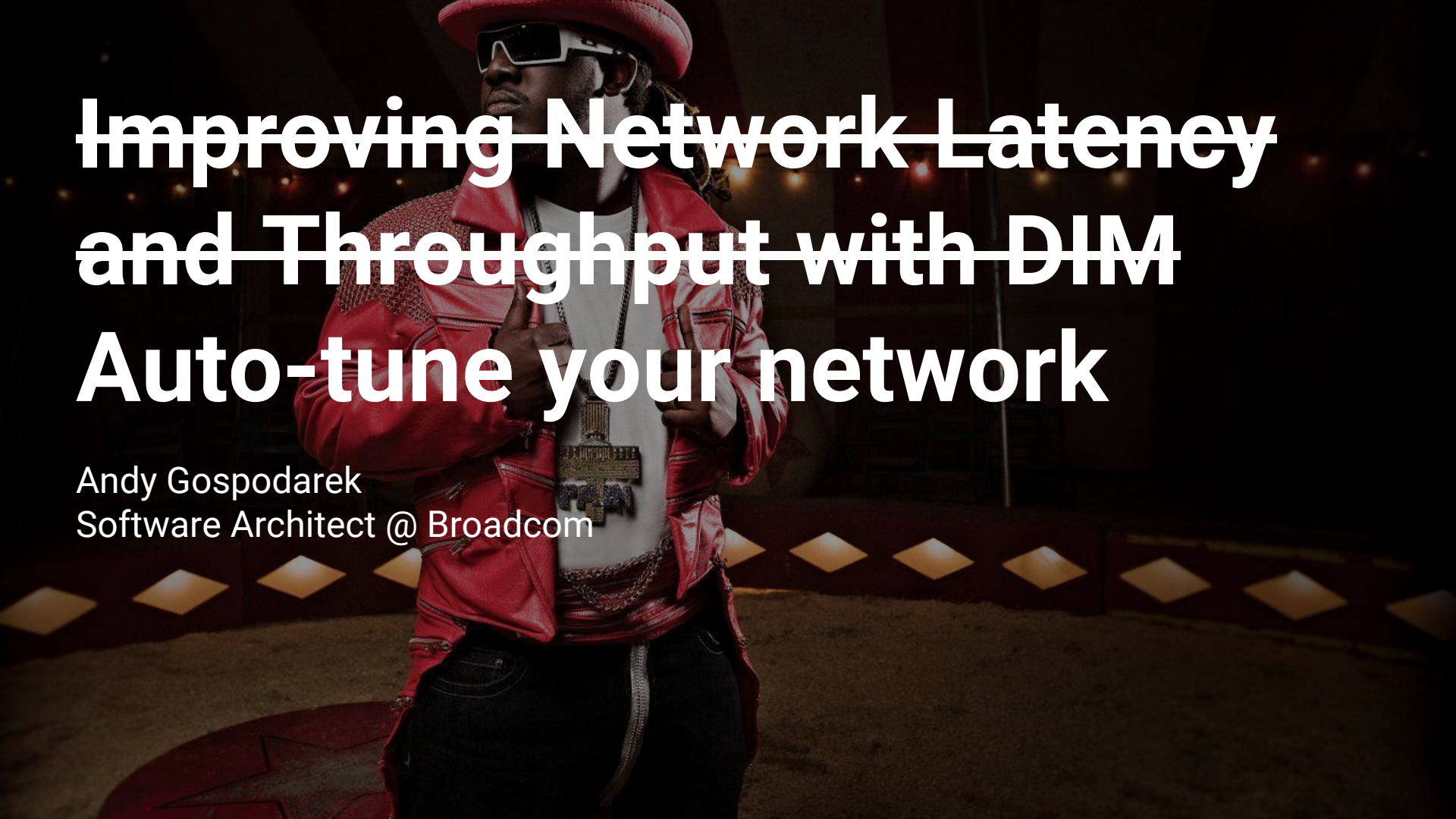
Improving Network Latency and Throughput with DIM

