by adding several new application server racks connected to the fabric through additional leaf switches. The Leaf, Spine and UPF Switches are based on the Intel Tofino chipset (previously Barefoot Networks). The Intel Tofino device is fully programmable using the P4 Programming language. The UPF switches are an optional component of a typical Pod.

Cloud Edge Fabric™

The Cloud Edge Fabric seamlessly builds on the Software Defined Fabric solution for large scale centralized or regional data centers by expanding the range to include:

- Distributed Edge
- Central Office
- Virtual Central Office
- CORD
- Branch Office Applications

Figure 4 below shows the cloud edge DC configuration comprised of the Cloud Edge Fabric that sits in between the base stations and regional DCs. Base Stations are typically connected to a nearby Cloud Edge Data Center via an Optical Backhaul. These Cloud Edge DCs are connected to Regional Data Centers. Regional Data Centers can be connected to National/Central Data Centers. Regional and Central Data Centers can have optimized data connections to public clouds such as AWS, Azure, GCP, etc.

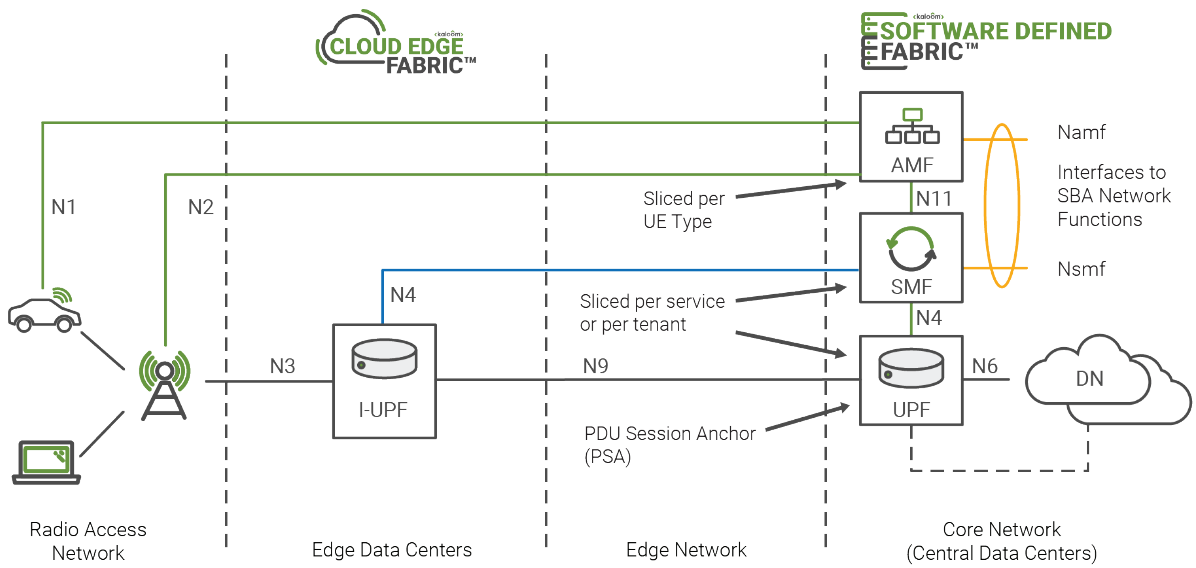

Figure 4: Cloud Edge Fabric™ and Software Defined Fabric™ Implementation

Note also Kaloom's UPF implementation within the overall 5G Mobile Packet Core as seen in Figure 5. A 5G EPC comprises several control and data plane nodes such as UPF, SMF, AMF, etc. Kaloom provides the networking environment to support the execution of these control and data plane nodes as either Virtualized Network Functions (VNFs) or Cloud Native Network Functions (CNFs). The Kaloom UPF is based on 3GPP specifications. It is designed to be deployed in a multi-vendor environment and is compatible with SMF, NRF, RAN and UPF nodes from other vendors. The Kaloom UPF can either be deployed as a stand-alone component or as an integrated network function inside a Kaloom Cloud Edge Fabric.

