P4 Expert Roundtable Series
April 28-29, 2020
Hosted by: ONF

Sponsored By
Google Intel STORDIS

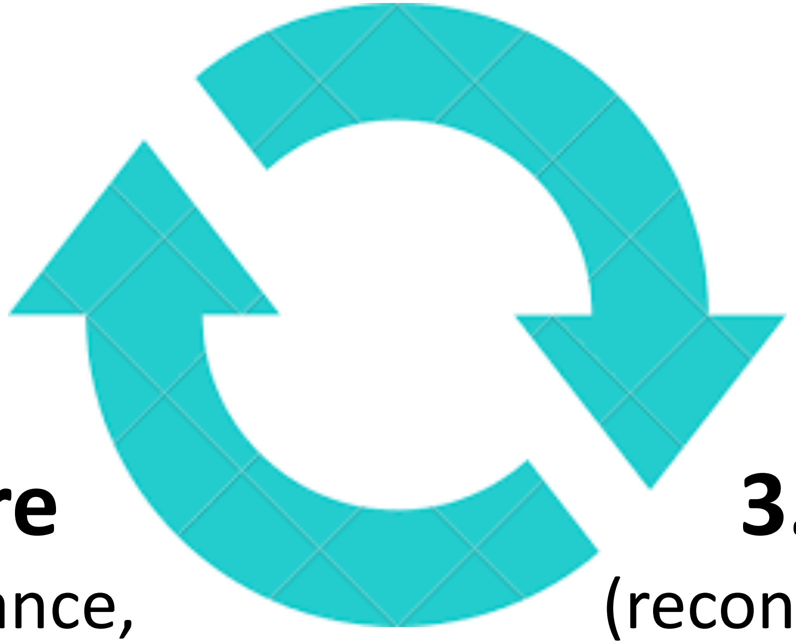
Closing the Loop: Network Control in the Data Plane

Jennifer Rexford
Princeton University

Traditional Network Management

2. Analyze

(traffic matrix, route optimization, anomaly detection, fault localization)



1. Measure

(load, performance, traffic, failures)

3. Configure

(reconfigure tunnels, link weights, access control lists)

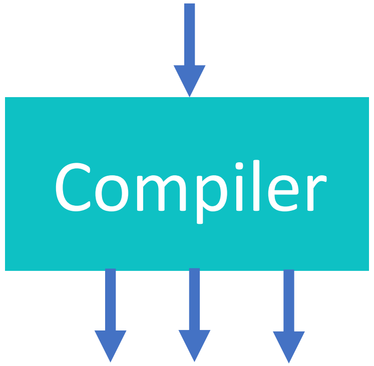
Limitations of Traditional Network Management

- Measurement mismatches
  - Overhead of collecting many device-local measurements
  - Simplistic statistics (counts, samples) not tailored to the task
- Indirect control
  - Configuration of complex protocols and mechanisms
  - Separate configuration of many distributed devices
- Complex reasoning
  - Many separate software components and protocols
  - Software bugs, configuration errors, and protocol interactions

Closing the Loop

- An integrated approach

Network-wide goals
(objectives and constraints)



Device-local programs
(measure and control)

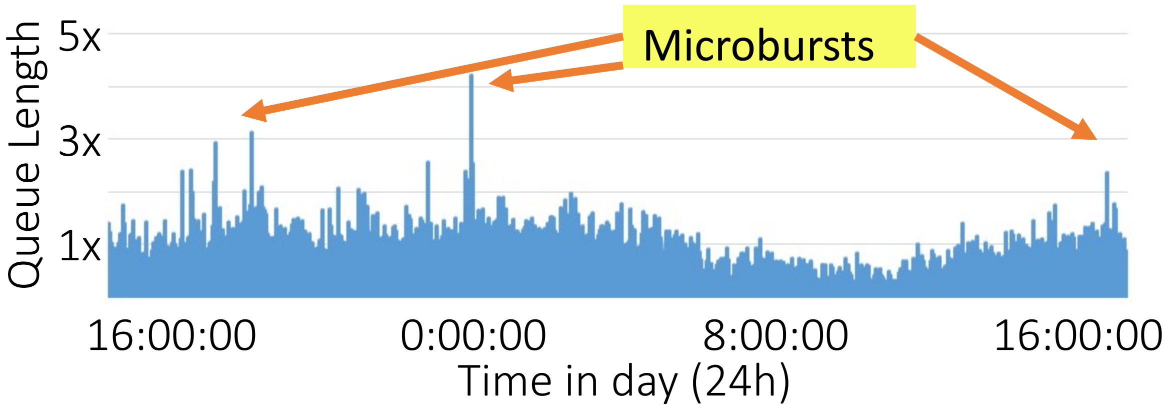
- Efficient measurement
  - Measurement tailored to the task
  - Data analysis in the data plane
- Direct control
  - Control actions in the data plane
- Correct by construction
  - Device-local programs synthesized from network-wide goals

