Carrier solution: BNG Gateway

In cooperation with HPE, Intel and the Beijing Research Institute (BRI) of China Telecom, BISDN has built a virtual Broadband Remote Access Server (vBRAS) prototype, based on the CUPS approach (Control User Plane Separation, split BRAS architecture) [1].

For that prototype, it was required that a standard x86 server with 2 CPU sockets should be able to process 100Gbit/s. The controller was required to process traffic coming from both PPPoE and IPoE.

Intel integrated the FPGA directly on the network card (XL710) in its prototype of a programmable network card. This allowed a server to be equipped with 3x40G cards, i.e. 120Gbit/s. It was to be shown whether and, if yes, with which packet size the virtualized BRAS could achieve that throughput.

Split BRAS architecture

The implementation selected was based on a split architecture, which processed control and data traffic on different machines (CUPS). In addition, all mayor components were based on Open Source software, and standard hardware was used. The architecture is depicted in Figure 1.

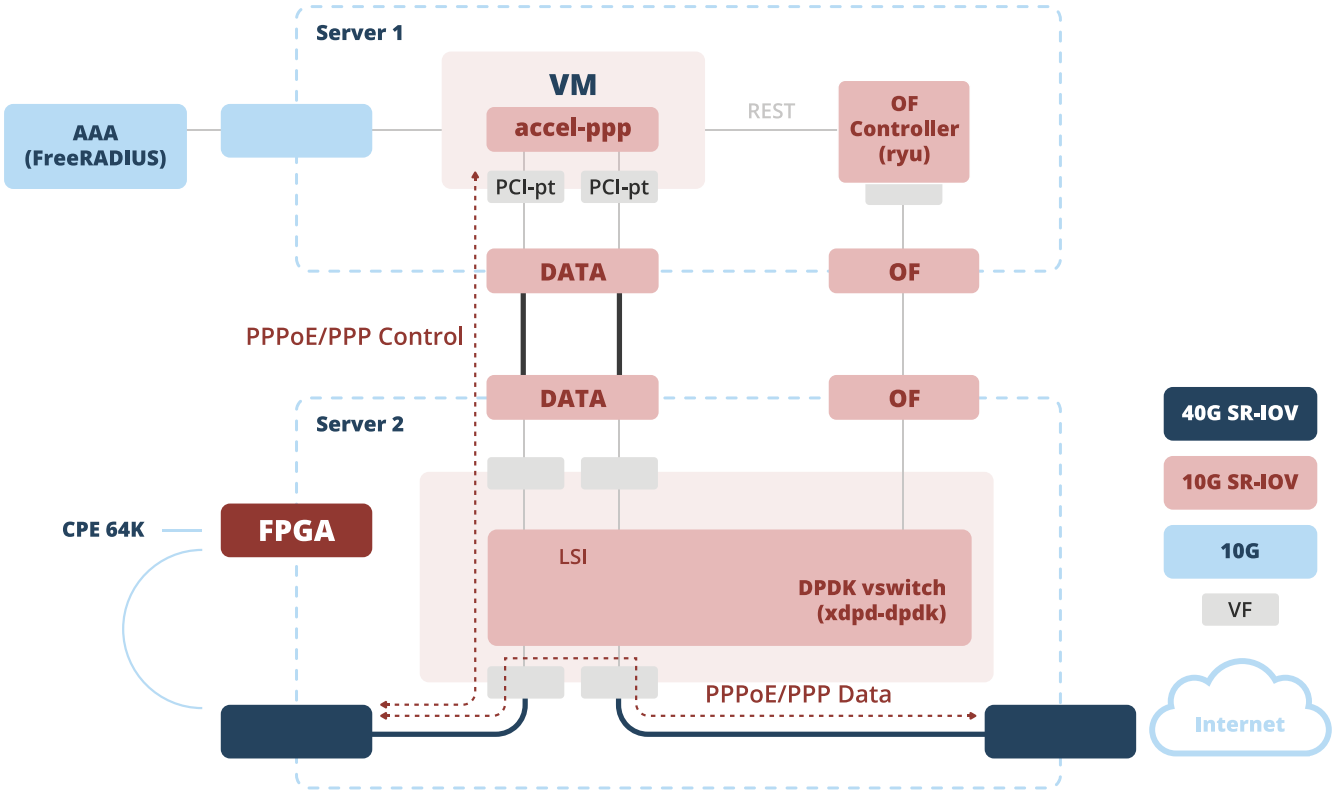

Figure 1: BRAS architecture with split control and data plane

[1] DPDK – Data Plane Developer Kit – Intel's SDK for forwarding acceleration

BISDN

The main software components of the solution are the programs “accel-ppp”, which realizes the core of control functions, and “xdpd-dpdk”, which is a high-performance soft switch for the data path. A ryu controller controlled the switch via OpenFlow, while a radius server is acting as an authentification backend. On the hardware side, a specific FPGA card with an Altera FPGA (http://www.nallatech.com/store/pcie-accelerator-cards/nallatech-385a-arria10-1150-fpga/) was used together with standard hardware, since the network card is first in line and provides the necessary QoS functions. The card was a prototype developed by a large network card vendor. The solution uses the software components listed in Table 1.

Extensions of existing software

The following existing software components were used and amended:

- accel-ppp (a REST interface was added to connect to ryu-controller)
- xdpd (adaptation of the software to dpdk 17.02 and adaptation of the matching algorithms to trie match), use of RSS (Receive Side Scaling)
- Integration of an FPGA board in xdpd-dpdk (driver integration)
- VM-based test environment

In particular the use of optimized matching algorithms and the so-called Receive Side Scaling (RSS) resulted in substantially improved speed, compared to the original solution. In addition, the memory was optimized in such a way that the cache can be used more efficiently.

In total a data rate of 106Gbit/s for 512byte packets could be achieved. This number could be demonstrated for IPoE, whereas for PPPoE/PPP values slightly above 80Gbit/s could be shown. The results were exhibited during the Mobile World Congress in Shanghai in June 2017 at the HPE booth.

Name	Function	Source
accel-ppp	High performance PPTP/L2TP/PPPoE/IPoE server for Linux	https://sourceforge.net/projects/accel-ppp/
ryu	Components-based Software Defined Networking framework	https://osrg.github.io/ryu/
docker.io	Core of the Docker platform is a lightweight container runtime that contains the orchestration	https://www.docker.com/
xdpd	OpenFlow switch (multi-platform) – the OpenFlow eXtensible DataPath daemon	https://github.com/bisdn/xdpd
dpdk	DPDK is a set of libraries and drivers for fast packet processing	http://dpdk.org/
python	Programming language	https://www.python.org/
boost	C++ source libraries	http://www.boost.org/

Table 1: Open Source software components

[1] Finding an Efficient Virtual Network Function Architecture for Next-Generation Telecommunications Infrastructure: https://www.sdxcentral.com/articles/featured/hpe-efficient-vnf-architecture/2017/09/