PPPoE Demo for Split Data Plane Performance using P4 for ISPs

Abstract

P4 is a powerful domain-specific language for implementing high-speed packet processing applications. It is a target independent language that facilitates deploying P4 and non-P4 programmable components and their interfaces on a large variety of target platforms, e.g., CPUs, FPGAs, ASICs, SOCs (system-onchip), and network processors. However, real world applications require lot more than a control plane, data plane, and architecture descriptions expressed in P4 language. xFlow Research is building a highly scalable PPPoE solution using off-the-shelf Linux computers in P4. It is implementing a 4-layer application architecture comprising of: a) P4 data plane, b) P4 control plane, c) non-P4 application client software, and d) non-P4 application server software.

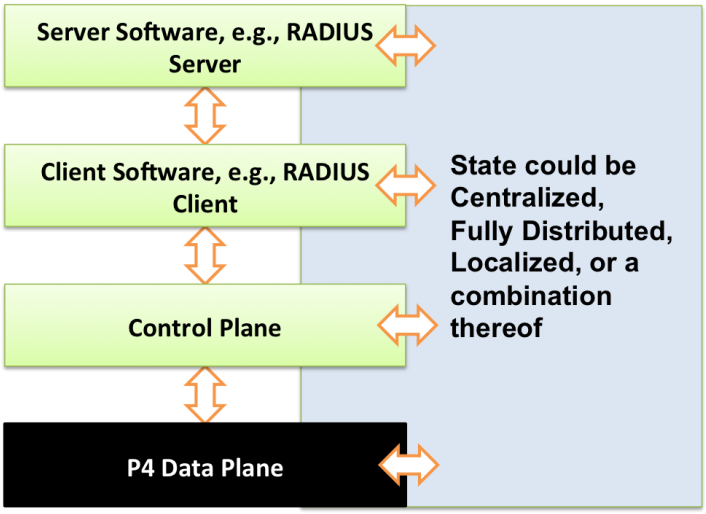
Building efficient interfaces between these layers to guarantee performance, scalability, high availability, and fast recovery from component failures is a big challenge. An equally important concern involves balancing a large ISP's workload using a cluster of stateful P4 and non-P4 components communicating and distributing state information between these components including the data about individual subscribers' virtual interfaces, lookup tables, buffers/queues for ensuring policy-based upload and download bandwidth capacity, and other data related to the monitoring, management and orchestration of all P4 and non-P4 components.

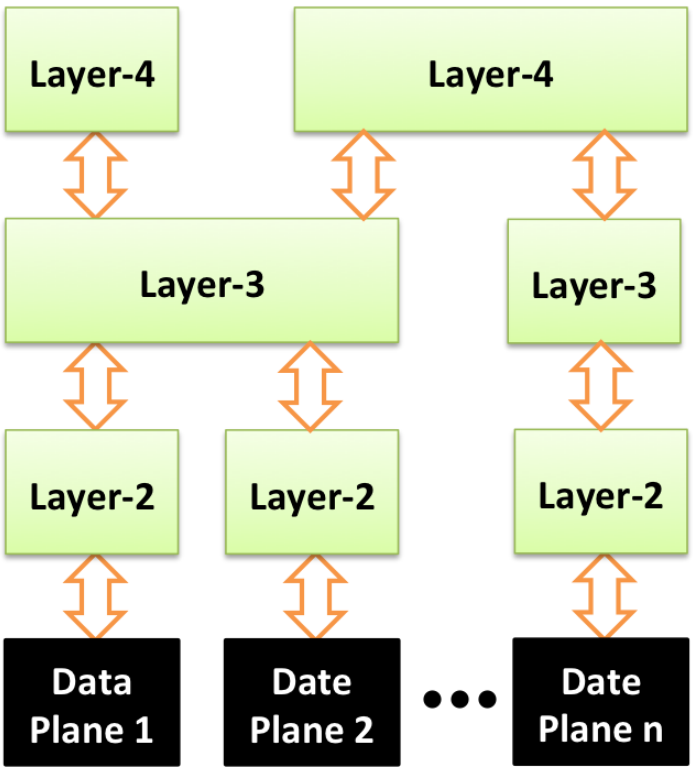
xFlow Research is using P4 to build its load balancers in addition to using it for allocating IP addresses and forwarding packets. xFlow has also researched the possibility of splitting its data plane into more than one P4 data planes, e.g., a data plane optimized for authenticating subscribers and allocating IP addresses using PPPoE/RADIUS/DIAMETER/SRC protocols, and a separate data plane optimized for faster packet forwarding between the subscribers and routers connecting them with the Internet backbone.