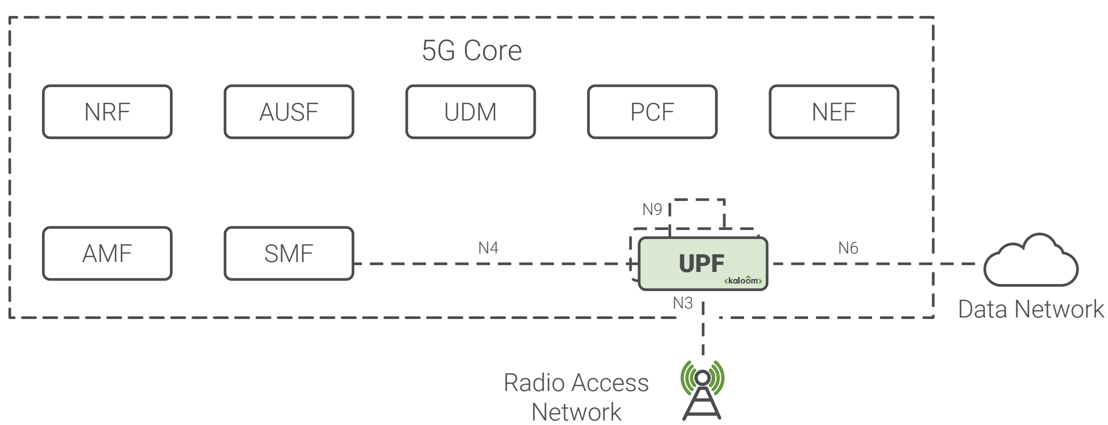

Figure 5: Kaloom's UPF Node Within the 5G Core

Cloud Edge Fabric, being a scaled down Software Defined Fabric, does not require Spine switches and uses a meshed Leaf switch topology.

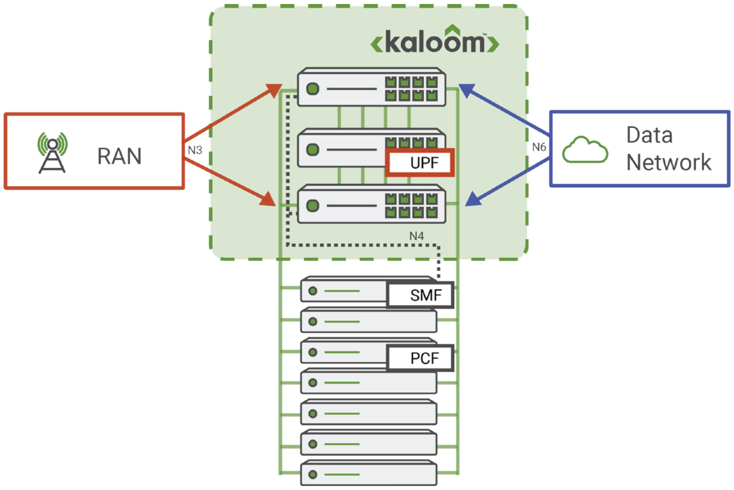

Figure 6: Logical UPF Implementation Overview

Figure 6 provides a conceptual and high-level architectural overview illustrating Kaloom's logical UPF implementation that consists of distributed functionality within a Cloud Edge Fabric. Incoming packets from the RAN or the Data Network are shown arriving on the edge switches of the fabric upon which a load balancing function distributes the packets to the various Leaf Switches.

The UPF control plane functions as the N4 reference point towards the SMF. The UPF control plane scales according to the capacity and number of physical fabric nodes used as controllers. The UPF data plane runs in a dedicated switch. The more UPF switches that are deployed in an edge fabric, the higher the resulting capacity of the UPF.

Network Slicing

Cloud Edge Fabric, and Software Defined Fabric, natively supports 5G network slicing where by an edge data center can be partitioned by a Data Center Infrastructure Provider into multiple independent virtual data centers, with each virtual data center being provided its own virtual fabric called a vFabric. Each virtual data center with an associated vFabric can be assigned to a different virtual Data Center Operator or Cloud Service Provider that can offer differing SLAs per cloud service user. In this regard, slicing permits multiple operators and large enterprises to share, as Cloud Service Users (CSUs), a common distributed cloud infrastructure, with each CSU enjoying full isolation down to the hardware level for better security and a better quality of experience provided to the individual Cloud Service Consumer consuming the services/applications offered by the CSU.

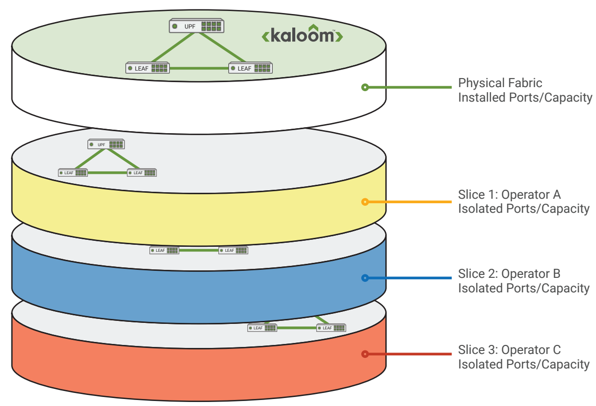

Figure 7: Network Slicing

Figure 7 above illustrates the concept of network slicing. In Kaloom's implementation, a slice corresponds to one vfabric which corresponds to a single logical UPF. Each vfabric provides a fully distributed and isolated domain, thereby making it possible to deploy isolated packet cores for either specific applications or large-scale and strategic customers.

