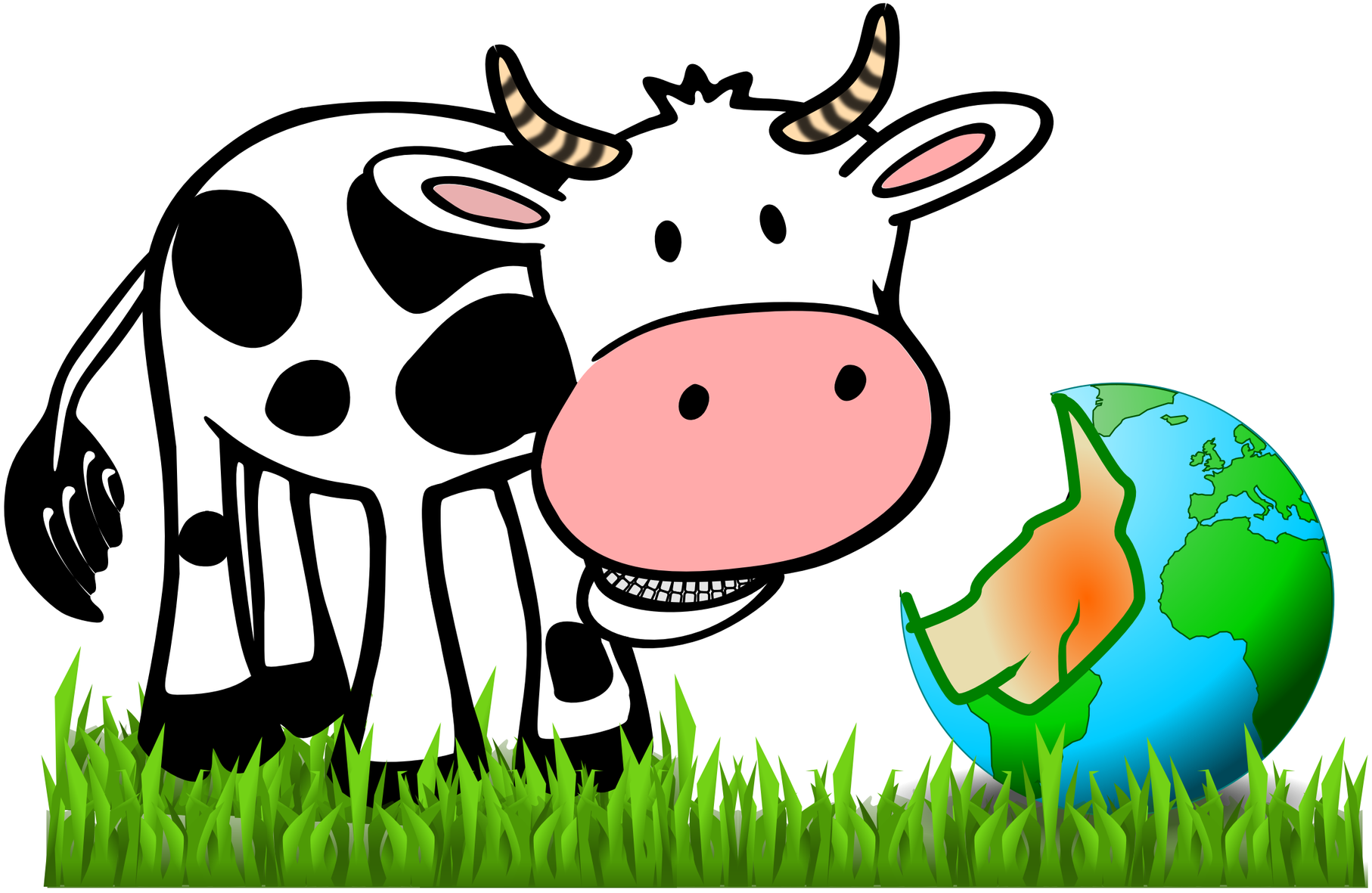
The Benefits of Open

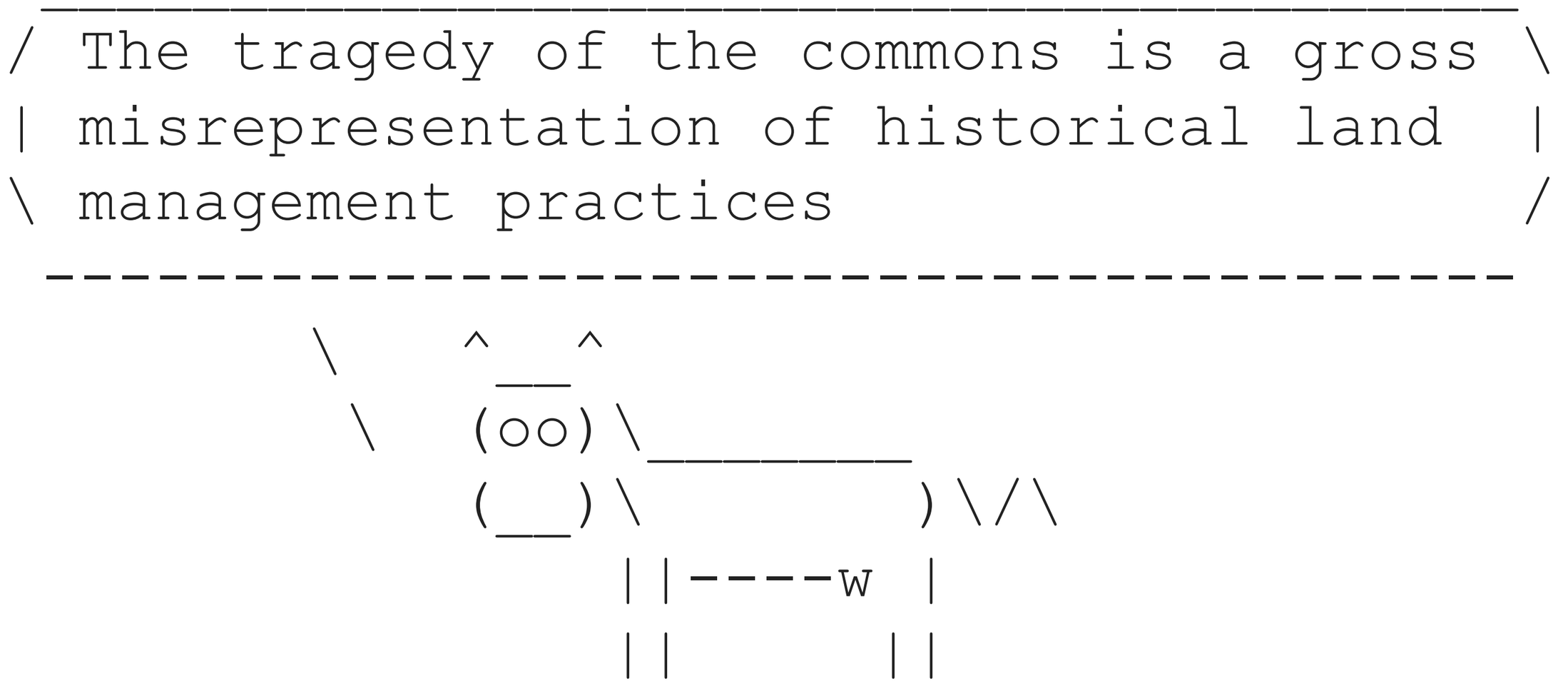
For Research and Open Source

Toke Høiland-Jørgensen
Principal Kernel Engineer, Red Hat

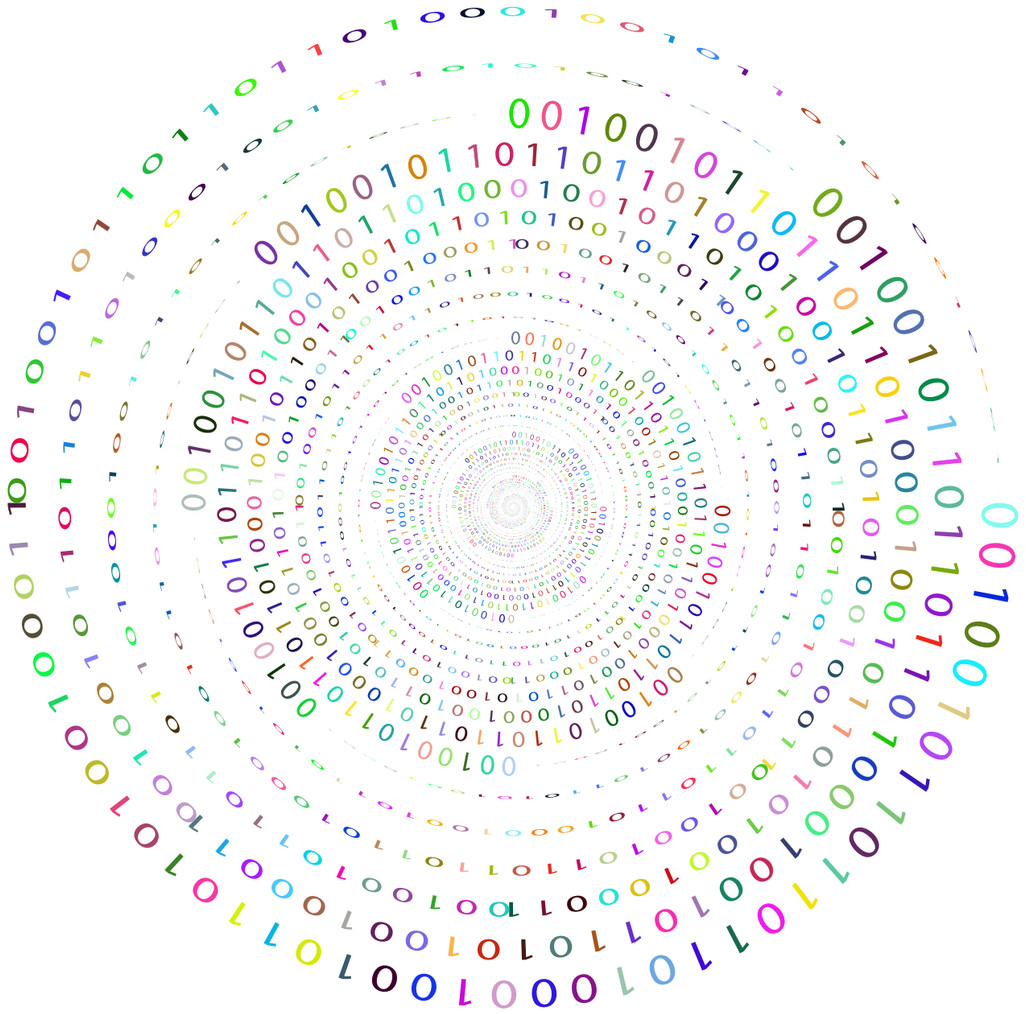
Why are you here?

Tragedy of the Commons



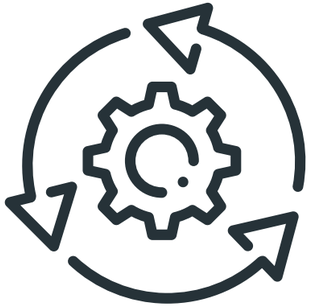

The tragedy of the commons is a gross misrepresentation of historical land management practices

The information-based commons

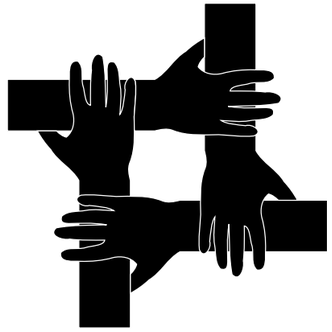


Benefits of openness

Accountability



Efficiency



Diversity

Research	**Open Source**
Knowledge	Software
Papers	Code
Peer review	Code review
Researchers	Developers/users
Journals/conferences	Code forges

Similarities (ideal)

- Openly available for anyone to use
- Peer review
- Anyone can participate
- Results disseminated into society

Failure modes

- Hard to become part of community
- Funding woes
- Obscurity
- Not reproducible / not really open

Benefits for researchers

- Feedback from experts outside research community
- Software widely used / results disseminated
- Code quality, production focus (varies)

Benefits for open source community

- New and original ideas, exploratory work
- Sound theoretical foundations
- More contributors

Examples

Ending the Anomaly: Achieving Low Latency and Airtime Fairness in WiFi

Toke Høiland-Jørgensen
Karlstad University

Michał Kazior
Tieto Poland

Dave Täht
TekLibre

Per Hurtig
Karlstad University

Anna Brunstrom
Karlstad University

Abstract

With more devices connected, delays and jitter at the WiFi hop become more prevalent, and correct functioning during network congestion becomes more important. However, two important performance issues prevent modern WiFi from reaching its potential: increased latency under load caused by excessive queueing (i.e. *bufferbloat*) and the 802.11 performance anomaly.

To remedy these issues, we present a novel two-part solution. We design a new queueing scheme that eliminates bufferbloat in the wireless setting. Leveraging this queueing scheme, we then design an airtime fairness scheduler that operates at the access point and doesn't require any changes to clients.

