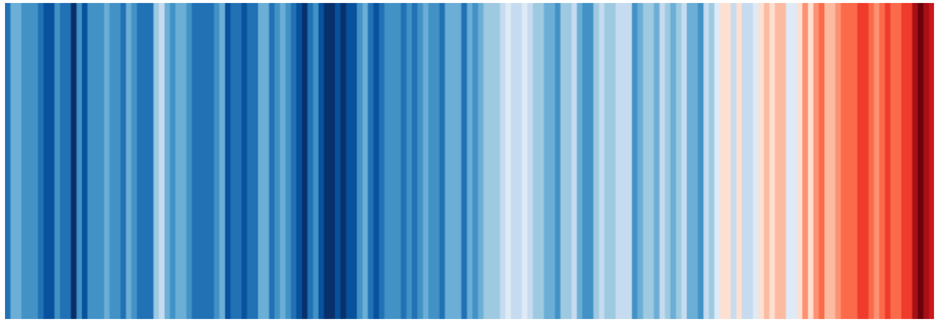
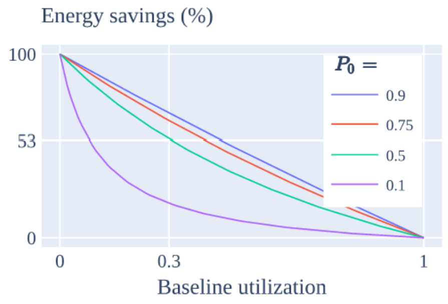
Does rate adaptation at daily timescales make sense?



Romain Jacob
Jackie Lim
Laurent Vanbever
ETH Zürich

2nd HotCarbon Workshop
July 9, 2023

We could save 50% energy
in today's ISP networks.



Possible savings are

≥ 50% given

- $P_0/P_1 \geq 0.5$
- Utilization $\lesssim 30\%$

[in an ideal world]

Tomorrow's Internet must sleep more and grow old.

HotCarbon 2022

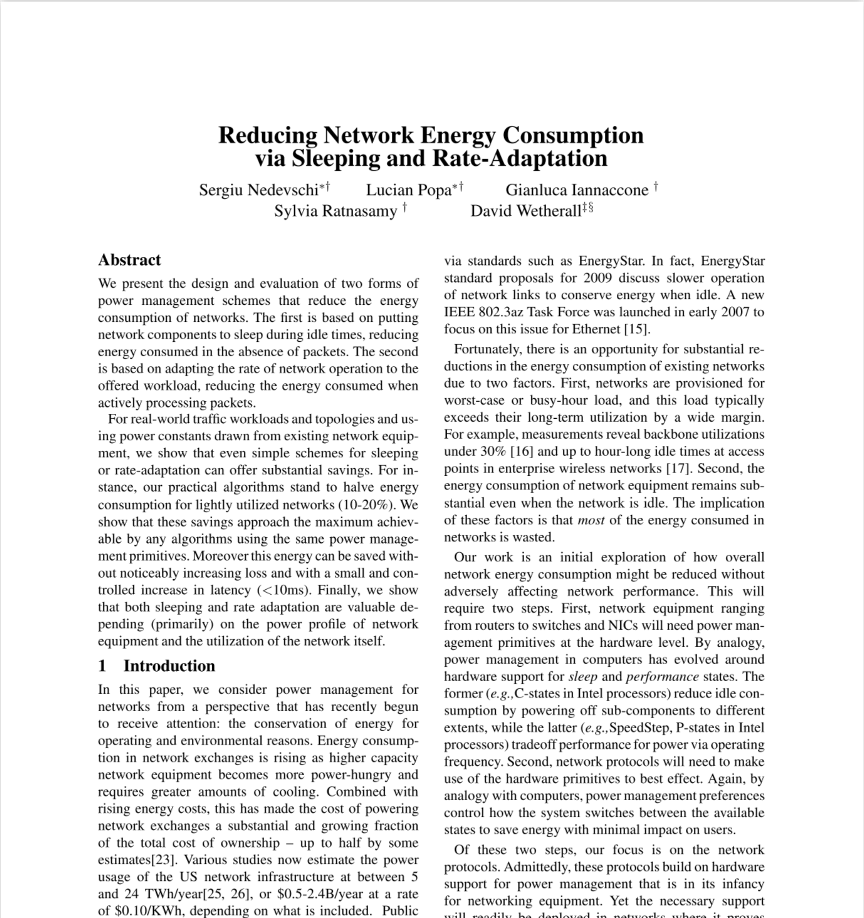
To harness these benefits, we must speed up the devices "start-up time."

From "low-power" to "ready-to-forward"

Routing protocols x Networking software x Networking hardware

We know we can save a lot by turning line cards to sleep

NSDI 2008



Reducing Network Energy Consumption via Sleeping and Rate-Adaptation

Sergiu Nedevschi[*†] Lucian Popa[*†] Gianluca Iannaccone [†]
Sylvia Ratnasamy [†] David Wetherall[‡§]

Abstract

We present the design and evaluation of two forms of power management schemes that reduce the energy consumption of networks. The first is based on putting network components to sleep during idle times, reducing energy consumed in the absence of packets. The second is based on adapting the rate of network operation to the offered workload, reducing the energy consumed when actively processing packets.

For real-world traffic workloads and topologies and using power constants drawn from existing network equipment, we show that even simple schemes for sleeping or rate-adaptation can offer substantial savings. For instance, our practical algorithms stand to halve energy consumption for lightly utilized networks (10-20%). We show that these savings approach the maximum achievable by any algorithms using the same power management primitives. Moreover this energy can be saved without noticeably increasing loss and with a small and controlled increase in latency (<10ms). Finally, we show that both sleeping and rate adaptation are valuable depending (primarily) on the power profile of network equipment and the utilization of the network itself.

1 Introduction

In this paper, we consider power management for networks from a perspective that has recently begun to receive attention: the conservation of energy for operating and environmental reasons. Energy consumption in network exchanges is rising as higher capacity network equipment becomes more power-hungry and requires greater amounts of cooling. Combined with rising energy costs, this has made the cost of powering network exchanges a substantial and growing fraction of the total cost of ownership – up to half by some estimates[23]. Various studies now estimate the power usage of the US network infrastructure at between 5 and 24 TWh/year[25, 26], or $0.5-2.4B/year at a rate of $0.10/KWh, depending on what is included. Public via standards such as EnergyStar. In fact, EnergyStar standard proposals for 2009 discuss slower operation of network links to conserve energy when idle. A new IEEE 802.3az Task Force was launched in early 2007 to focus on this issue for Ethernet [15].

Fortunately, there is an opportunity for substantial reductions in the energy consumption of existing networks due to two factors. First, networks are provisioned for worst-case or busy-hour load, and this load typically exceeds their long-term utilization by a wide margin. For example, measurements reveal backbone utilizations under 30% [16] and up to hour-long idle times at access points in enterprise wireless networks [17]. Second, the energy consumption of network equipment remains substantial even when the network is idle. The implication of these factors is that *most* of the energy consumed in networks is wasted.

Our work is an initial exploration of how overall network energy consumption might be reduced without adversely affecting network performance. This will require two steps. First, network equipment ranging from routers to switches and NICs will need power management primitives at the hardware level. By analogy, power management in computers has evolved around hardware support for *sleep* and *performance* states. The former (*e.g.*,C-states in Intel processors) reduce idle consumption by powering off sub-components to different extents, while the latter (*e.g.*,SpeedStep, P-states in Intel processors) tradeoff performance for power via operating frequency. Second, network protocols will need to make use of the hardware primitives to best effect. Again, by analogy with computers, power management preferences control how the system switches between the available states to save energy with minimal impact on users.

