Offloading Media Traffic to Programmable Data Plane Switches

ABSTRACT

According to estimations, approximately 80% of Internet traffic represents media traffic. Much of it is generated by end users communicating with each other (e.g., voice, video sessions). A key element that permits the communication of users that may be behind Network Address Translation (NAT) is the relay server. This project presents a scheme for offloading media traffic from relay servers to programmable switches. The proposed scheme relies on the capability of a P4 switch with a customized parser to de-encapsulate and process packets carrying media traffic. The switch then applies multiple switch-actions over the packets. As these actions are simple and collectively emulate a relay server, the scheme is capable of moving relay functionality to the data plane operating at terabits per second. Performance evaluations show that the proposed scheme not only produces optimal results regarding Quality of Service (QoS) parameters (no packet loss, minimum delay, negligible delay variation, high Mean Opinion Score) but also scales much better than current solutions. Evaluations conducted on a Tofino chip with up to 35Gbps of media traffic or its equivalent of 400,000 simultaneous G.711 media sessions (limited only by the traffic generator rather than by the switch) show an ideal operation of the switch-based solution (using ≈1% of the switching capacity). In contrast, a relay server with a modern CPU model used for evaluations can process up to 900 simultaneous G.711 media sessions.

1 SYSTEM DESIGN

The proposed architecture uses a programmable switch to emulate the behavior of the relay server, which must: 1) parse the incoming packet carrying media traffic from the first party, say user A; 2) identify the session this packet belongs to by using the 5-tuple {source IP, source port, destination IP, destination port, protocol}. The destination IP and the destination port in the incoming packet refer to a local socket in the relay server; 3) replace the source IP with that of the relay server and the source port with that used by the relay server to receive traffic from user B; 4) replace the destination IP and the destination port with the NAT-translated IP and port corresponding to user B; 5) recalculate both IPv4 and UDP checksums, and 6) forward the packet to user B. These steps can be implemented with match-action pipelines available at programmable switches.

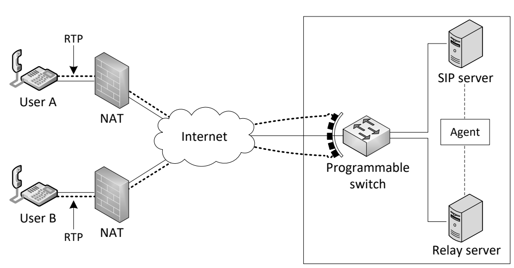

Figure 1: High-level system architecture.

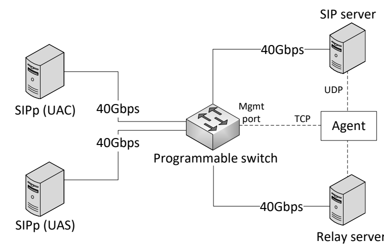

Figure 2: Evaluation topology.

A custom software (agent) learns the ports allocated to a media session by the relay server. It uses the 5-tuple allocated to the media session to construct a unique session identifier, then stores identifiers of the media sessions and the new headers' values in the switch It also clears media sessions allocated in the switch when a call is teared down.

2 EVALUATION

The topology used to conduct experiments is shown in Fig. 2. The following components are used for the experiment: 1) OpenSIPS, an open source implementation of a SIP server. 2) RTPProxy, a high-performance relay server for RTP streams. 3) SIPp, an open source SIP traffic generator that can establish multiple concurrent sessions and generate media (RTP) traffic. 4) Iperf, traffic generator used to generate background UDP traffic. Edgecore Wedge100BF-32X is the programmable switch used in the experiment. Fig. 3 depicts the results obtained when executing the experiment for 900 simultaneous sessions with both scenarios (switch-based relay and server-based relay). The CPU of the relay server is also measured. In the server-based relay scenario, the relay server consumes a significant portion of the CPU. The relay server's CPU in the switch-based relay decreases to approximately zero once sessions stop arriving.

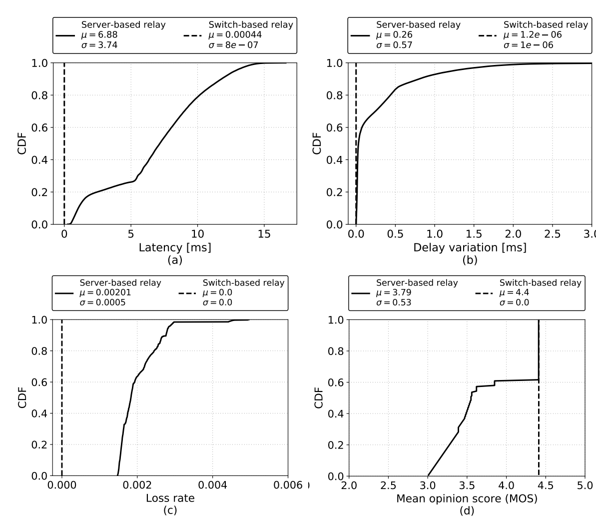

Figure 3: CDF of delay (a), delay variation (b), loss rate (c), and (d) Mean Opinion Score (MOS) in both scenarios.

REFERENCES

[1] E. Kfoury, J. Crichigno, E. Bou-Harb, Vladimir Gurevich, "Offloading Media Traffic to Programmable Data Plane Switches", IEEE International Conference on Communications (ICC), Dublin, Ireland, June 2020.