Andy Gospodarek
Software Architect @ Broadcom



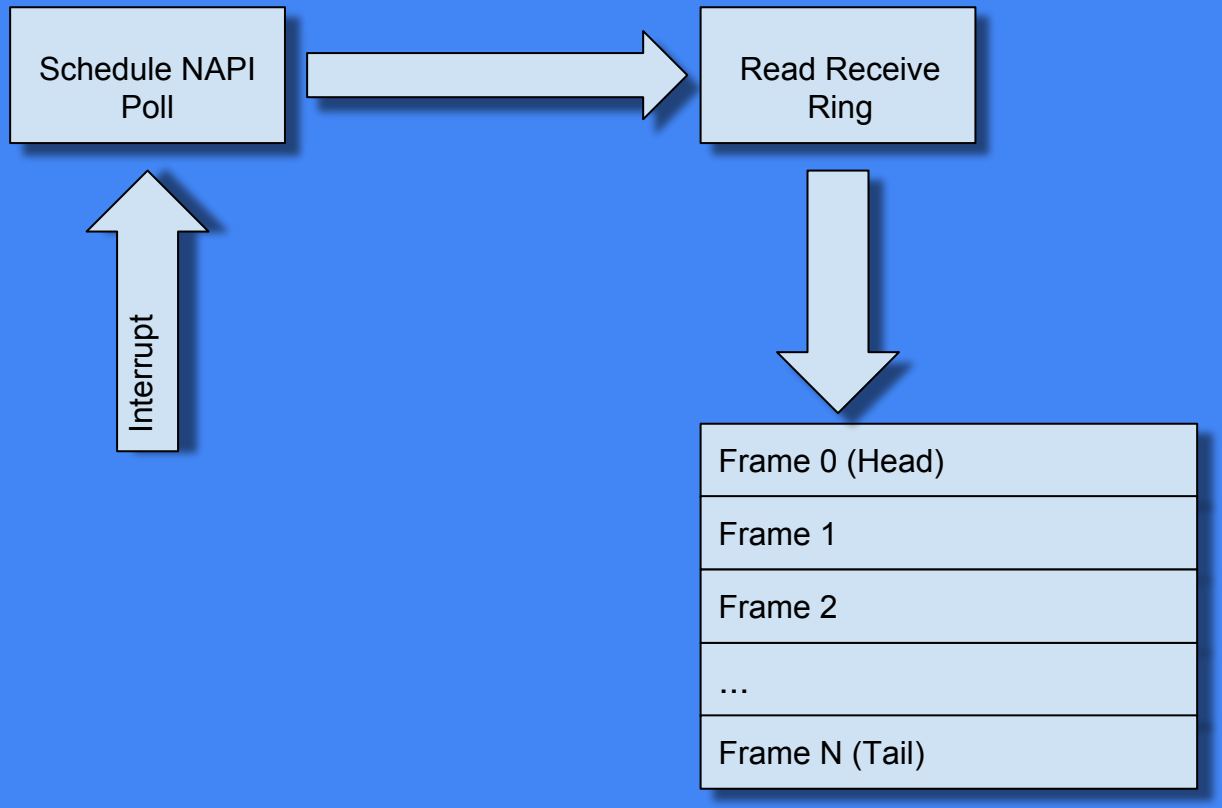
~~Improving Network Lateney and Throughput with DIM~~

Auto-tune your network

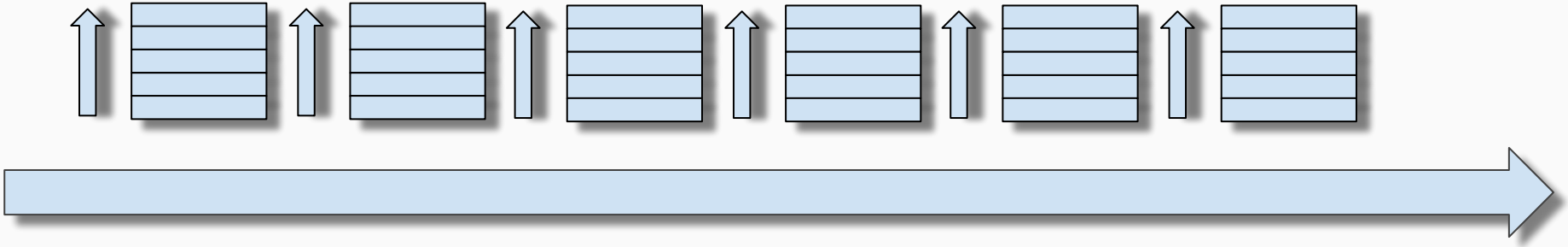
Andy Gospodarek
Software Architect @ Broadcom