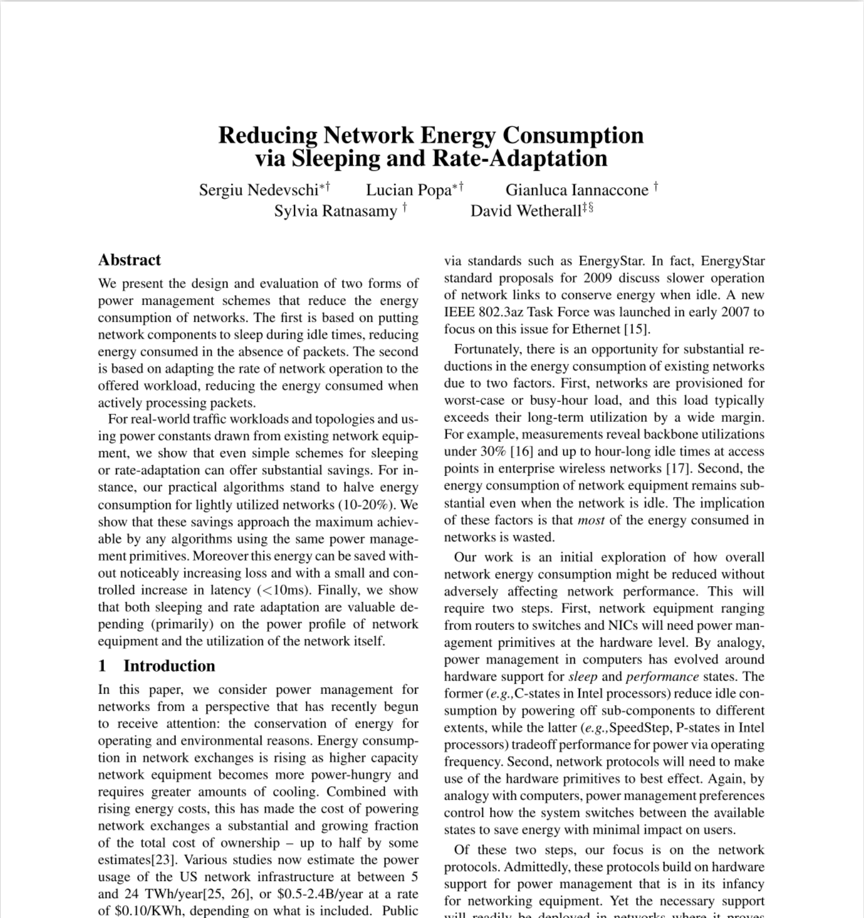
Of these two steps, our focus is on the network protocols. Admittedly, these protocols build on hardware support for power management that is in its infancy for networking equipment. Yet the necessary support

Two strategies

- **Sleeping** — Buffer and burst packets; Turn off links between burst
- **Down-rating** — Keep all links up; Match port rates with demand

We know we can save a lot by turning line cards to sleep… **but we can’t really do it.**

NSDI 2008



Reducing Network Energy Consumption via Sleeping and Rate-Adaptation

Sergiu Nedevschi[*†] Lucian Popa[*†] Gianluca Iannaccone [†]
Sylvia Ratnasamy [†] David Wetherall[‡§]

Abstract

We present the design and evaluation of two forms of power management schemes that reduce the energy consumption of networks. The first is based on putting network components to sleep during idle times, reducing energy consumed in the absence of packets. The second is based on adapting the rate of network operation to the offered workload, reducing the energy consumed when actively processing packets.

For real-world traffic workloads and topologies and using power constants drawn from existing network equipment, we show that even simple schemes for sleeping or rate-adaptation can offer substantial savings. For instance, our practical algorithms stand to halve energy consumption for lightly utilized networks (10-20%). We show that these savings approach the maximum achievable by any algorithms using the same power management primitives. Moreover this energy can be saved without noticeably increasing loss and with a small and controlled increase in latency (<10ms). Finally, we show that both sleeping and rate adaptation are valuable depending (primarily) on the power profile of network equipment and the utilization of the network itself.

1 Introduction

In this paper, we consider power management for networks from a perspective that has recently begun to receive attention: the conservation of energy for operating and environmental reasons. Energy consumption in network exchanges is rising as higher capacity network equipment becomes more power-hungry and requires greater amounts of cooling. Combined with rising energy costs, this has made the cost of powering network exchanges a substantial and growing fraction of the total cost of ownership – up to half by some estimates[23]. Various studies now estimate the power usage of the US network infrastructure at between 5 and 24 TWh/year[25, 26], or $0.5-2.4B/year at a rate of $0.10/KWh, depending on what is included. Public via standards such as EnergyStar. In fact, EnergyStar standard proposals for 2009 discuss slower operation of network links to conserve energy when idle. A new IEEE 802.3az Task Force was launched in early 2007 to focus on this issue for Ethernet [15].

Fortunately, there is an opportunity for substantial reductions in the energy consumption of existing networks due to two factors. First, networks are provisioned for worst-case or busy-hour load, and this load typically exceeds their long-term utilization by a wide margin. For example, measurements reveal backbone utilizations under 30% [16] and up to hour-long idle times at access points in enterprise wireless networks [17]. Second, the energy consumption of network equipment remains substantial even when the network is idle. The implication of these factors is that *most* of the energy consumed in networks is wasted.

Our work is an initial exploration of how overall network energy consumption might be reduced without adversely affecting network performance. This will require two steps. First, network equipment ranging from routers to switches and NICs will need power management primitives at the hardware level. By analogy, power management in computers has evolved around hardware support for *sleep* and *performance* states. The former (*e.g.,*C-states in Intel processors) reduce idle consumption by powering off sub-components to different extents, while the latter (*e.g.,*SpeedStep, P-states in Intel processors) tradeoff performance for power via operating frequency. Second, network protocols will need to make use of the hardware primitives to best effect. Again, by analogy with computers, power management preferences control how the system switches between the available states to save energy with minimal impact on users.

Of these two steps, our focus is on the network protocols. Admittedly, these protocols build on hardware support for power management that is in its infancy for networking equipment. Yet the necessary support

Two strategies

- **Sleeping** — Buffer and burst packets; Turn off links between burst
- **Down-rating** — Keep all links up; Match port rates with demand

Hypothesis

- **Start-up delay** — Assumed to be $1ms$; Measured in minutes

Or can we?

How much could we save by turning things off only a couple of times per day?

- We consider networks where the average utilization is typically low: ISPs, Stub ASs, Entreprise networks
- We only consider strategies with no routing change.

