On-Demand Clock Boosting for Secure Remote Work System

Justus von der Beek

Studying M.Sc in Informatics, Technical University of Munich

beek@in.tum.de

Atsushi Shinoda†, Hajime Shimada†, Hirokazu Hasegawa*

† Nagoya University, Japan

*** National Institute of Informatics, Japan**

Short Bio

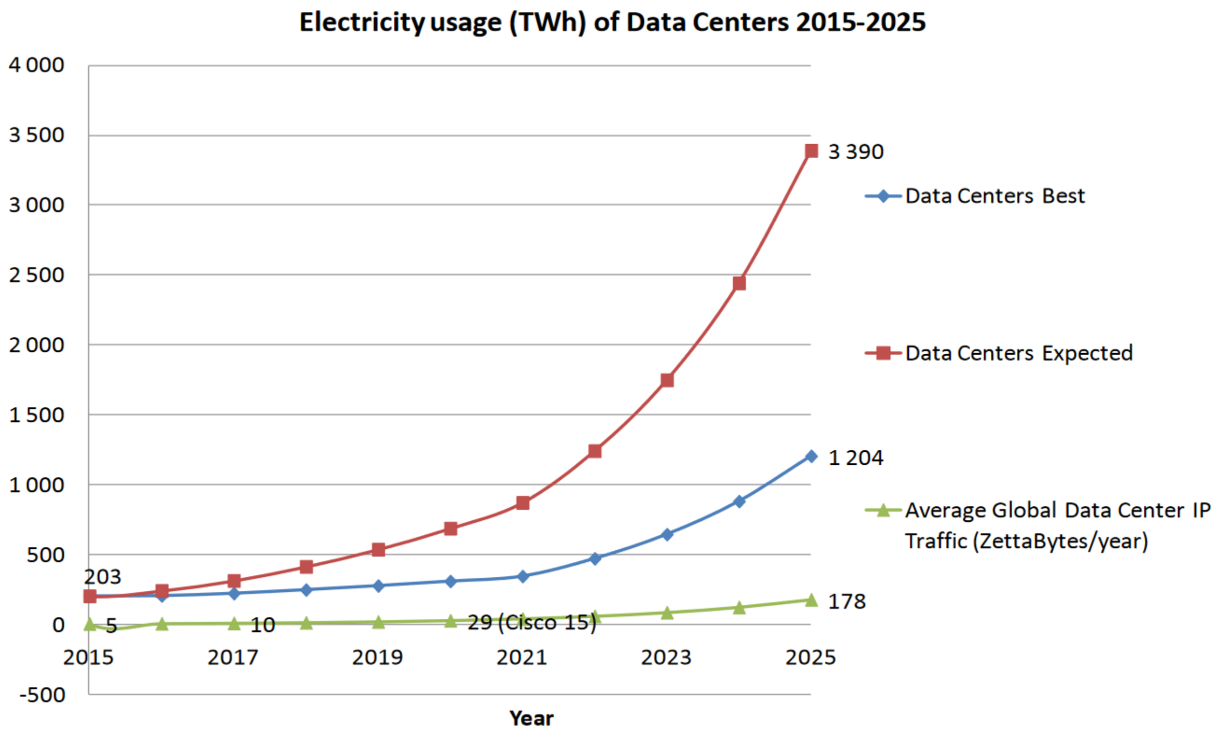
- Justus von der Beek is a master student at the Technical University of Munich
- He is currently pursing his master degree in informatics with interest in computer networking and network security
- During his exchange with the Nagoya University in Japan he researched on energy efficiency of computing systems