Dynamic
Interrupt
Moderation

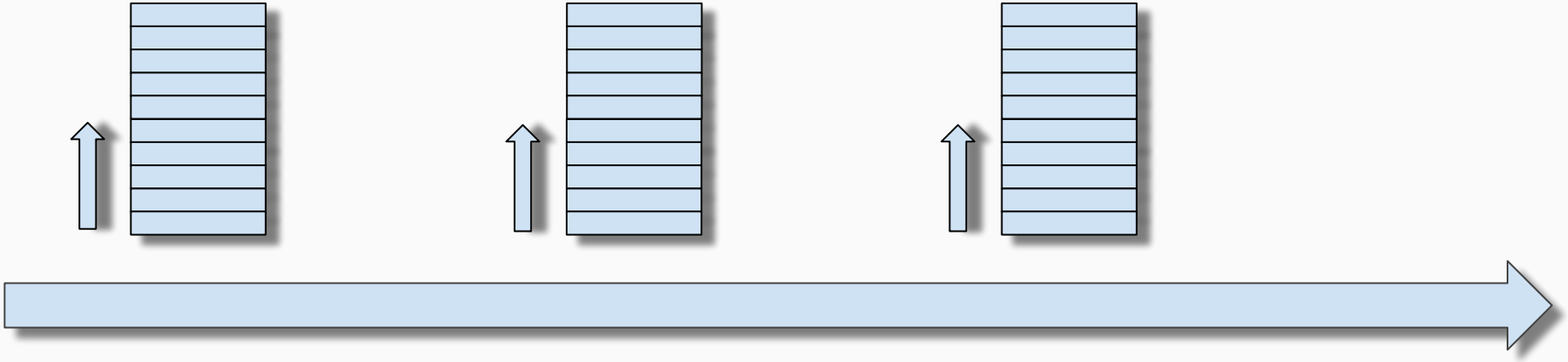
Tuning the time between when first frame arrives off the wire and when an interrupt pops


Schedule NAPI Poll
Read Receive Ring
Interrupt
Frame 0 (Head)
Frame 1
Frame 2
...
Frame N (Tail)

Short interrupt time means small number of frames read on each polling event



Double interrupt period, double number of frames received on each poll event



This is not a new problem

Admins have been tuning interrupt delay times for drivers for ~~years~~ decades

Intel Ethernet Adapters supported a feature called AIM -- Adaptive Interrupt Moderation

Liked by some, disabled by many

Hardware lacks flexibility available in software

Would a userspace daemon be helpful?

Considered having separate network tuning profiles to optimize for throughput or latency

Fast forward a few years....