We need two pieces

We need two pieces
a link utilization dataset

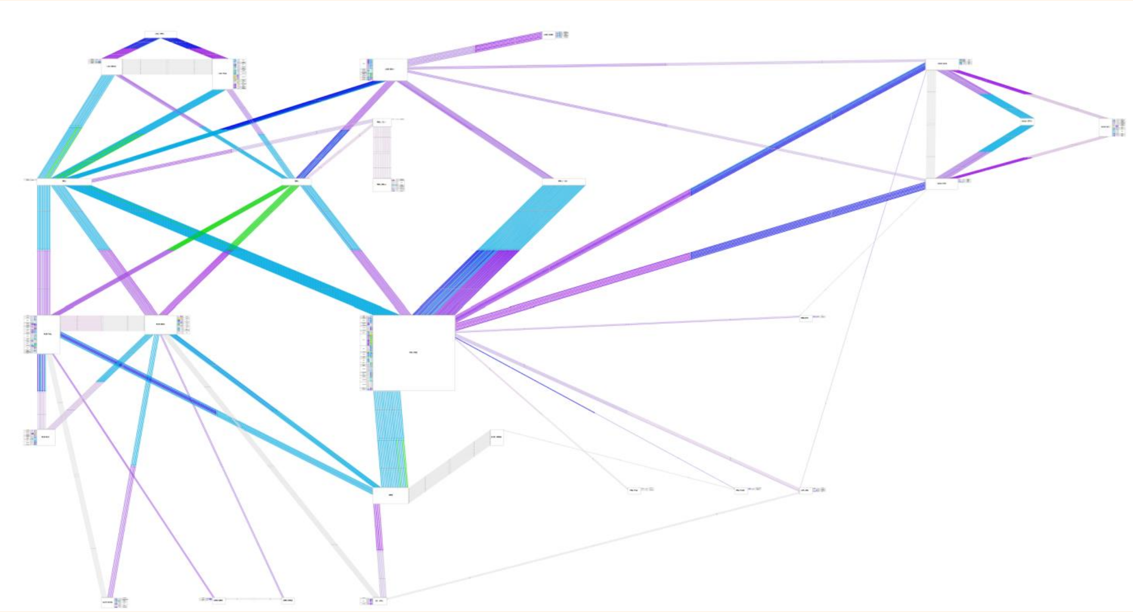
- For each physical link
- Fine-grained data