Huniya Arif, Shabbir Khan, Ashok Malani

Architecture and Topology:

Shown in the figure below is the 4 layer architecture and the split data plane topology:



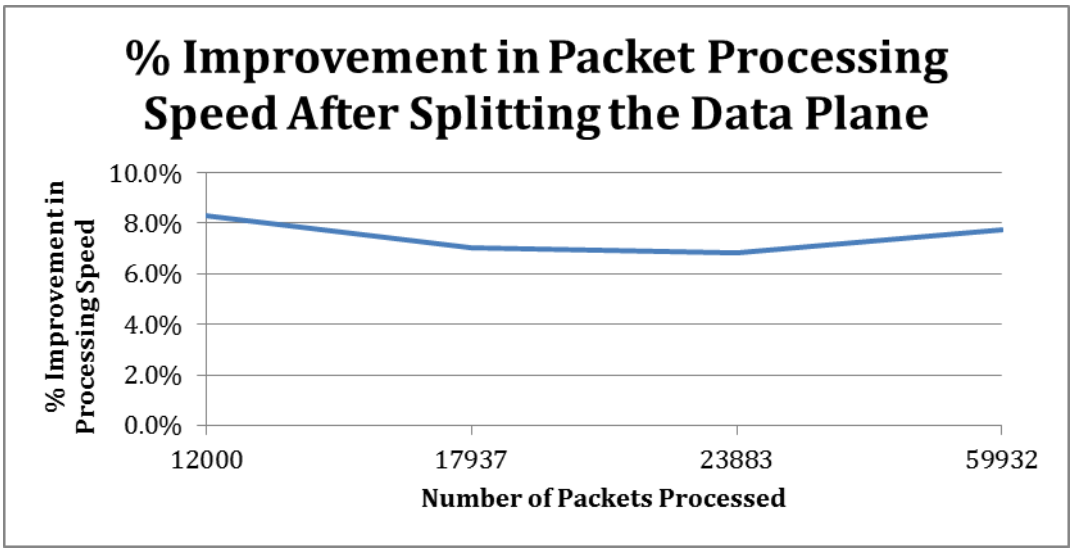


Results:

As shown below we see an 7%-8% improvement in packet processing, when using the split data plane.

System Configuration:

Hardware: CPU: 2.4GHz , RAM: 16GB , Core: i7, System processor: 64-bit, HardDrive: 1000GB.
Software: VMWare Workstation Player version 12., Mininet 2.2.1, P4 behavioral-model version 2 (aka bmv2)



Number of packets	Single Data Plane Packet Processing Speed (ms)	Split Data Plane Packet Processing Speed Latency (ms)	% Improvement in Packet Processing Speed due to split Data Plane
30	232.67	220.50	5.2%
60	276.68	241.48	12.7%
90	263.58	228.55	13.3%
120	223.98	212.01	5.3%
150	248.70	244.15	1.8%
180	245.82	240.83	2.0%
210	231.93	204.90	11.7%
240	244.94	220.88	9.8%
270	230.00	216.27	6.0%
300	234.43	214.25	8.6%
600	460.69	429.88	6.7%
12,000	272.99	250.34	8.3%
17,937	301.60	280.41	7.0%
23,883	281.84	262.58	6.8%
59,932	264.36	243.85	7.8%

Number of packets	Single Data Plane Average Latency (ms)	Split Data Plane Average Latency (ms)
30	6,980	6,615
60	16,601	14,489
90	23,722	20,570
120	26,877	25,441
150	37,305	36,623
180	44,247	43,349
210	48,705	43,028
240	58,785	53,012
270	62,099	58,393
300	70,328	64,276
600	276,411	257,926
12,000	3,275,861	3,004,129
17,937	5,409,858	5,029,730
23,883	6,731,254	6,271,139
59,932	15,843,742	14,614,597
119,476	38,559,911	39,078,452