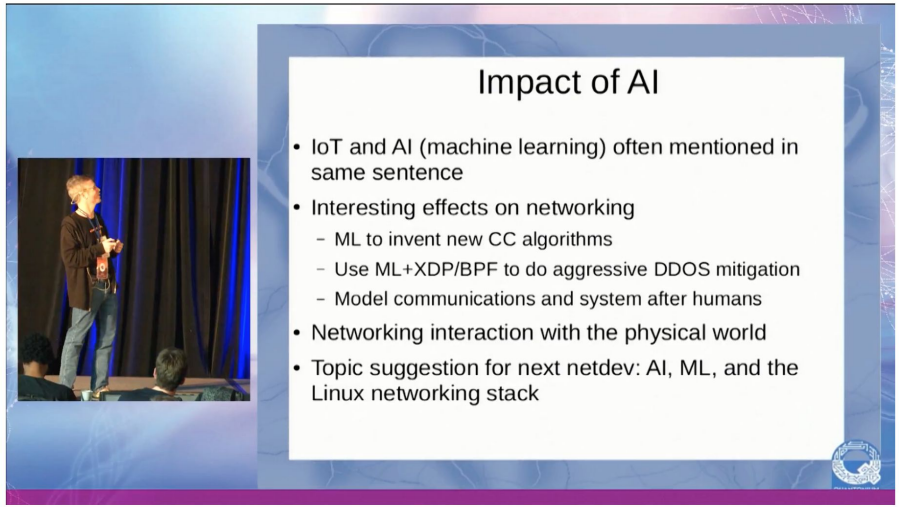
ML and AI are everywhere

Machine Learning for Protocols

At Netdev 2.1 in Montreal Tom Herbert wondered how Machine Learning will impact Linux kernel and protocols:

“Will [TCP] BBR be the last human-written congestion control algorithm?”

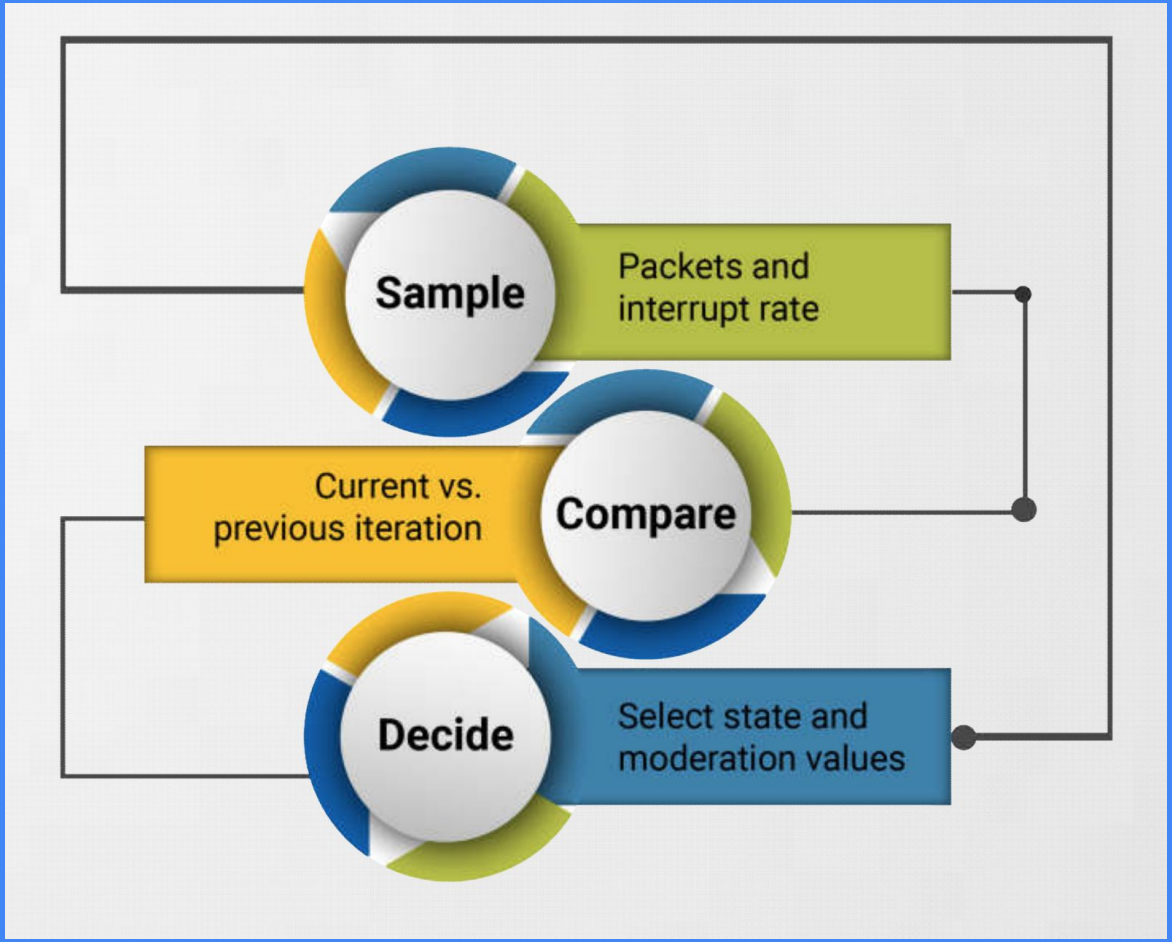
https://www.youtube.com/watch?v=mLDz-KnExiY