- For all links in a network
- Long-term data

We need two pieces
a link utilization dataset

Analyzing the OVH Weathermap

Europe Backbone



- For each physical link
- Fine-grained data

- For all links in a network
- Long-term data

We need two pieces
a link utilization dataset   a router power model

- Per active port
- Per port configuration

We need two pieces
a link utilization dataset

- Per active port
- Per port configuration

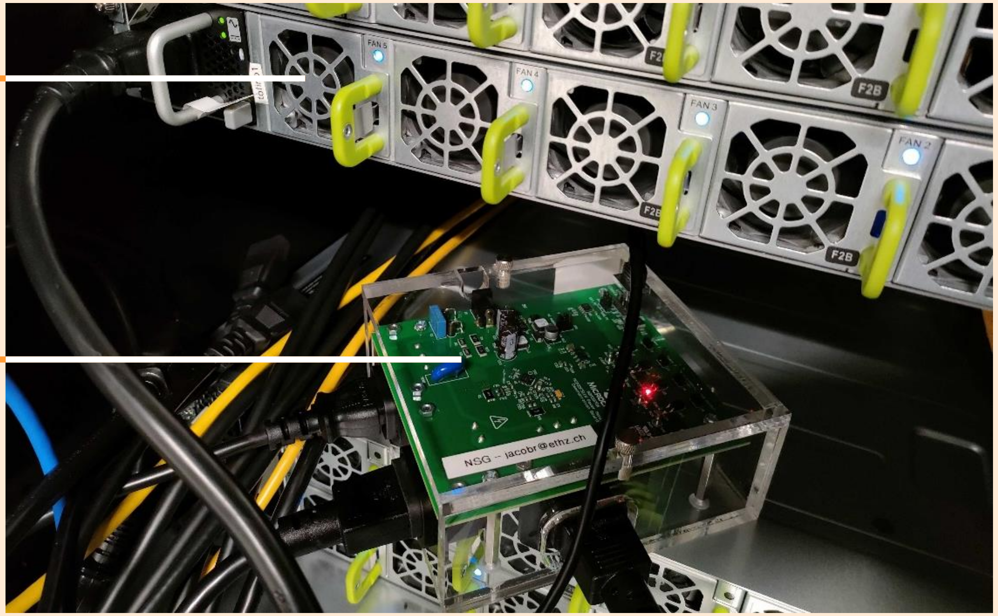
a router power model

Profiling a Tofino switch