Outline

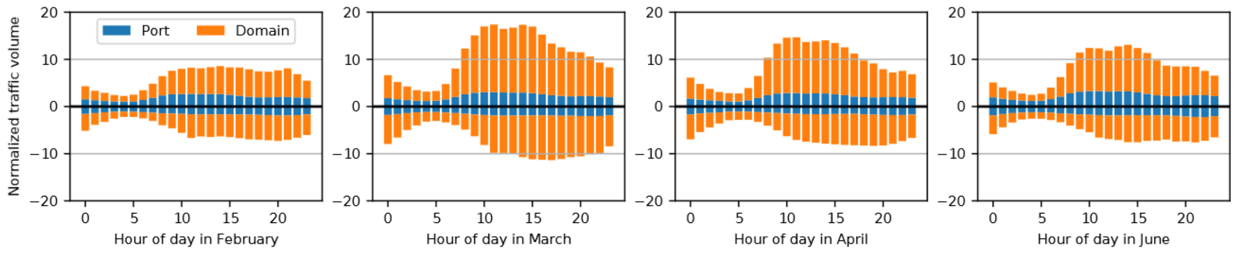
- Motivation
- Work System Overview
- Boosting CPU Frequency On-Demand
- Implementation Designs
- Clock Boosting Algorithm
- Evaluation

Motivation

- Electricity usage estimation of Liu et al. [1] for data centers of 2015 and beyond
  - Expected to be one of the major energy consumers
- Rise in remote learning [2] and VPN traffic since 2019 [3]



Data center electricity usage estimation, Liu et al. [1]

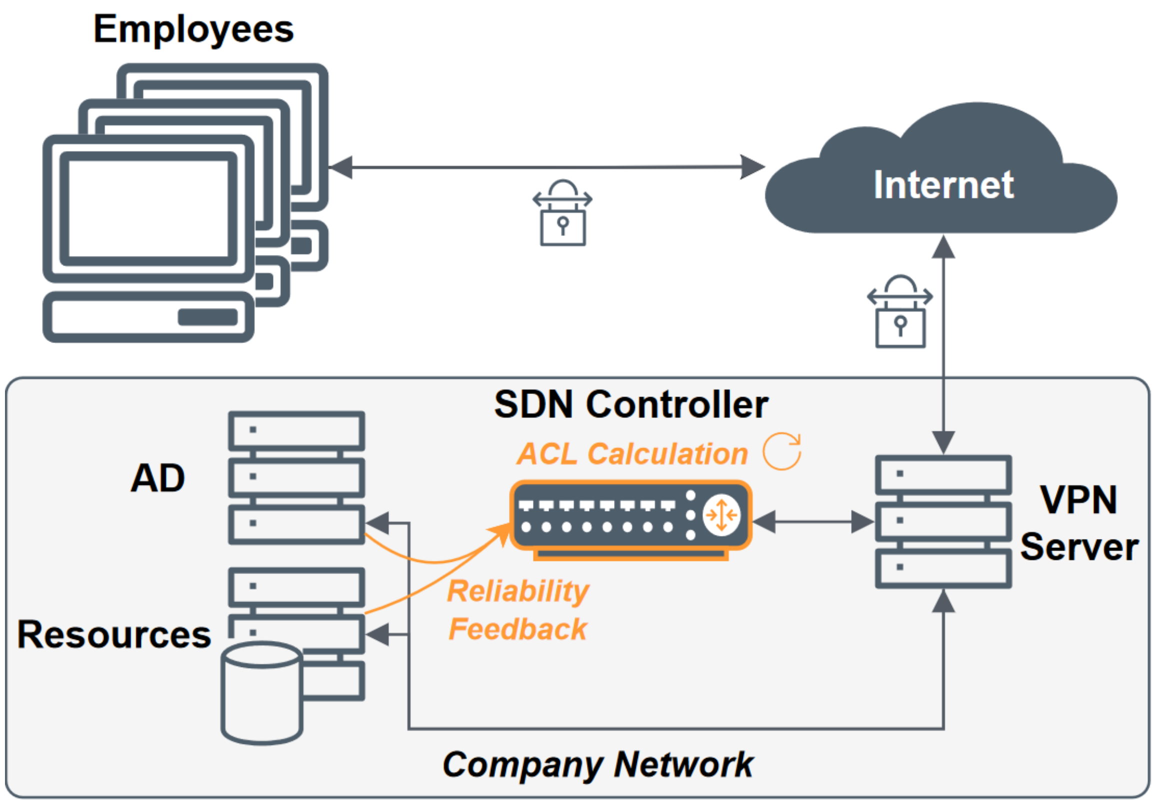


Normalized VPN traffic volume at IXP-CE in 2020, Feldmann et al. [3]