Netdev 2.1 - Real-time IoT networking on the Internet By Tom Herbert

Mellanox added support for DIM to `mlx5_core` in 2016

Data rates and interrupt rates are used to determine optimal interrupt timer settings in real-time

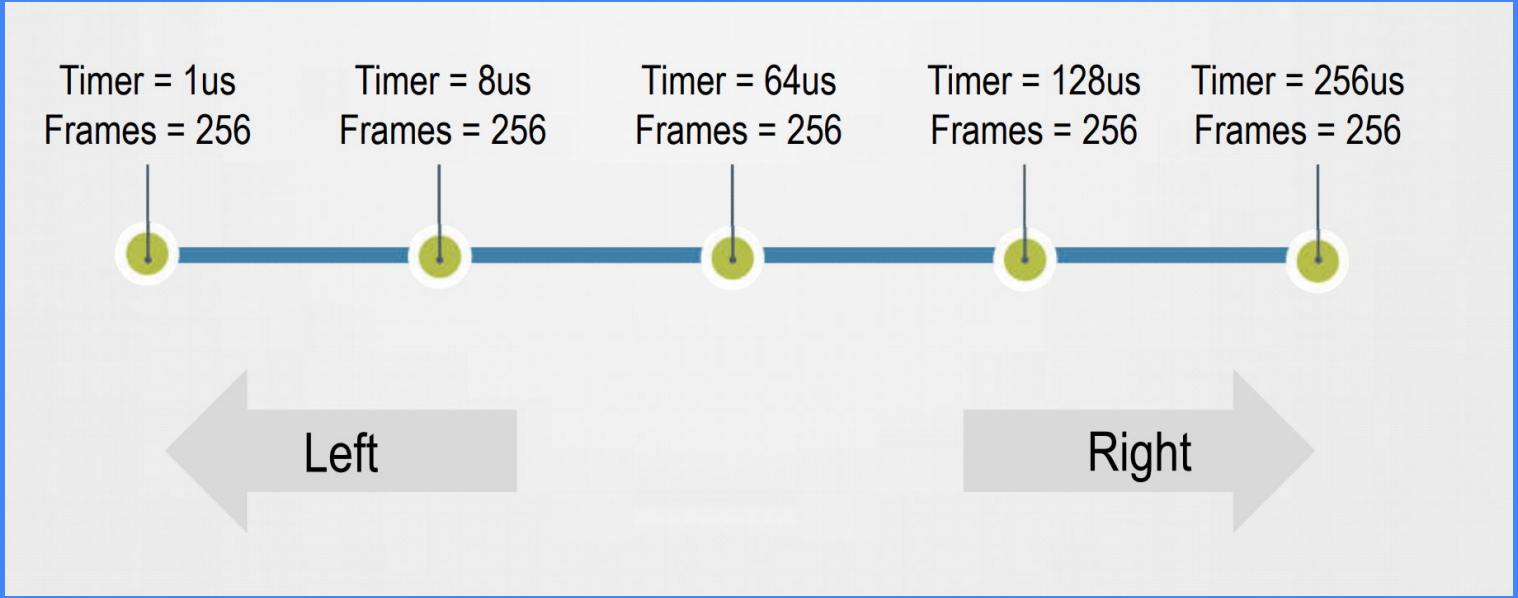


(Image credit Tal Gilboa)

