Offloading Media Traffic to P4 Programmable Data Plane Switches

Elie Kfoury¹, Jorge Crichigno¹, Elias Bou-Harb², Vladimir Gurevich³

¹University of South Carolina, Columbia, SC
²University of Texas, San Antonio, TX
³Barefoot Networks, an Intel Company

CI Lab @ UofSC: http://ce.sc.edu/cyberinfra

CI Engineering Brown Bag
Friday, February 28, 2020
• Introduction

• Background Information
  ➢ Session Initiation Protocol (SIP) and Real Time Protocol (RTP)
  ➢ Network Address Translation (NAT) traversal problem
  ➢ P4 switches

• Proposed solution

• Evaluation

• Lessons learned
INTRODUCTION

• According to estimations, media traffic represents approximately 80% of the total traffic over the Internet\(^1\)

• Much media traffic is generated by end users communicating with each other

• Media services (voice, video) running alongside the data network in campuses are becoming standard

---

VOICE AND VIDEO

• Conversational Voice- and Video-over-IP are widely used today
  - Open and proprietary (Skype, WhatsApp) solutions

• Supporting protocols are divided into two main categories
  - Session control protocols (signaling)
    ✓ Session Initiation Protocol (SIP)
    ✓ Establish and manage the session
  - Media protocols (media)
    ✓ Real Time Protocol (RTP)
    ✓ Transfer audio and video streams between the end-users

• Desirable Quality-of-Service (QoS) characteristics
  - Delay- and jitter-sensitive, low values
  - Occasional losses are tolerated
SIP initiates, maintains, and terminates multimedia sessions between endpoints
- User agent client (UAC)
- User agent server (UAC)

RTP transports real-time data, such as audio and video
NETWORK ADDRESS TRANSLATION

- Network Address Translation (NAT)
  - Maps ports, private IP addresses to public IP addresses
- Used in campus / enterprise networks, operators
- NAT introduces various issues
  - Violation of the end-to-end principle
  - Traversal of end-to-end sessions

---

NAT prevents a user from outside from initiating a session
If both users have NATs, then neither can accept a call
- IP translation is recorded by a SIP registrar server
SIP carries the IP addresses and ports to be used by RTP to send/receive media
- NAT-translated IP, ports are unknown until RTP traffic starts
Several solutions proposed for NAT traversal
- STUN - RFC 5389\(^1\), TURN - RFC 7566\(^2\), ICE - RFC 8445\(^3\)

---

• Intermediary device

<table>
<thead>
<tr>
<th>SIP server</th>
<th>Relay server</th>
<th>RTP Information at relay server</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Device IP - port</th>
<th>Allocated IP - port</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>
• Intermediary device
• SIP establishes the session
  ➢ RTP ports are unknown
  ➢ The relay server allocates one port for each device

## RTP Information at relay server

<table>
<thead>
<tr>
<th>Device</th>
<th>Allocated IP - port</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>IP_R - p_{R-A}</td>
</tr>
<tr>
<td>B</td>
<td>IP_R - p_{R-B}</td>
</tr>
</tbody>
</table>

---

[Diagram showing SIP server, relay server, and RTP information]
• Intermediary device
• SIP establishes the session
  ➢ RTP ports are unknown
  ➢ The relay server allocates one port for each device
• The relay server receives and relays the RTP traffic
Intermediary device
SIP establishes the session
- RTP ports are unknown
- The relay server allocates one port for each device
The relay server receives and relays the RTP traffic

RTP Information at relay server

<table>
<thead>
<tr>
<th>Device IP - port</th>
<th>Allocated IP - port</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP&lt;sub&gt;A&lt;/sub&gt; - p&lt;sub&gt;A'&lt;/sub&gt;</td>
<td>IP&lt;sub&gt;R&lt;/sub&gt; - p&lt;sub&gt;R-A&lt;/sub&gt;</td>
</tr>
<tr>
<td>IP&lt;sub&gt;B&lt;/sub&gt; - p&lt;sub&gt;B'&lt;/sub&gt;</td>
<td>IP&lt;sub&gt;R&lt;/sub&gt; - p&lt;sub&gt;R-B&lt;/sub&gt;</td>
</tr>
</tbody>
</table>
OVERVIEW P4 SWITCHES

• P4 switches permit programmer to program the data plane
• Add proprietary features
  ➢ Parse packet headers, including UDP
  ➢ Header inspection; identify media session using the 5-tuple
  ➢ Modify fields; IP addresses and ports

```
136  /******************************************************************************
137  * PARSER  **************************************************************************/
138  /******************************************************************************
139
140  state parse_ethernet {
141  packet.extract(hdr.ethernet);
142  transition select(hdr.ethernet.etherType) {
143    TYPE_IPV4: parse_ipv4;
144    default: accept;
145  }
146  }
147
148  state parse_ipv4 {
149  packet.extract(hdr.ipv4);
150  verify(hdr.ipv4.ihl >= 5, error.IPHeaderTooShort);
151  transition select(hdr.ipv4.ihl) {
152    5: accept;
153    default: parse_ipv4_option;
154  }
155  }
```

P4 code
Several programmable switches implement the Protocol Independent Switch Architecture (PISA)

- Abstract processing model
- Programmers specify how a packet should be parsed and processed through match-action tables

If the P4 program compiles, it runs on the chip at line rate
The proposed architecture uses programmable switches to emulate the behavior of the relay server:

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
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 A
4. Replace the destination IP and the destination port with those of user B
5. Recalculate both IPv4 and UDP checksums
6. Forward the packet to user B
A custom software (agent) learns the ports allocated to a media session by the relay server.