[1] Y. Liu et al., “Energy consumption and emission mitigation prediction based on data center traffic and PUE for global data centers,” Global Energy Interconnection, vol. 3, no. 3, pp. 272–282.
[2] T. Favale et al., “Campus traffic and e-learning during COVID-19 pandemic,” Computer Networks, vol. 176, article 107290.
[3] A. Feldmann et al., “The lockdown effect: Implications of the COVID-19 pandemic on internet traffic,” ACM, pp. 1–18., 2020

Work System Overview

- Shinoda et al. [4] developed a Software Defined Networking (SDN) remote work system with additional Access Control List (ACL) feature
  - Gain additional system security through blocked access to (high-) risk data
  - Simulating company network via Software Defined Networking (SDN)
  - Deploying ACL in SDN Controller
  - Calculation on SDN Controller compute intensive



Work system overview developed by Shinoda et al. [4]

[4] A. Shinoda et al., “Implementation of access control method for telecommuting communication based on users’ reliability,” CSS, pp. 840–847.,2022

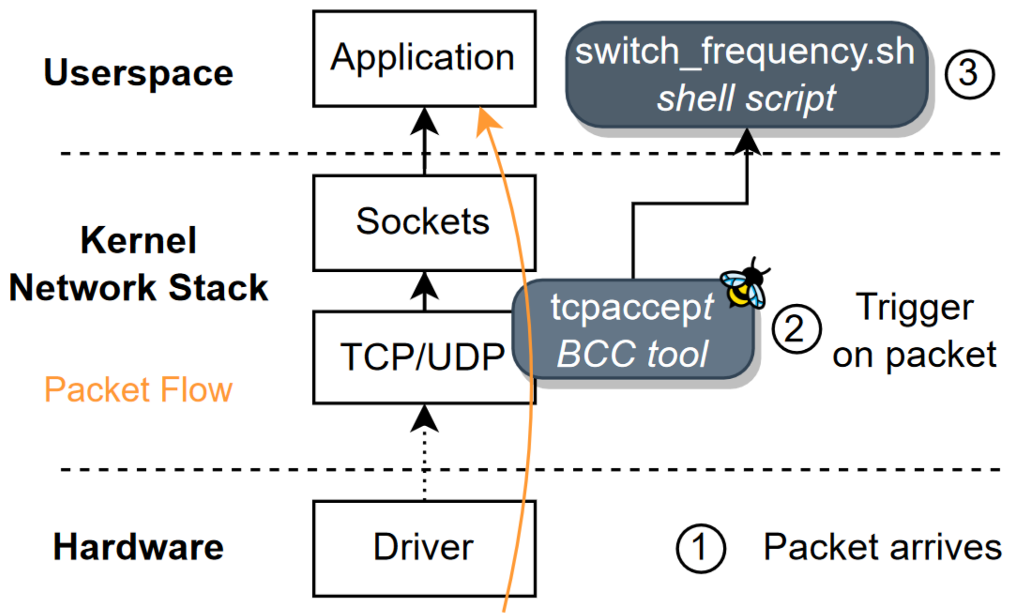
Idea: Boosting CPU Frequency On-Demand

- We don‘t need computation power if no client is connected
  - All power as soon as computation is started
- CPU power and voltage are quadratically connected
  - $P = a * C * F * V^2$ (C: Circuit capacitance, a: activity factor, F: frequency, V: supply voltage)
- SDN-Controller only works when client is connected
  - Increase frequency
  - From start of connection until the disconnect
- Challenge 1: do not increase the idle power consumption
  - Long idle times (hours) are possible
  - Adverse effect if idle power consumption is increased too much
- Challenge 2: predicting future connections & connection patterns
  - Out of scope for this paper
  - Reacting to feedback from system as fast as possible