We evaluate our solution using both a theoretical model and experiments in a testbed environment, formulating a suitable analytical model in the process. We show that our solution achieves an order of magnitude reduction in latency under load, improves throughput [illegible]

affects the performance of WiFi bottleneck links. This is a well-known property of WiFi networks: if devices on the network operate at different rates, the MAC protocol will ensure *throughput fairness* between them, meaning that all stations will effectively transmit at the lowest rate. The anomaly was first described in 2003, and several mitigation strategies have been proposed in the literature (e.g., [13, 26]), so one would expect the problem to be solved. However, none of the proposed solutions have seen widespread real-world deployment.

Recognising that the solutions to these two problems are complementary, we design a novel queue management scheme that innovates upon previous solutions to the bufferbloat problem by adapting it to support the 802.11 suite of WiFi protocols. With this queueing structure in place, eliminating the performance anomaly becomes possible by scheduling the queues appropriately. We develop a deficit-based airtime fairness scheduler to achieve this.

Ending the Anomaly

What	Fixing WiFi performance anomaly and bufferbloat
Paper contribution	Design and implementation of solution
Type of code	Patches to Linux WiFi stack
Main challenge	Getting the low-level WiFi details to work

Efficient Continuous Latency Monitoring with eBPF

Simon Sundberg[1(✉)], Anna Brunstrom[1], Simone Ferlin-Reiter[1,2], Toke Høiland-Jørgensen[3], and Jesper Dangaard Brouer[3]

[1] Karlstad University, Karlstad, Sweden
{simon.sundberg,anna.brunstrom}@kau.se
[2] Red Hat, Stockholm, Sweden
sferlinr@redhat.com
[3] Red Hat, Copenhagen, Denmark
{toke,brouer}@redhat.com

Abstract. Network latency is a critical factor for the perceived quality of experience for many applications. With an increasing focus on interactive and real-time applications, which require reliable and low latency, the ability to continuously and efficiently monitor latency is becoming more important than ever. Always-on passive monitoring of latency can provide continuous latency metrics without injecting any traffic into the network. However, software-based monitoring tools often struggle to keep up with traffic as packet rates increase, especially on contemporary multi-Gbps interfaces. We investigate the feasibility of using eBPF to enable efficient passive network latency monitoring by implementing an evolved Passive Ping (ePPing). Our evaluation shows that ePPing delivers accurate RTT measurements and can handle over 1 Mpps, or correspondingly over 10 Gbps, on a single core, greatly improving on state-of-the-art soft-

Efficient Continuous Latency Monitoring with eBPF

What	Passive latency monitoring with eBPF
Paper contribution	Show feasibility of eBPF technology for application
Type of code	Standalone open source implementation
Main challenge	Getting code past the eBPF verifier

The eXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel

Toke Høiland-Jørgensen
Karlstad University
toke@toke.dk

Jesper Dangaard Brouer
Red Hat
brouer@redhat.com

Daniel Borkmann
Cilium.io
daniel@cilium.io

John Fastabend
Cilium.io
john@cilium.io

Tom Herbert
Quantonium Inc.
tom@herbertland.com

David Ahern
Cumulus Networks
dsahern@gmail.com

David Miller
Red Hat
davem@redhat.com

ABSTRACT

Programmable packet processing is increasingly implemented using kernel bypass techniques, where a userspace application takes complete control of the networking hardware to avoid expensive context switches between kernel and userspace. However, as the operating system is bypassed, so are its application isolation and security mechanisms; and well-tested configuration, deployment and management tools cease to function.

To overcome this limitation, we present the design of a novel approach to programmable packet processing, called the eXpress Data Path (XDP). In XDP, the operating system kernel itself pro-

ACM Reference Format:
Toke Høiland-Jørgensen, Jesper Dangaard Brouer, Daniel Borkmann, John Fastabend, Tom Herbert, David Ahern, and David Miller. 2018. The eXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel. In *CoNEXT '18: International Conference on emerging Networking EXperiments and Technologies, December 4–7, 2018, Heraklion, Greece.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3281411.3281443

1 INTRODUCTION

High-performance packet processing in software requires very tight bounds on the time spent processing each packet. Network stacks

The eXpress Data Path

What	High-performance programmable networking in the Linux kernel
Paper contribution	Describe XDP system design and show performance numbers
Type of code	Existing Linux subsystem
Main challenge	Coherently describing the existing subsystem

Tips for working with the open source community

- Clearly explain context: goals, assumptions, limitations
- Engage early and be ready to make adjustments
- Remember, it’s just code!

https://docs.kernel.org/process/development-process.html

Questions?