
Pablo Fondo Ferreiro
Miguel Rodríguez Pérez
Manuel Fernández Veiga
September 15, 2018
CONTEXT
Previous work on Aggregates of Energy Efficient Ethernet Links

**Straightforward Solution**
Power off unused links

- Slow response time
- What about half used links?
• Formally IEEE 802.3az.
• Low Power Idle (LPI) state.
• Sleeping and waking up is not instantaneous.

Figure 1: Energy-Efficient Ethernet model. Retrieved from [1].
**PROBLEM STATEMENT**

**Goal**
Minimize energy consumption in bundles of EEE links leveraging SDN.

\[ \lambda_i = \frac{\lambda}{4} \]
Problem Statement

Goal
Minimize energy consumption in bundles of EEE links leveraging SDN.

\[ \lambda_i = \min \left\{ C, \lambda - \sum_{k=1}^{i-1} \lambda_i \right\} \]
Goal
Minimize energy consumption in bundles of EEE links leveraging SDN.

\[ \lambda_i = \min \left\{ C, \lambda - \sum_{k=1}^{i-1} \lambda_i \right\} \]
Goal
Minimize energy consumption in bundles of EEE links leveraging SDN.

\[
\lambda_i = \min \left\{ C, \lambda - \sum_{k=1}^{i-1} \lambda_i \right\}
\]
**Goal**
Minimize energy consumption in bundles of EEE links leveraging SDN.

**Theoretical solution**
Presented in [2], provides a

- Packet level algorithm
- Assumes real time access to individual occupation of each output port

**SDN Solution**

- Needs flow level operation
- Cannot take real-time decisions based on queue occupation
- Will use ONOS for portability
SDN Algorithm
Main Tasks

- Flow identification
- Flow characterization
- Port allocation
Challenge
Which fields of the packets will identify our flows?

• We need:
  • Enough flows to distribute them along the bundle.
  • Few flows to keep flow tables small.
  • Flows with predictable demand.

• Two alternatives: *Flow tagging* vs *field matching*.

• We will aggregate IP flows:
  • MAC flows can be insufficient (e.g., transit networks).
  • Transport flows would be excessive.
Flow rate estimation

Figure 2: Average error in the estimation of the flow rate for different periods.

Use rate of previous interval with sampling rate around 0.2s
In essence, a **bin packing** problem.

**Heuristics**

**Greedy**  Corresponds to *first fit decreasing*. A flow level water-filling.

**Bounded Greedy**  Variation to reduce loses:

\[
\text{Maximum usable capacity of a link: } 1 - \frac{\text{bound}}{|\text{flows}|}
\]

**Conservative**  
- Balanced distribution among needed ports.
- Safety margin to further avoid losses.
- Note: Energy consumption raises very rapidly with traffic load.
**Conservative Algorithm**

**Behavior**
- Determines the number of needed links
- Distributed flows evenly among the links

**Basis**
EEE energy usage rises rapidly with load.

![Graph showing normalized energy usage vs. incoming traffic load]

- Ideal share
- Conservative share

Legend:
- 2-bundle link
- Incoming traffic load (Gb/s)
- Normalized Energy Usage (%)
**Conservative Algorithm**

**Behavior**
- Determines the number of needed links
- Distributed flows evenly among the links

**Basis**
EEE energy usage rises rapidly with load.

![Graph showing energy usage and traffic load relationship](image)
CONSERVATIVE ALGORITHM

Behavior

- Determines the number of needed links
- Distributed flows evenly among the links

Basis

EEE energy usage rises rapidly with load.
Experimental setup

- **Topology:** Two switches connected by 5 EEE interfaces 10 GBASE-T.
- We have used real traffic traces retrieved from CAIDA [3].
- **Baseline:** Equitable algorithm.

- **Metrics:**
  - Energy consumption
  - Packet losses
  - Packet delay
**Results: Energy Consumption**

![Graph of energy consumption vs. sampling period for different algorithms](image)

(a) 32.5 Gbit/s trace.  
(b) Sampling period = 0.5 seconds.

**Figure 3:** Normalized energy consumption (buffer = 10000 packets).

- Theoretical bound for the consumption of the 32.5 Gbit/s: 78.5%.
Results: Packet losses

Figure 4: Packet loss percentage (sampling period = 0.5 seconds).

(a) 32.5 Gbit/s trace.

(b) buffer = 10000 packets.
**Results: Packet delay**

(a) 32.5 Gbit/s trace.

(b) Sampling period = 0.5 seconds.

**Figure 5:** Average per packet delay (buffer = 10000 packets).
QoS-aware algorithms
Goal
Provide low-latency service while reducing energy consumption.

- The previous algorithms manage to reduce energy consumption.
- However, they increase the delay of the packets.
- We consider now the QoS latency requirements of the flows.
- Two types of traffic:
  - Best-effort.
  - Low-latency.
- Modifications to the previous algorithms.
Spare Port

1. Apply energy-efficient algorithm to best-effort flows.
2. Low-latency flows are allocated to the most empty port.

Two Queues

1. Apply energy-efficient algorithm to all the flows.
2. Low-latency flows are allocated to the high-priority queue of the assigned ports.
• Same topology: 5-link bundle of 10 GBASE-T EEE interfaces.
• Real traces for best-effort traffic.
• Synthetic traffic for low-latency packets.
• Baseline: Conservative algorithm.
• Parameters:
  • Buffer = 10 000 packets.
  • Sampling period = 0.5 seconds.
• Metrics:
  • Delay of low-latency packets.
  • Delay of best-effort packets.
  • Energy consumption.
RESULTS: DELAY OF LOW-LATENCY PACKETS

Figure 8: Average delay of low-latency packets.

(a) 32.5 Gbit/s trace.

(b) 45.5 Gbit/s trace.
Figure 9: Average delay of best-effort packets (32.5 Gbit/s trace).
Figure 10: Normalized energy consumption (32.5 Gbit/s trace).
CONCLUSIONS
Conclusions

- SDN can be leveraged to implement energy saving algorithms
- Results match theoretical model
- Provided low latency service based on QoS requirements

Future work

- Reuse edge allocations for inner switches.
- Reduce control plane traffic (e.g., minimize flow re-allocations).
THANK YOU FOR LISTENING!

EMAIL: miguel@det.uvigo.es