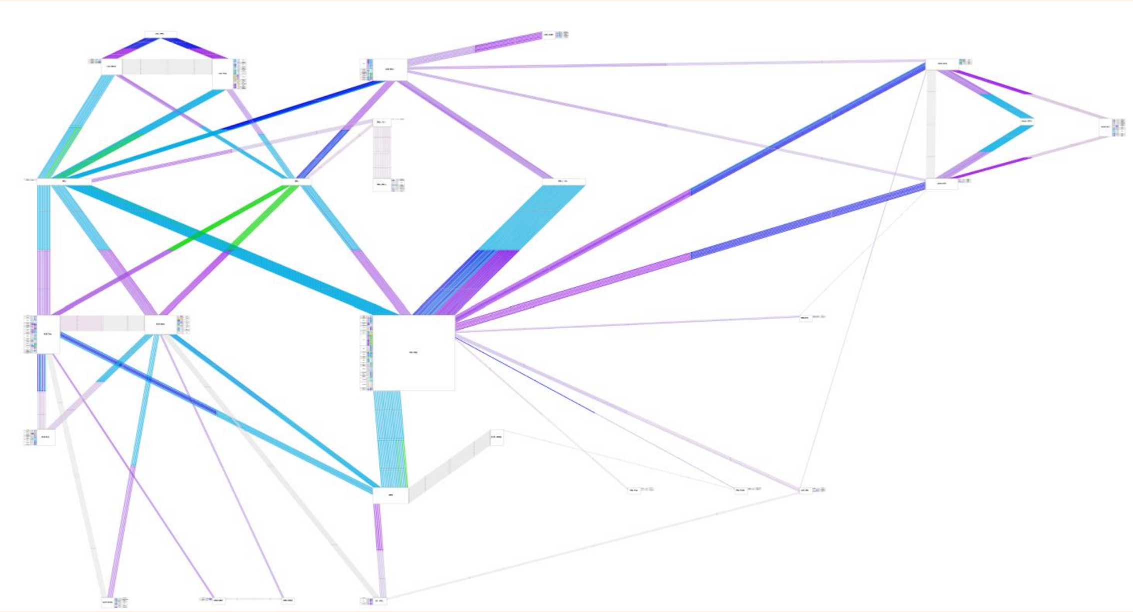
WEDGE 100BF-32X

Wedge switch

Power meter

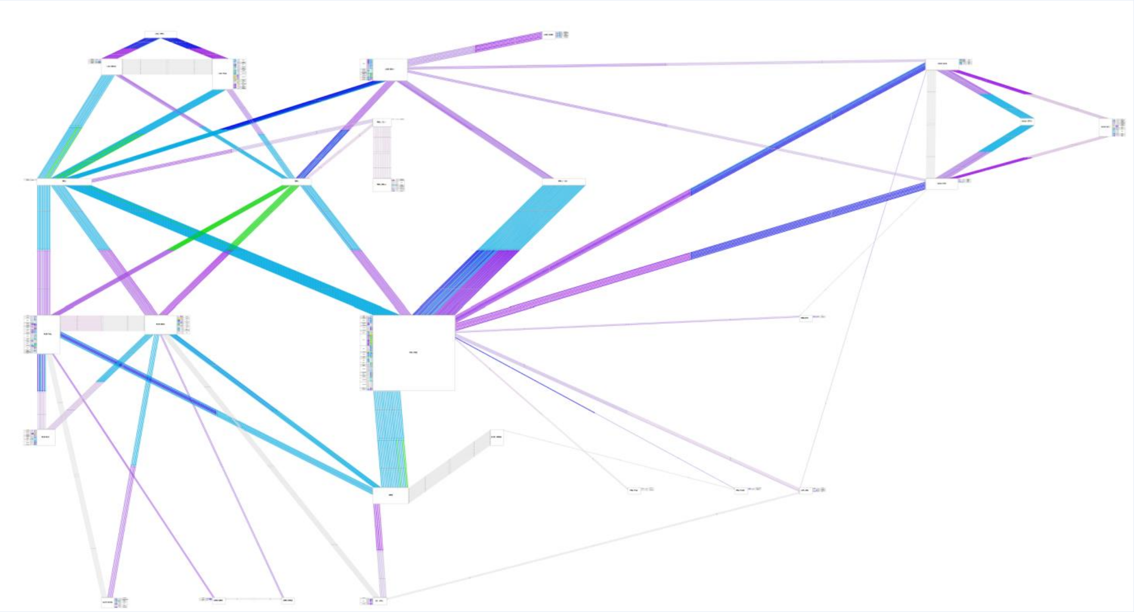


Analyzing the OVH Weathermap



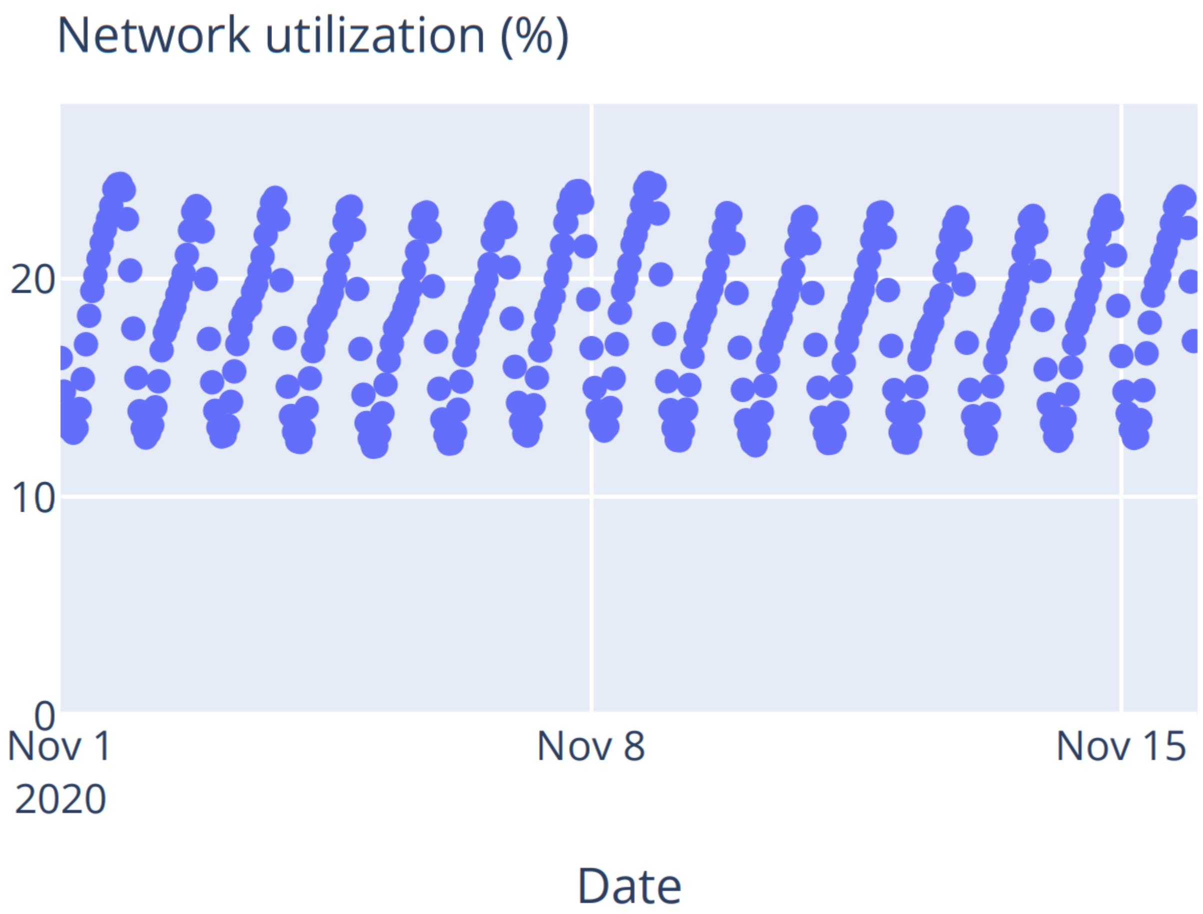
Profiling a Tofino switch

Analyzing the OVH Weathermap

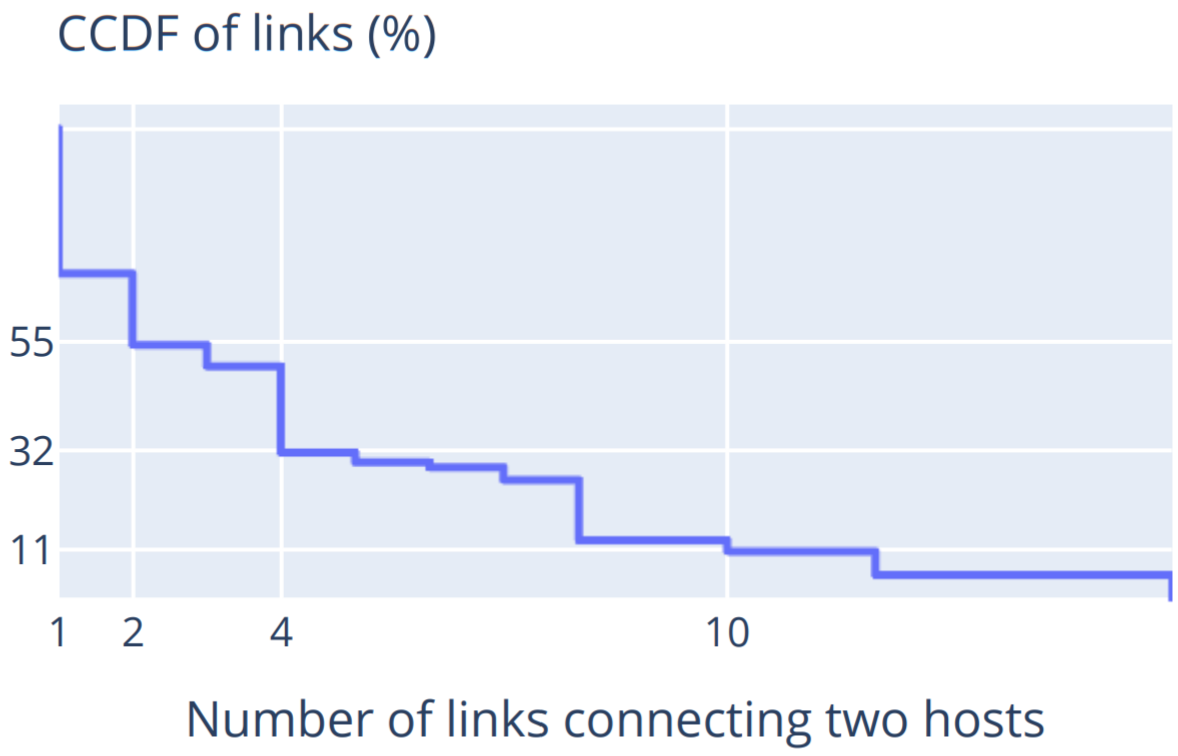


Profiling a Tofino switch

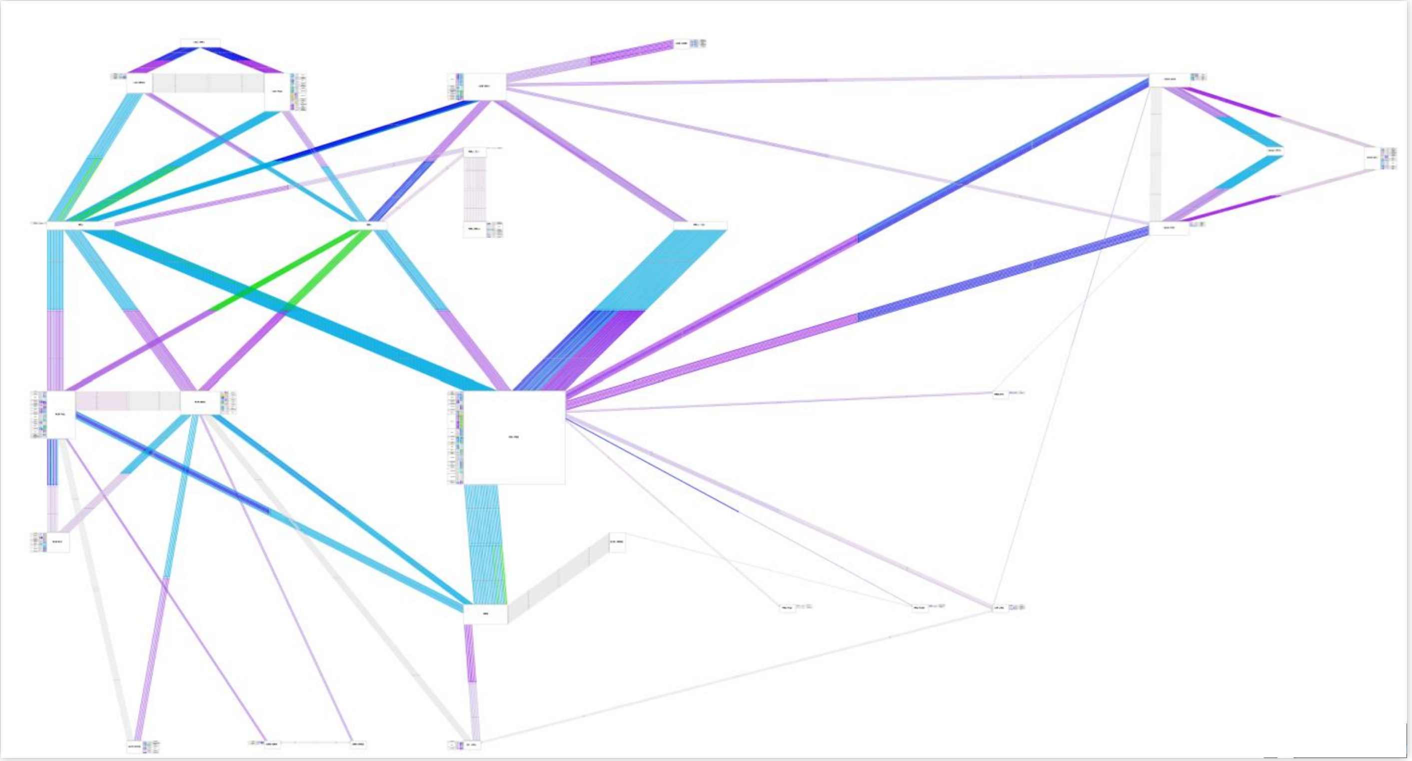
The network link utilization is both low and seasonal



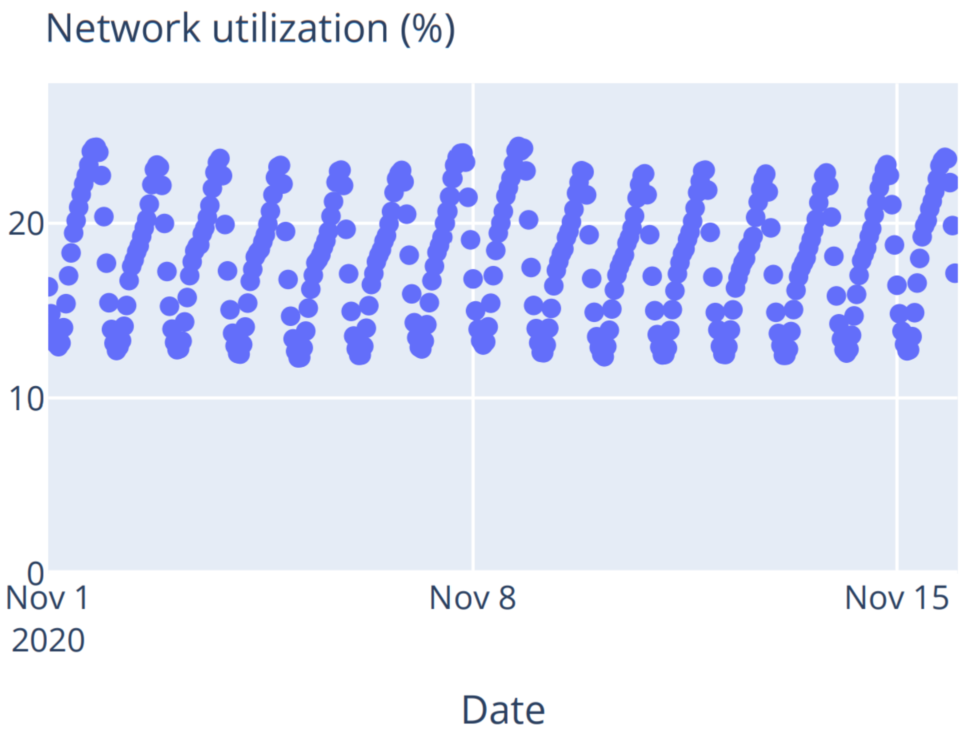
There are often many parallel links between router pairs