Implementation Designs: BCC & XDP

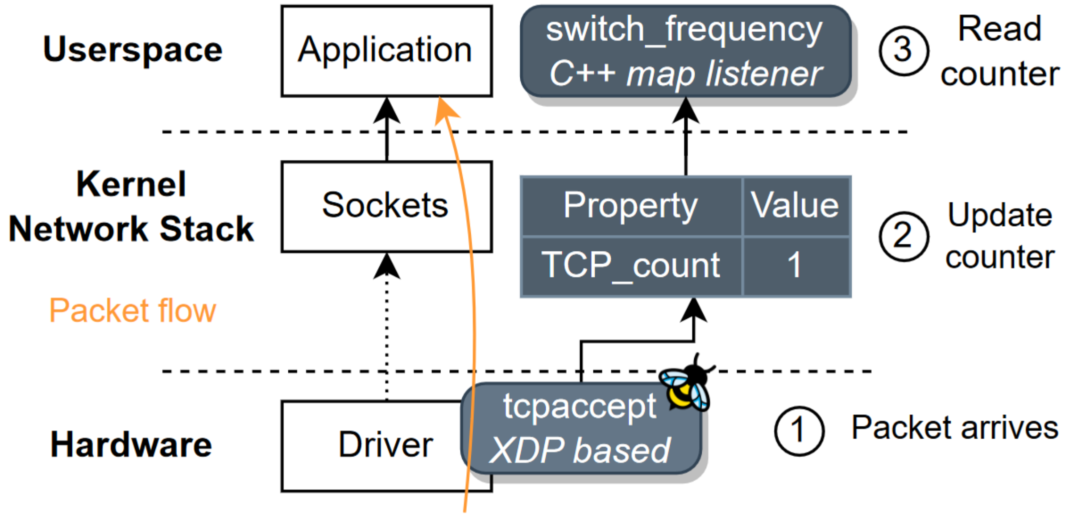
Idea 1: BPF compiler collection (BCC) & cpupower

- Pros: less hardware requirments, easier to deploy
- Cons: Slower reaction time, higher overhead



Idea 2: eXpress Data Path (XDP) & sysfs (Linux)

- Pros: lower in network stack, hence faster reaction
- Cons: requires XDP driver support



Clock Boosting Algorithm

```
if packet is TCP then
    if packet.flags is SYN then
        map[TCP_count] ← map[TCP_count] + 1
    else if packet.flags is FIN then
        map[TCP_count] ← map[TCP_count] − 1
    end if
end if
```

The XDP program listening for TCP connects and disconnects

```
prv_count ← 0
while true do
    count ← map[TCP_count]
    if count ≥ 1 and prv count = 0 then
        boost_frequency()
        prv_count ← count
    end if
    if count = 0 and prv count ≥ 1 then
        reset_frequency()
        prv_count ← count
    end if
end while
```

