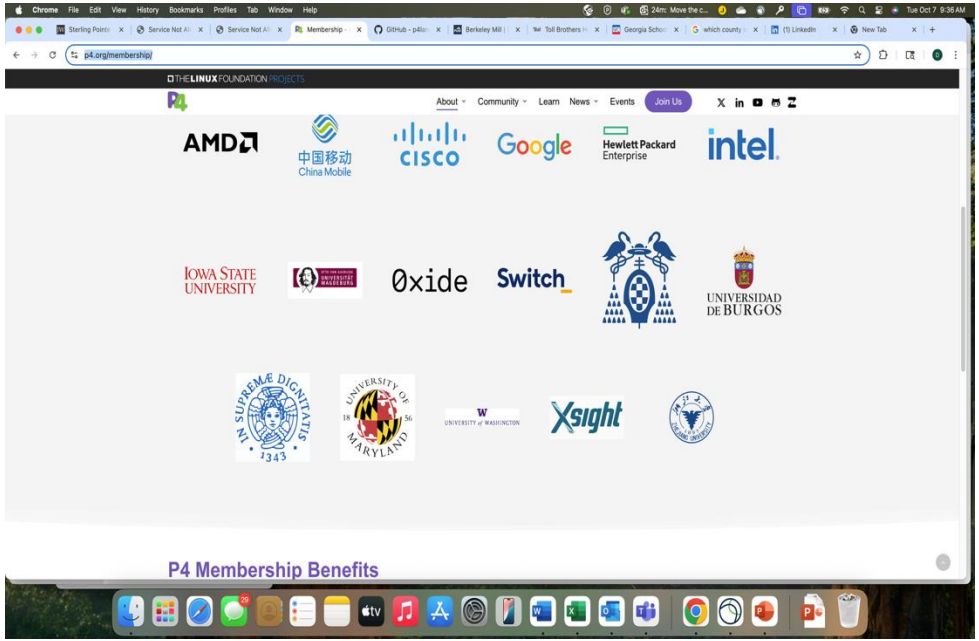
New Dawn of P4

Deb Chatterjee, Intel

THE LINUX FOUNDATION

We continue to have a vibrant community



- Many new members
- CISCO is now another premium member along with Google and Intel
- Discussion ongoing with several other companies
- Academic interest is still strong
- Open sourcing of Tofino SW components

List of P4-programmable devices expanding!

- Intel announced P4-programmable E2200 (400 Gbps IPU)
- AMD/Pensando Salina 400 Gbps NICs
- Xsight Lab's X-Switch family
- P4 programable FPGA products
  - AMD/Xilinx VitisNetP4
  - Renewed interest in the reborn Altera
- This is in addition to existing Tofino installations (EdgeCore DCS 810 and others) and other P4 devices

So what's the bad news?

- Perceived headwinds
  - Merchant ASIC roadmap and supply uncertainty
  - Access Barriers (SDKs, NDAs, tooling)
  - Toolchain fragmentation
  - Operational skills and verification
  - Competing paradigms

Additional challenges brought by AI boom

- Competition for silicon/investment bandwidth
- Mismatch of compute vs memory / resource constraints
- Diverging design priorities (AI vs programmable packet processing)
- Ecosystem gravitation toward AI stacks
- “Black box” closed loop preference vs transparency
- Potential misalignment in scaling and associated fear of obsolesce

Lots of new opportunities for P4

- Match/action as the common HW abstraction for high-speed packet processing, with P4 as the enabler.
  - Reduces Silicon respin risk compared to protocol-aware approaches
- P4 as the spec language (exemplary work by Steffen Smolka and others here)
- In-band Network Telemetry (INT/IOAM), postcard sampling, heavy-hitter & flowlet detection
- P4 as the fast feedback plane for ROCE/UEC style RDMA transports
- Opportunities in 5G/6G such as MEC service chaining with SRV6 SIDs in P4

Long term bets

- P4 in in-network compute primitives
- Data sovereignty using P4
- Better resource utilization and load balancing in AI networking, using P4
- Many other use cases

P4tify – generating target-specific P4 using generative AI

- P4 skill gap often seen as a barrier to adoption
- HW targets can perform better with hints provided through P4 annotation or other means, instead of fully generic P4
- But this immediately makes writing such HW-efficient P4 difficult
- P4tify is an attempt to autogenerate that code using LLMs (internal proof-of-concept project within Intel ongoing)
- Must guard against threat landscape and leaking of intellectual secrets

P4tify system architecture

Components

- Orchestrator Agent
- LLM Code Generator
- Compiler Feedback Loop
- Simulator/Test Harness
- Static Analyzer RAG Engine

Workflow

- Input
- Planning
- Code Skeleton
- Filling and Compilation
- Repair Loop
- Testing
- Output