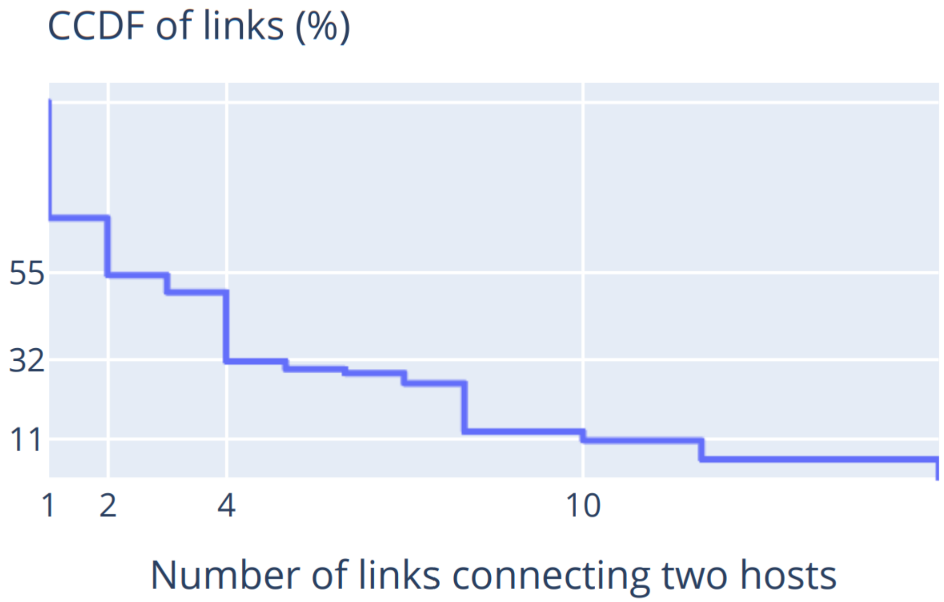


There are often many parallel links between router pairs

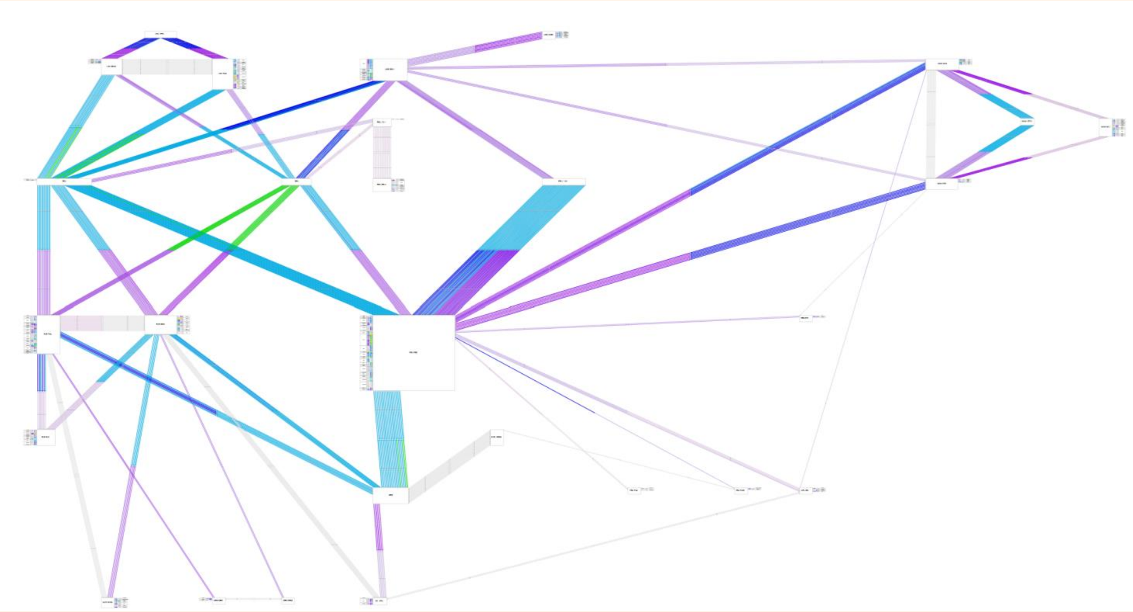


Both aspects combined suggest a large potential for turning off links





Analyzing the OVH Weathermap



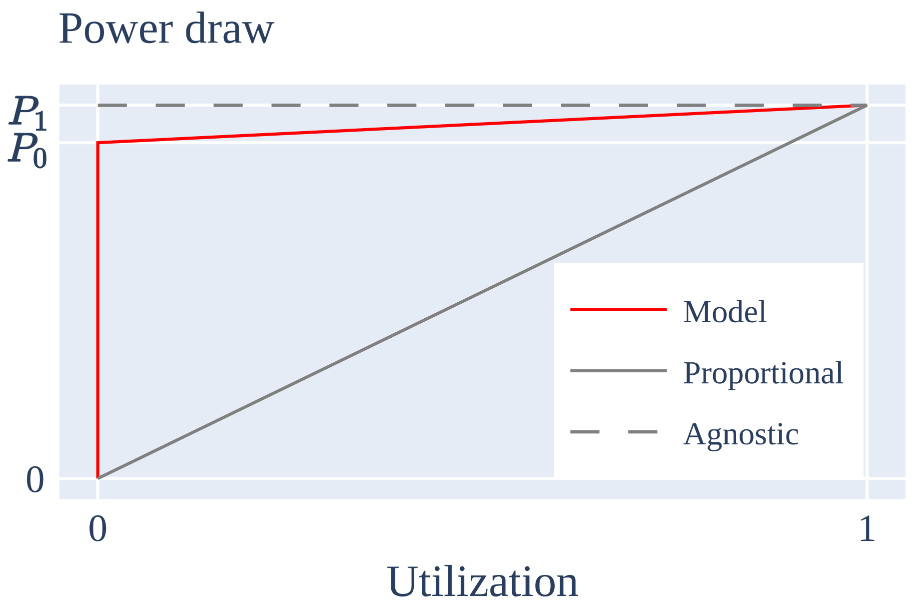
Profiling a Tofino switch

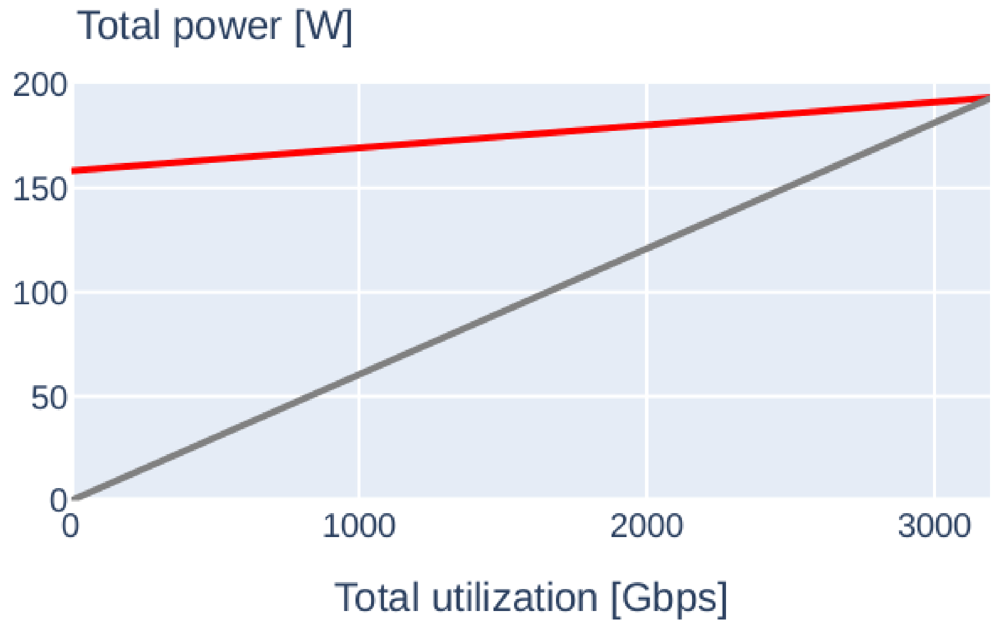
We derive a power profile for a Tofino switch under various loads and port configurations

Main insights	Added power*	Total power*
The idle power is low	$\sim$ 108W	108W
There is a variable power cost per port	0.3 – 1.6W/port	118 – 158W
Power increseases linearly but slowly with traffic	$\sim$ 1W/100Gbps	122 – 193W
The data plane program matters little	20-30W at most	125 – 223W