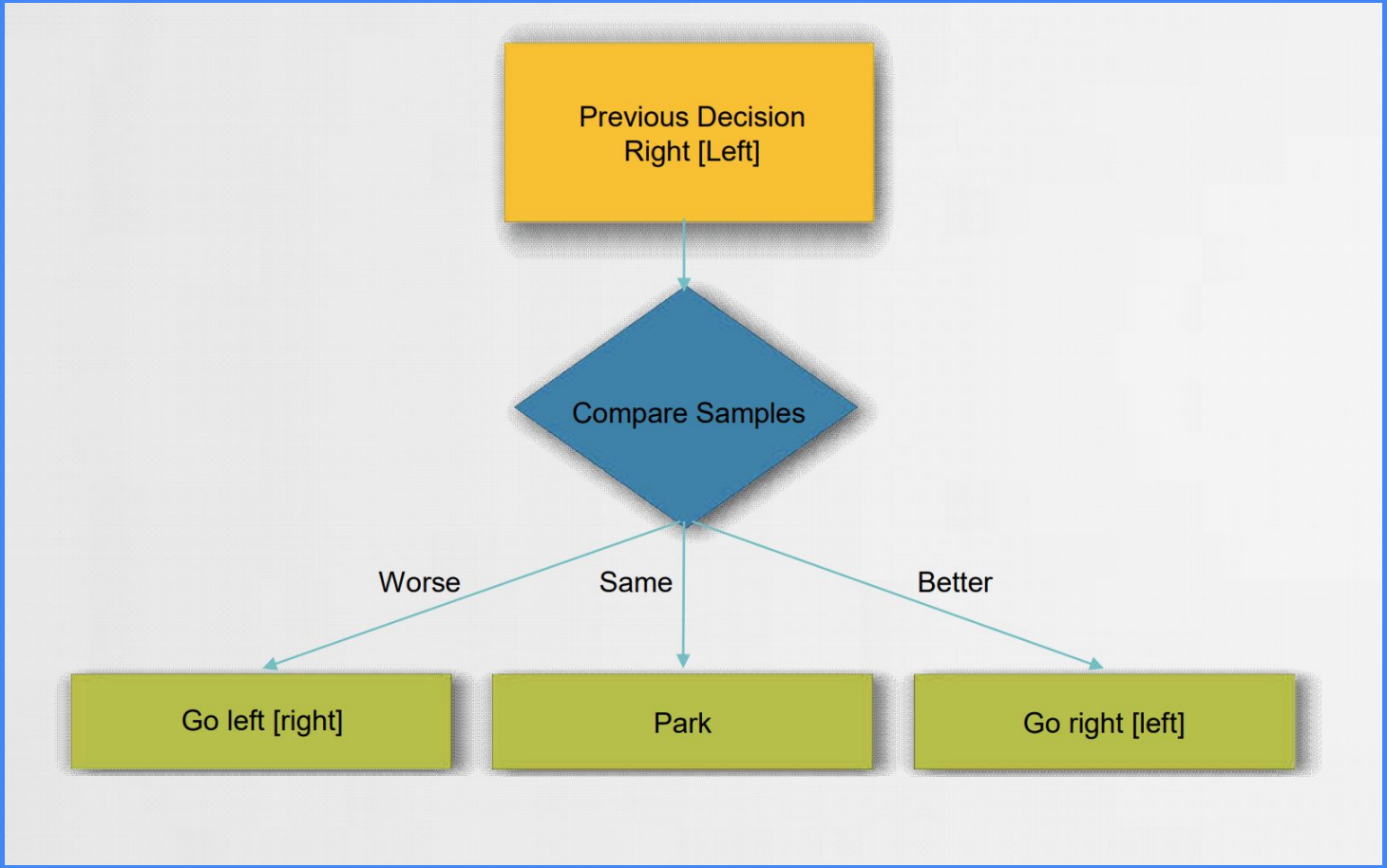
No longer locked into global settings used by ethtool API

DIM could also operate independently on receive rings so each core handling traffic could be optimally utilized

Each profile currently contains entries for minimum number of frames and minimum interrupt delay



(Image credit Tal Gilboa)