Supported Standards

The following information in the table below provides a list of the standards to which Kaloom is compliant.

Standards	
IEEE	• IEEE 802.1 AB Link Layer Discovery Protocol • IEEE 802.1Q VLAN Tagging • IEEE 802.3ad Link Aggregation with LACP (dynamic) • IEEE 802.3ba 40 and 100 Gigabit Ethernet Architecture • IEEE 802.3bm 40 and 100 Gigabit Ethernet Physical layer • IEEE 802.3ab 1 Gigabit Ethernet • IEEE 802.3ae 10 Gigabit Ethernet • IEEE 802.3by 25 Gigabit Ethernet • IEEE 802.3cd 50 Gigabit Ethernet • IEEE 802.3bs 200 and 400 Gigabit Ethernet
IETF RFCs	• RFC 826 Address Resolution Protocol • RFC 1997 BGP Communities Attribute • RFC 2131 Dynamic Host Configuration Protocol • RFC 2328 OSPF Version 2 (UNH certification) • RFC 2385 Protection of BGP Sessions via the TCP MD5 Signature Option • RFC 2464 Transmission of IPv6 Packets over Ethernet Networks • RFC 2796 BGP Route Reflection An alternative to full mesh IBGP • RFC 2842 Capabilities Advertisement with BGP-4 • RFC 3046 DHCP Relay Agent Information Option • RFC 3065 Autonomous System Confederations for BGP • RFC 3101 The OSPF Not-So-Stubby Area (NSSA) Option • RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) • RFC 4271 A Border Gateway Protocol 4 (BGP-4) (UNH certification) • RFC 4291 IP Version 6 Addressing Architecture • RFC 4443 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification • RFC 4511 Lightweight Directory Access Protocol (LDAP) • RFC 4760 Multiprotocol Extensions for BGP-4 • RFC 4861 Neighbor Discovery for IP version 6 (IPv6) • RFC 4862 IPv6 Stateless Address Autoconfiguration • RFC 4941 Privacy Extensions for Stateless Address Autoconfiguration in IPv6 • RFC 5277 NETCONF Event Notifications • RFC 5340 OSPF for IPv6 • RFC 5880 Bidirectional Forwarding Detection (BFD) • RFC 5881 Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop) • RFC 5883 Bidirectional Forwarding Detection (BFD) for Multihop Paths • RFC 6022 YANG Module for NETCONF monitoring • RFC 6241 Network Configuration Protocol (NETCONF) • RFC 6242 NETCONF over SSH • RFC 6991 Common YANG Data Types • RFC 7348 Virtual eXtensible Local Area Network (VXLAN) • RFC 7432 BGP MPLS-Based Ethernet VPN • RFC 7950 The YANG 1.1 Data Modeling Language (Netconf/Yang data model published with every SWrelease) • RFC 8040 RESTCONF Protocol • RFC 8106 IPv6 Router Advertisement Options for DNS Configuration • RFC 8200 Internet Protocol, Version 6 (IPv6) Specification • RFC 8201 Path MTU Discovery for IP version 6 • RFC 8341 Network Configuration Access Control Model • RFC 8342 Network Management Datastore Architecture (NMDA) • RFC 8343 A YANG Model for Interface Management • RFC 8345 A YANG Data Model for Network Topologies • RFC 8349 A YANG Data Model for Routing Management (NMDA Version) • RFC 8365 Ethernet VPN (EVPN) as a Network Virtualization Overlay (NVO) solution • RFC 8415 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) • RFC 8525 YANG Library • RFC 8528 YANG Schema Mount • RFC 9127 YANG Data Model for Bidirectional Forwarding Detection (BFD)

3GPP Release 15 - 5G UPF	‣ 3GPP TR 29.891 ‣ 3GPP TS 23.214 ‣ 3GPP TS 29.281 ‣ 3GPP TS 29.244 ‣ 3GPP TS 23.503 ‣ 3GPP TS 23.502 ‣ 3GPP TS 23.501 ‣ 3GPP TS 23.007 ‣ 3GPP TS 23.527 ‣ 3GPP TS 28.554 ‣ 3GPP TS 29.561 ‣ 3GPP TS 33.501 ‣ 3GPP TS 38.415

Standardization Bodies Participation

Kaloom is an active participant in the following standardization bodies.

- IETF
- P4 Consortium
- Linux Foundation
- Linux Foundation Edge
- Linux Foundation Networking (Founding Member)
- ONAP
- P4 Consortium
- OPNFV
- CNTT
- Cloud Native Computing Foundation (CNCF)
- Open Networking Foundation (ONF)
- Broadband Forum (BBF)
- FRR

 Document Version - 1.2 Publication Date: 04.07.2022