The userspace program updating the frequency based on the number of TCP clients

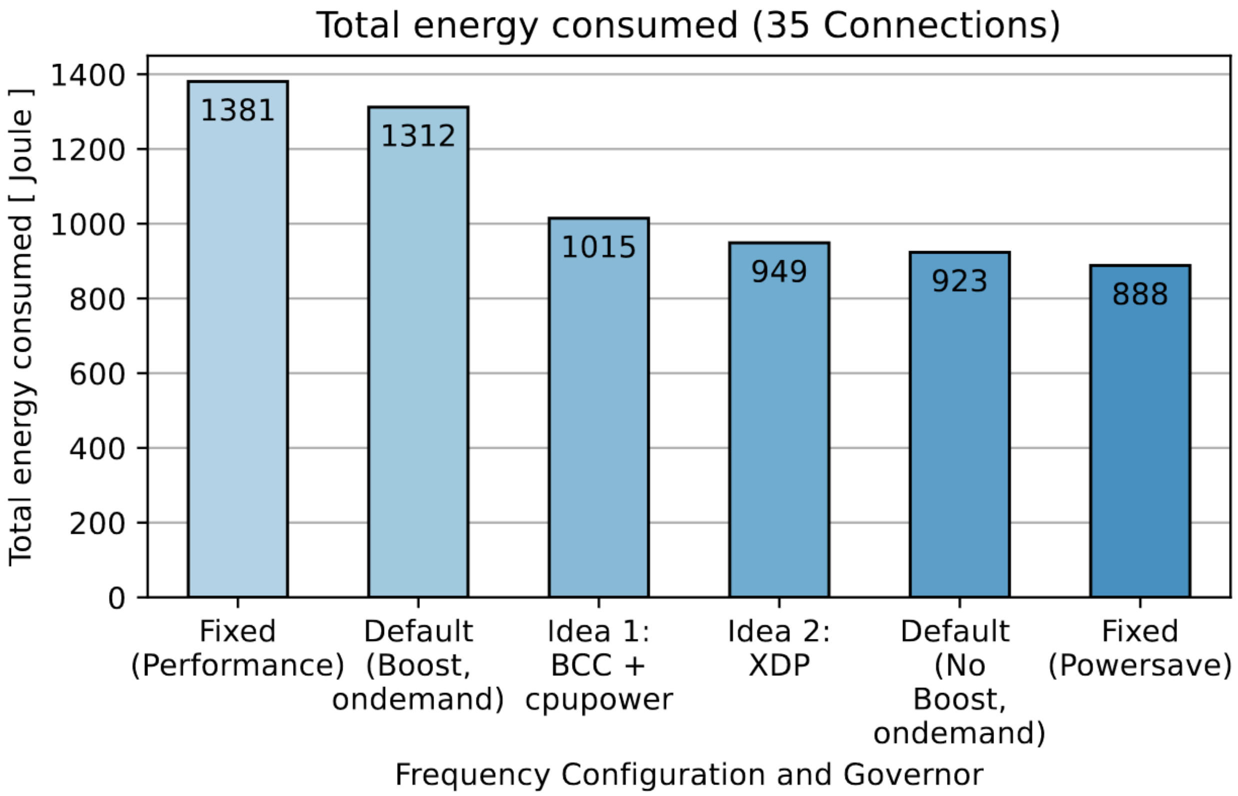
Evaluation: Setup

- System under test (SDN-Controller):
  - Ryzen 3700U with 4 cores, SMT, 15W maximum power draw
  - 3 frequencies available: 1.4GHz, 1.7GHz, 2.4GHz
  - *Performance*, *Powersave*, *Ondemand* governors (Linux) available
  - RockyLinux with kernel version 6.3
  - Measuring power via Running Average Power Limit (RAPL) interface
    - turbostat tool
- Company employees:
  - 9x Windows 10 machines
  - Per gigabit ethernet connected
  - Repeatedly connecting and disconnecting via VPN as fast as possible

Evaluation: Stress Power Consumption

For 35 connections (~2:30min):

- 23% less power consumed compared to unmodified system
- XDP implementation requires 5% less power than BCC
- Disabling Turbo Core yields 2% less energy consumption than XDP
- Powersave mode requires least energy