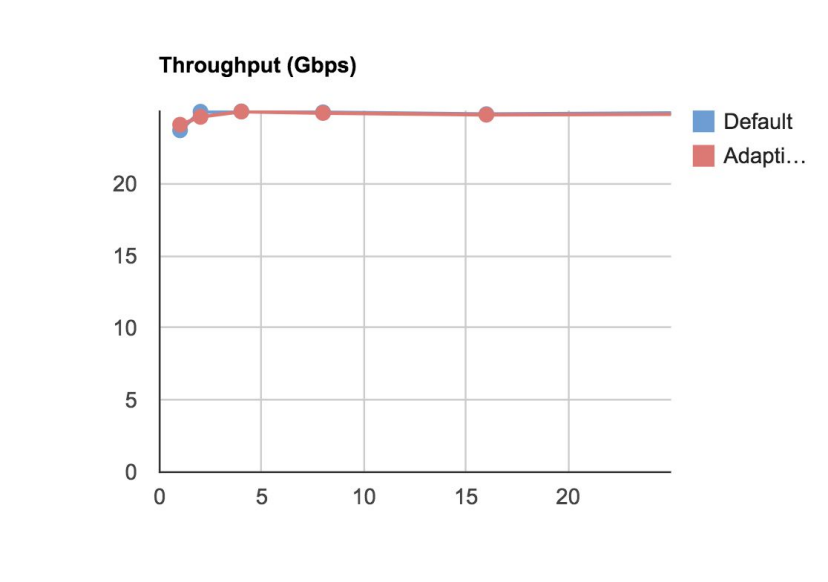


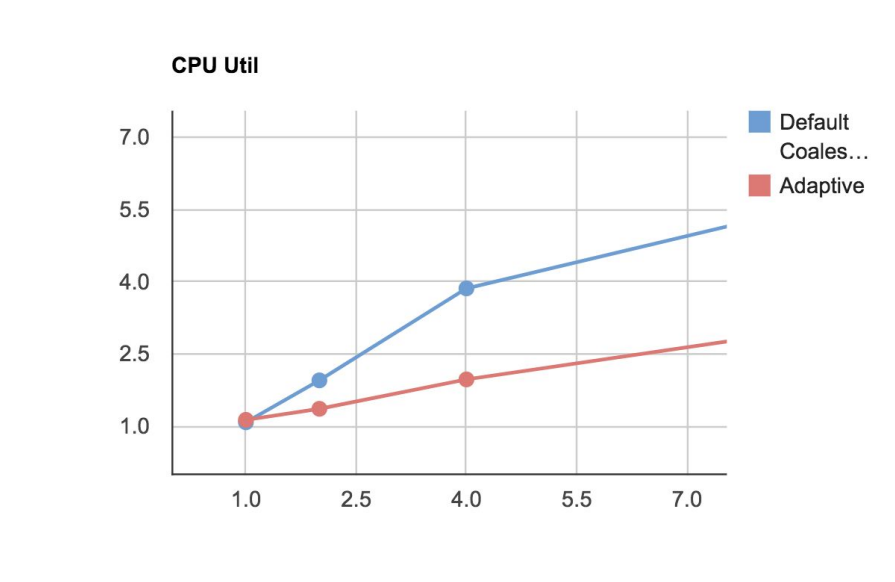
(Image credit Tal Gilboa)

This talk mentions Intel and Mellanox cards, what about Broadcom?

Ported and tested DIM to `bnxt_en` driver and liked the results

Improved CPU Utilization





Maintaining TCP_RR Performance

Static Coalescing	20,360 trans/sec
Adaptive Coalescing	19,513 trans/sec
Difference	~4% Reduction*

Confirmed that one receive ring can be optimized for low-latency and one for high-throughput.

Generic solution can be used by any driver included in upstream kernel in early 2018.

After upstream inclusion DIM added to `bcmgenet` driver

More drivers to follow???