The Rule Generator uses the 5-tuple allocated to the media session to construct a unique session identifier.

It 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.
IMPLEMENTATION AND EVALUATION

- System components
  - OpenSIPS, an open source implementation of a SIP server
  - RTPProxy, a high-performance relay server for RTP streams
  - SIPp: an open source SIP traffic generator that can establish multiple concurrent sessions and generate media (RTP) traffic
  - Iperf3: traffic generator used to generate background UDP traffic
  - Edgecore Wedge100BF-32X: programmable switch
IMPLEMENTATION AND EVALUATION

• Two scenarios are considered:
  1. “Server-based relay”: relay server is used to relay media between end devices, without the intervention of the switch
  2. “Switch-based relay”: the switch is used to relay media

• UAC (SIPp) generates 900 media sessions
• The rate at which sessions arrive is 30 per second
• The test lasts for 300 seconds
• G.711 media encoding codec (160 bytes every 20ms)
• **Delay**: the time interval starting when a packet is received from the UAC by the switch’s ingress port and ending when the packet is forwarded by the switch’s egress port to the UAS
  - Delay contributions of the switch and the relay server

```
<table>
<thead>
<tr>
<th></th>
<th>Server-based relay</th>
<th>Switch-based relay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay Mean</td>
<td>μ = 6.88ms</td>
<td>μ = 0.00044ms</td>
</tr>
<tr>
<td>Delay Standard Deviation</td>
<td>σ = 3.74ms</td>
<td>σ = 8e−07ms</td>
</tr>
</tbody>
</table>
```

![CDF Graph](image)
• **Delay variation**: the absolute value of the difference between the delay of two consecutive packets
  ➢ Analogous to jitter, as defined by RFC 4689
RESULTS

- **Loss rate**: number of packets that fail to reach the destination
  - Calculation is based on the sequence number of the RTP header
RESULTS

- **CPU usage**: the percentage of the CPU’s capacity used by the relay server
RESULTS

- **Mean Opinion Score (MOS):** estimation of the quality of the media session
  - A reference quality indicator standardized by ITU-T
  - Maximum for G.711 is \(\sim 4.4\)

(a) 750 simultaneous sessions.

(b) 1500 simultaneous sessions.

(c) 1800 simultaneous sessions.
The prototype is implemented in two different scenarios:

- On top of the baseline switch program (switch.p4)
  - Implements various features including Layer 2/3 functionalities, ACL, QoS, etc.
- Standalone implementation

<table>
<thead>
<tr>
<th>On top of switch.p4</th>
<th>Table size</th>
<th>SRAM</th>
<th>Hash Bits</th>
<th>TCAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32,000</td>
<td>+8.45%</td>
<td>+2.7%</td>
<td>+0%</td>
</tr>
<tr>
<td></td>
<td>64,000</td>
<td>+16.2%</td>
<td>+4.6%</td>
<td>+0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standalone program</th>
<th>Table size</th>
<th>SRAM</th>
<th>Hash Bits</th>
<th>TCAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1,000,000</td>
<td>+97.84%</td>
<td>+86.4%</td>
<td>+0%</td>
</tr>
<tr>
<td></td>
<td>1,050,000</td>
<td>+107.5%</td>
<td>+89.8%</td>
<td>+0%</td>
</tr>
</tbody>
</table>

Additional hardware resources used when the solution is deployed on top of switch.p4 and as a standalone program.
LESSONS LEARNED

• Advantages of using a switch-based relay:
  ➢ Performance
    ~1,000,000 sessions vs ~1,000 sessions per core
  ➢ QoS
    Optimal QoS parameters: delay, delay variation, packet loss rate
  ➢ Flexibility
    The switch permits to modify / forward packets using non-standard fields
  ➢ Timing information
    Measuring delay and its variation on the P4 switch results in precise high-resolution timing information
  ➢ Programmer can free unused resources and customize program
    Accommodate additional sessions

• Limited resources

• Avoid complex application logic
ACKNOWLEDGEMENT

• Thanks to the National Science Foundation (NSF)!

• Activities in the CI Lab at the University of South Carolina are supported by NSF, Office of Advanced Cyberinfrastructure (OAC)
ADDITIONAL SLIDES
DESIGN REQUIREMENTS

- Quality of Service (QoS) parameters
  - Bandwidth
  - Delay
  - Jitter
  - Loss

QoS requirements; stringency of applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Bandwidth</th>
<th>Delay</th>
<th>Jitter</th>
<th>Packet Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoIP</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Video conference</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Data (e.g., file transfer)</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>

MOTIVATION

• According to estimations, media traffic represents approximately 80% of the total traffic over the Internet
  ➢ Much of it is generated by end users communicating with each other
• Media services (voice, video) running alongside the data network in campuses are becoming standard
• Wide Area Networks (WANs) connect centers, campuses
  ➢ E.g., SIP Trunk Network CenturyLink; 10,000 centers, 10,000 centers, 3 billion minutes of voice over IP (VoIP) conversations per month

SIP Trunk Network, CenturyLink

https://tinyurl.com/som38qv