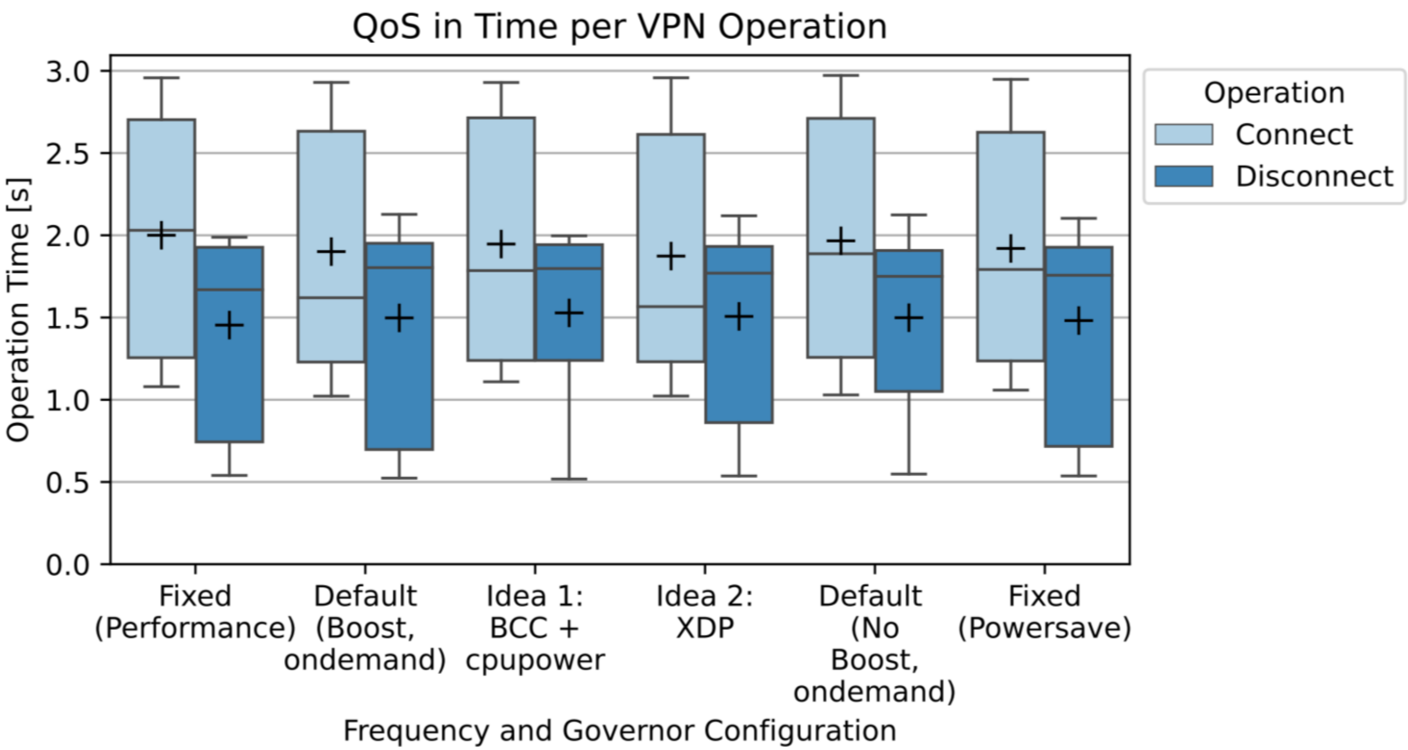
Evaluation: Idle Power Consumption

- Testing idle consumption for 2 minutes
  - Due to event based trigger low impact on idle consumption
  - Both designs only trigger when packet arrives
  - Otherwise, no code is executed

Mode	Default	XDP	BCC
Energy Consumption	256.62	262.02	266.62
Relative Consumption	100%	102%	104%

Evaluation: Quality of Service

- Time in seconds to connect/disconnect
- Black line: median operation time
- Cross: mean operation time
- Default powersaving mechanisms
  - 1-3% increase in mean connection time
- BCC
  - 2.4% mean connection time increase
- XDP architecture is fastest
  - 3% faster than the default configuration
  - 5% faster than the No Boost, ondemand configuration
  - Slow outliers degrade mean connection time



Selected References

1. Y. Liu et al., “Energy consumption and emission mitigation prediction based on data center traffic and PUE for global data centers,” Global Energy Interconnection, vol. 3, no. 3, pp. 272–282.
2. T. Favale et al., “Campus traffic and e-learning during COVID-19 pandemic,” Computer Networks, vol. 176, article 107290.
3. A. Feldmann et al., “The lockdown effect: Implications of the COVID-19 pandemic on internet traffic,” ACM, pp. 1–18., 2020
4. A. Shinoda, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura, “Implementation of access control method for telecommuting communication based on users’ reliability,” in Proceedings of Computer Security Symposium, pp. 840–847.,2022