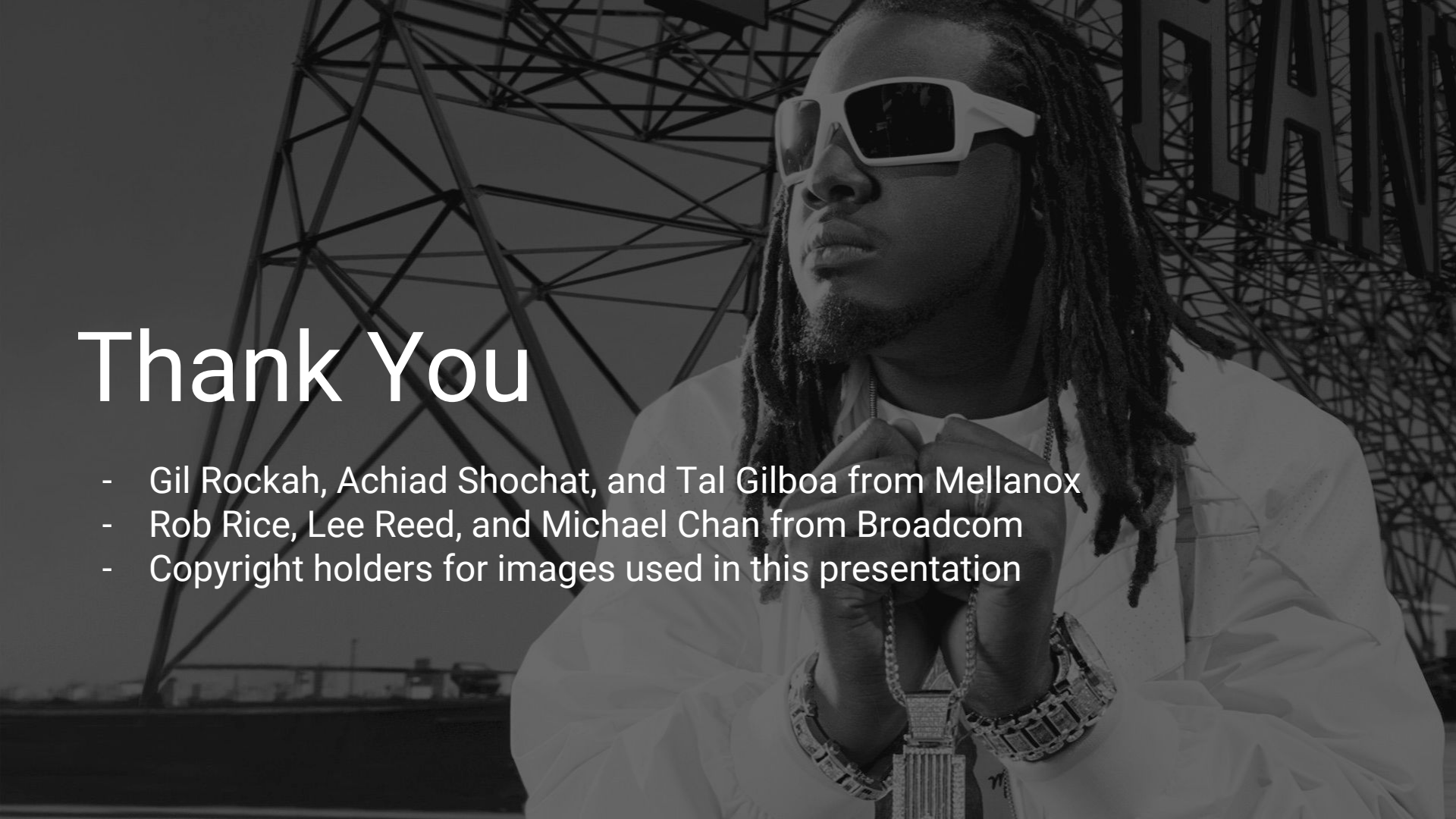
Observations -- some less surprising than others

Programming hardware can be expensive

Sometimes benefits appear unexpectedly

ACKs became seen as low-latency traffic and improved transmit performance

Real-time analysis and modification or kernel config options can be successful



Thank You

- Gil Rockah, Achiad Shochat, and Tal Gilboa from Mellanox
- Rob Rice, Lee Reed, and Michael Chan from Broadcom
- Copyright holders for images used in this presentation