CEDRO: an in-switch elephant flows rescheduling scheme for data-centers

Davide Sanvito*, Andrea Marchini+
Ilario Filippini+, Antonio Capone+

* NEC Laboratories Europe, Germany
+ DEIB, Politecnico di Milano, Italy

*Work carried out while at Politecnico di Milano
Introduction

- Data-center networks
- Equal Cost Multi-Path (ECMP)
- Elephant vs mice flows
- Latest advances in programmable network devices
  - Opportunities for network self-adaptation
  - More scalable and prompt reaction compared to CP reactive approaches
**CEDRO**

**Congested Elephant Detection and Re-routing Offloading**
- In-switch mechanism to detect and re-route large flows colliding on a same downstream path
- Based on programmable stateful data planes
- ECMP override
- Local and remote congestion scenarios w/o controller

![Diagram of network with switches and host connections showing re-routing and offloading mechanisms.](image)
Open Packet Processor (OPP)

Stateful extension to OpenFlow

Control function offloaded to the data plane

- In-network forwarding self-adaptation

Programming abstraction based on EFSM

- Flows associated to persistent context (state and data variables)
- Forwarding based on packet header and state
- State transition in the DP based on time-/packet-events or conditions

<table>
<thead>
<tr>
<th>Stateless stage</th>
<th>Stateful stage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Match</strong></td>
<td><strong>Action</strong></td>
</tr>
<tr>
<td>Flow Key</td>
<td>Flow Context</td>
</tr>
<tr>
<td>State</td>
<td>FDV[0]</td>
</tr>
<tr>
<td>*</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>C[0]</td>
</tr>
</tbody>
</table>
OPP toy example

Per-user (i.e. per-src-IP) tx bytes quota limiter

Match | Action
--- | ---
State | F\textsubscript{size} \geq F\textsubscript{max} | Pkt header | Action
\( \emptyset \) | * | * | SetState(MON), F\textsubscript{size} = \text{PKTsize}, fwd()
MON | 0 | * | F\textsubscript{size} += \text{PKTsize}, fwd()
MON | 1 | * | SetState(DROP), F\textsubscript{size} += \text{PKTsize}, drop()
DROP | * | * | drop()
CEDRO pipeline overview

- Leaf-spine topology
  - Leaf vs spine pipelines

- 2 stateless stages and 4 stateful stages
CEDRO pipeline: stateless stages

- **SHT**
  - Handle remote congestion signaling packets

- **LLT**
  - Handle traffic local to the leaf node
CEDRO pipeline: Sample & Hold

- **Signalling Handling Table (SHT)**
  - Match | Action
  - [ ] | [ ]

- **Leaf Local Table (LLT)**
  - Match | Action
  - [ ] | [ ]

- **Sampling Table (ST)**
  - Match | Action
  - [ ] | [ ]

- **Holding Table (HT)**
  - Match | Action
  - [ ] | [ ]

- **Routing Table (RT)**
  - [ ]

- **Link Utilization Table (LUT)**
  - [ ]

**ST Components**:
- **Sampling**: \( time < W^G \) & \( time \geq W^F \)
- **Holding**: \( time < W^G \) & \( time \geq W^F \)
- **Match**: \( shift \ W^F; W^F; F_{size}=PKT_{size} \)
- **Action**: \( F_{size}=PKT_{size} \)

**HT Components**:
- **Match**: \( shift \ W^F; W^F; F_{size}=PKT_{size} \)
- **Action**: \( F_{size}=PKT_{size} \)

**Routing Components**:
- **Routing Table**:
  - \( F_{size}=PKT_{size}; set \ ECN \)

**Link Utilization Components**:
- **Link Utilization Table**:
  - \( F_{size}=PKT_{size} \)
CEDRO pipeline: state-aware ECMP

Signalling Handling Table (SHT)

Leaf Local Table (LLT)

Sampling Table (ST)

Holding Table (HT)

Routing Table (RT)

Link Utilization Table (LUT)

**Match** | **Action**
---|---
---|---
---|---
---|---
---|---

**Match** | **Action**
---|---
---|---
---|---
---|---
---|---

ECN^& in_port=DP1 <drop>
ECN^ & ECMP=DP1 & DPI is congested <fwd(ECMP), Meta=ECMP>

ECN^ & in_port=DP2 <drop>
ECN^ & ECMP=DP2 & DP2 is congested <fwd(ECMP), Meta=ECMP>

ECN^ & in_port=DPn <drop>
ECN^ & ECMP=DPn & DPn is congested <fwd(ECMP), Meta=ECMP>

IDLE_TIMEOUT

<drop>
ECN^ & ECMP=DPx & DPx is congested <fwd(ECMP), Meta=ECMP>

<drop>
ECN^ & ECMP=DPx & DPx is congested <fwd(ECMP), Meta=ECMP>

<drop>
ECN^ & ECMP=DPx & DPx is congested <fwd(ECMP), Meta=ECMP>

<drop>
ECN^ & ECMP=DPx & DPx is congested <fwd(ECMP), Meta=ECMP>
CEDRO pipeline: link utilization

Signalling Handling Table (SHT)

Leaf Local Table (LLT)

Sampling Table (ST)

Holding Table (HT)

Routing Table (RT)

Link Utilization Table (LUT)
ofsoftswitch13 (BOFUSS) and Ryu
Leaf-spine (10 HPR, 10 Leaves, 5 Spines)
FCT comparison wrt ECMP
- 13k flows (DCT²Gen)
Memory requirements

### Analytical analysis

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Flow Table (FT)</th>
<th>Flow Context Table (FCT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHT</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>LLT</td>
<td>O(HPR)</td>
<td>-</td>
</tr>
<tr>
<td>ST</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>HT</td>
<td>10</td>
<td>O(sampled flows)</td>
</tr>
<tr>
<td>RT</td>
<td>O(DP^2)</td>
<td>O(re-routed elephant flows)</td>
</tr>
<tr>
<td>LUT</td>
<td>O(DP)</td>
<td>O(DP)</td>
</tr>
</tbody>
</table>

### Experimental analysis

**Number of sampled flows** (HT's FCT size)

**Number of re-routed flows** (RT's FCT size)
Discussion

Multiple re-routings

Failures handling
- ECMP override: microflow vs macroflow
- SPIDER: in-network failure detection & recovery scheme

Deeper multi-rooted topologies
Conclusion and future works

CEDRO

- In-switch scheme to detect and re-route congested large flows
- No external controller involvement
- No end-host cooperation or network stack modifications
- Reduction of avg and 95th FCT

Future works

- Large scale simulation
- Extension to deeper topologies
- Alternative rescheduling strategies based on the capabilities of OPP
Orchestrating a brighter world

NEC