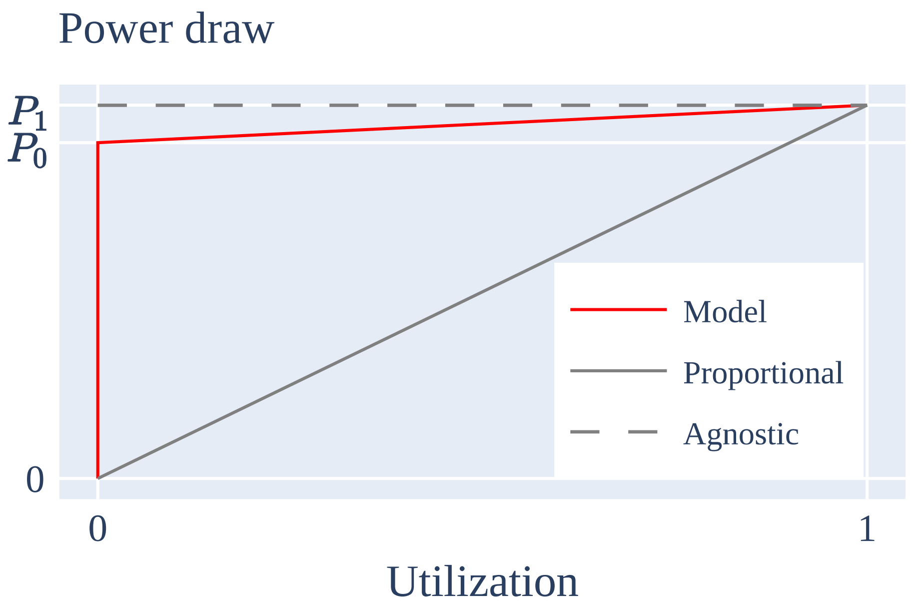
*Connected with Passive DAC (direct attach copper) cables

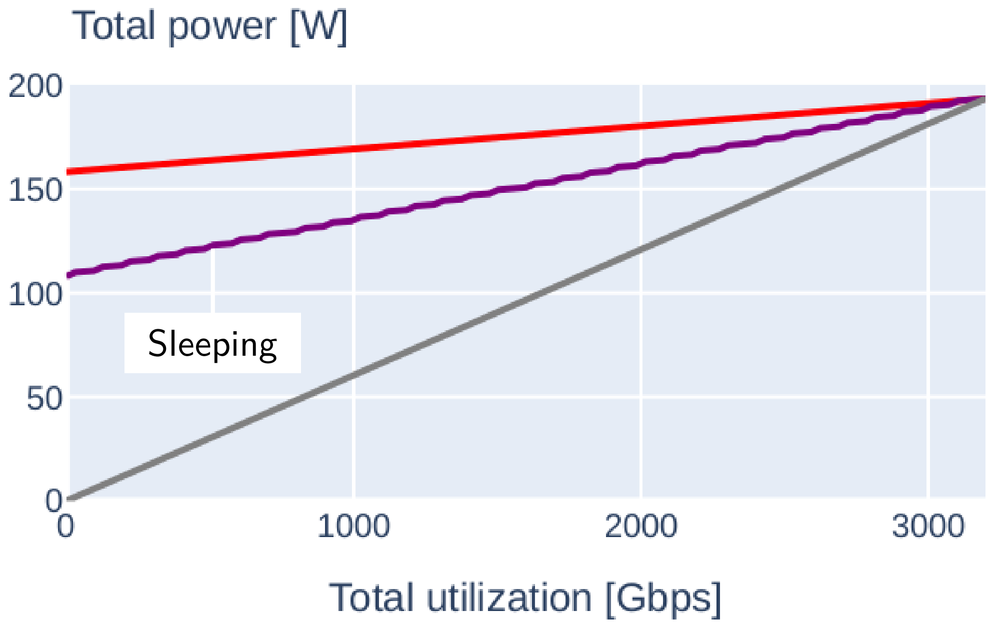
We derive a power profile for a Tofino switch under various loads and port configurations





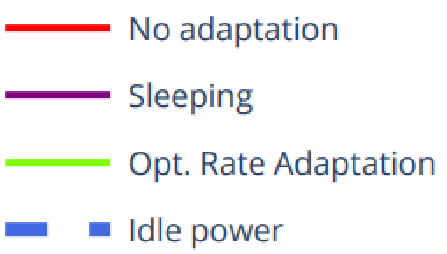
We derive a power profile for a Tofino switch under various loads and port configurations

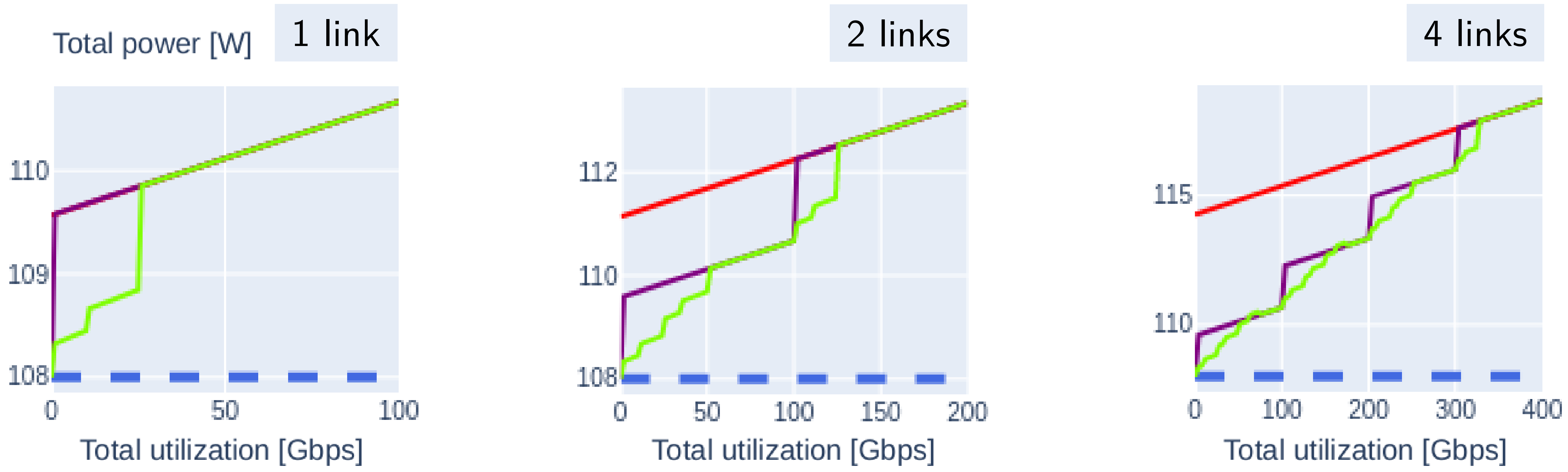




Still far from proportional
but already much better!

Changing the port rate yields noticeable power savings





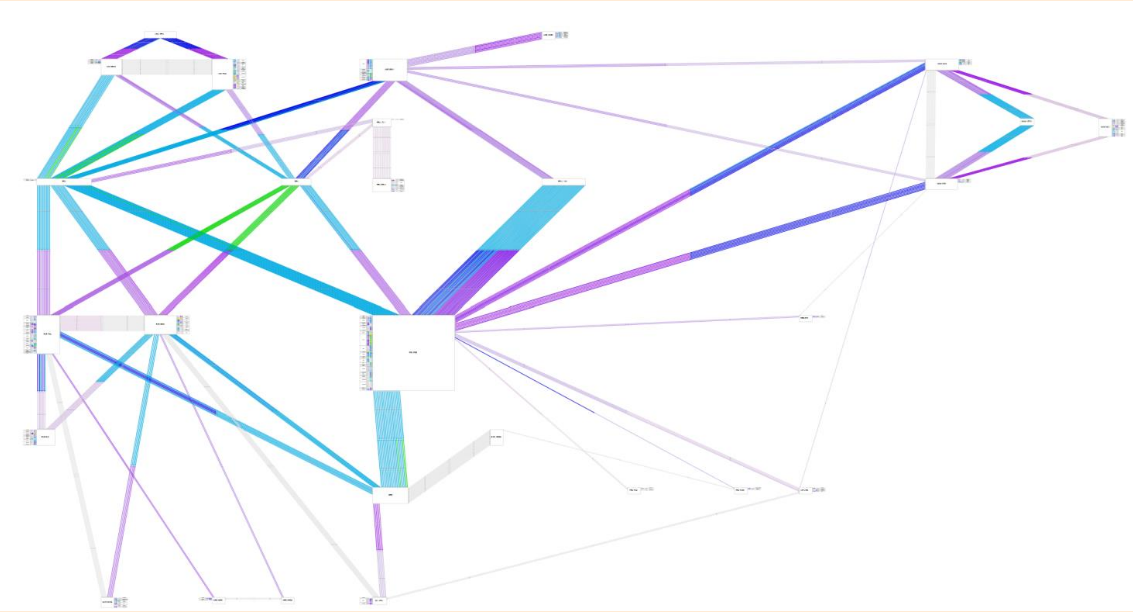
- Cannot sleep
- Down-rating helps

- Sleeping is simple and effective
- Down-rating helps further

- The more parallel links, the more possible savings

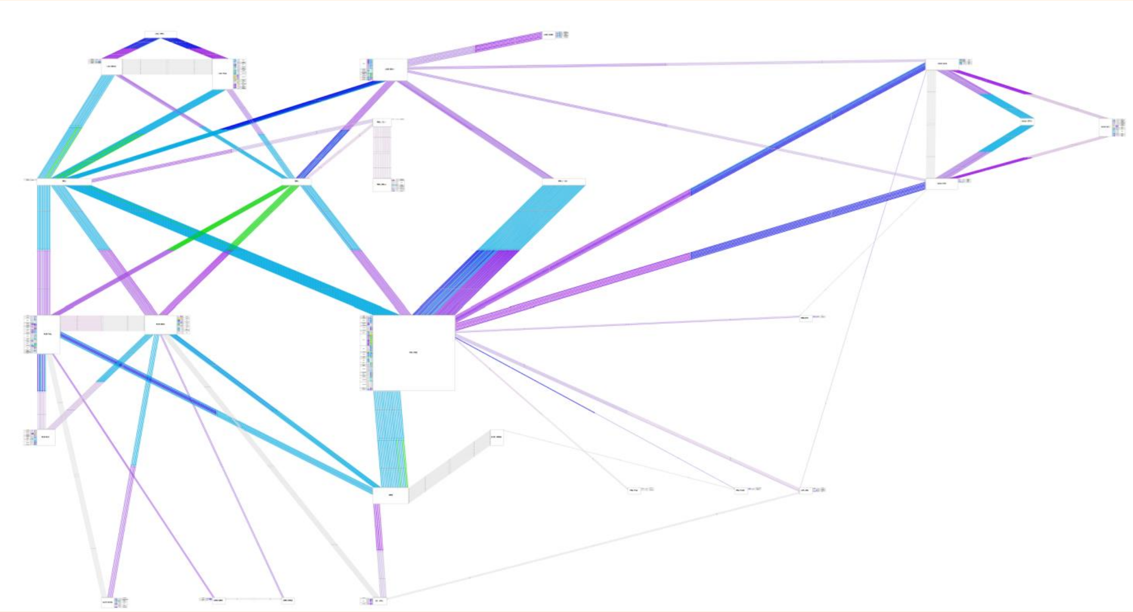
So, how much can we save?

Analyzing the OVH Weathermap + Profiling a Tofino switch = ??



So, how much can we save? Tens of MWh/year. But ...

Analyzing the OVH Weathermap + Profiling a Tofino switch = MWh/year



Naively applying this power model to the OVH dataset is a gross approximation

Putting those two things together
is somewhat far-fetched

- Not the right routers
- Not the right transceivers
- Neglect overheads

Putting those two things together is somewhat far-fetched

- Not the right routers
- Not the right transceivers
- Neglect overheads

To do better, we need yet-unavailable data

- Deployment data

  Which device and cable models are deployed in the network?

- Power benchmarks

  How much power those devices draw depending on thir utilization?

- Sleeping implementation

  What are the overheads induced by sleeping or down-rating?

To do better, we need yet-unavailable data

Putting those two things together is somewhat far-fetched

- Not the right routers
- Not the right transceivers
- Neglect overheads

Help welcome!

- Deployment data
  Which device and cable models
  d in the network?
- marks
  How much power those devices draw depending on thir utilization?
- Sleeping implementation
  What are the overheads induced by sleeping or down-rating?

Some thoughts about researching sustainable networking

Admittedly subjective opinions

You might be thinking...

1 The network footprint is small relative to the whole ICT.

- That is likely true. And it does not matter.

You might be thinking...

1. The network footprint is small relative to the whole ICT.
   - That is likely true. And it does not matter.
2. The Internet wasn't built to be power proportional.
   - True. But maybe it should be made more so.

You might be thinking...

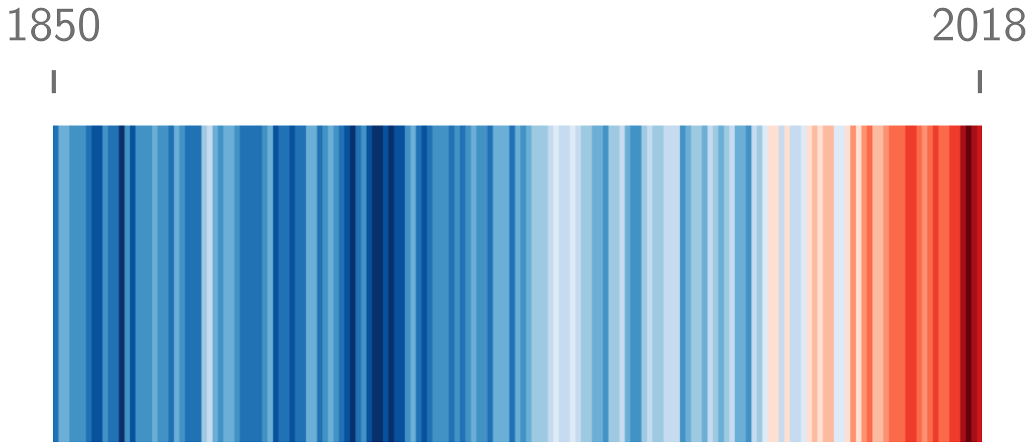
1. The network footprint is small relative to the whole ICT.
   - That is likely true. And it does not matter.
2. The Internet wasn't built to be power proportional.
   - True. But maybe it should be made more so.
3. I am not convinced putting networks to sleep is a good idea.
   - Me neither! But I'm convinced it must be investigated carefully.

Does rate adaptation at daily timescales make sense?

Possibly.

Still a lot of research needed

- Deployment data
- Power benchmarks
- Sleeping implementation



Climate stripes. Ed Hawkins, 2018
portrays the increase of average global temperature

Romain Jacob
Jackie Lim
Laurent Vanbever
ETH Zürich

jacobr@ethz.ch