Example Network-Wide Goals

- Alleviating microbursts
- Performance-aware routing
- DDoS mitigation
- Block hosts with old OSes
- <Insert your “app” here>

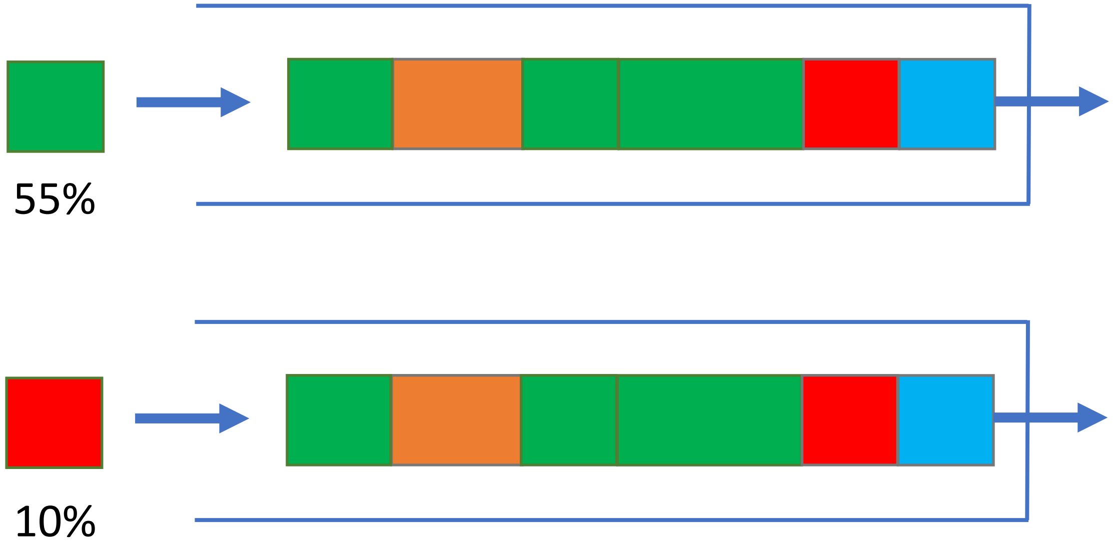
Alleviating Microbursts: ConQuest [CoNext'19]

- Small timescale traffic bursts
  - Clog the packet queues
  - Cause packet delay and loss
- Manage microbursts to handle
  - Bursty workloads
  - Low-cost switches (shallow buffers)
  - High link utilization
- Goal: penalize the most responsible flows



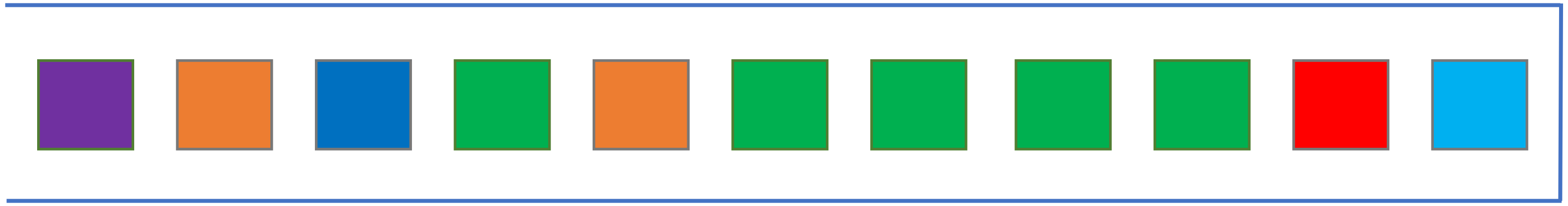
Microburst Management Policy

- Active queue management
  - Mark each packet probabilistically
  - In proportion to its flow's contribution to the heavy queue



Detecting Heavy Flows in the Queue

- For each flow, how many packets are in the queue?

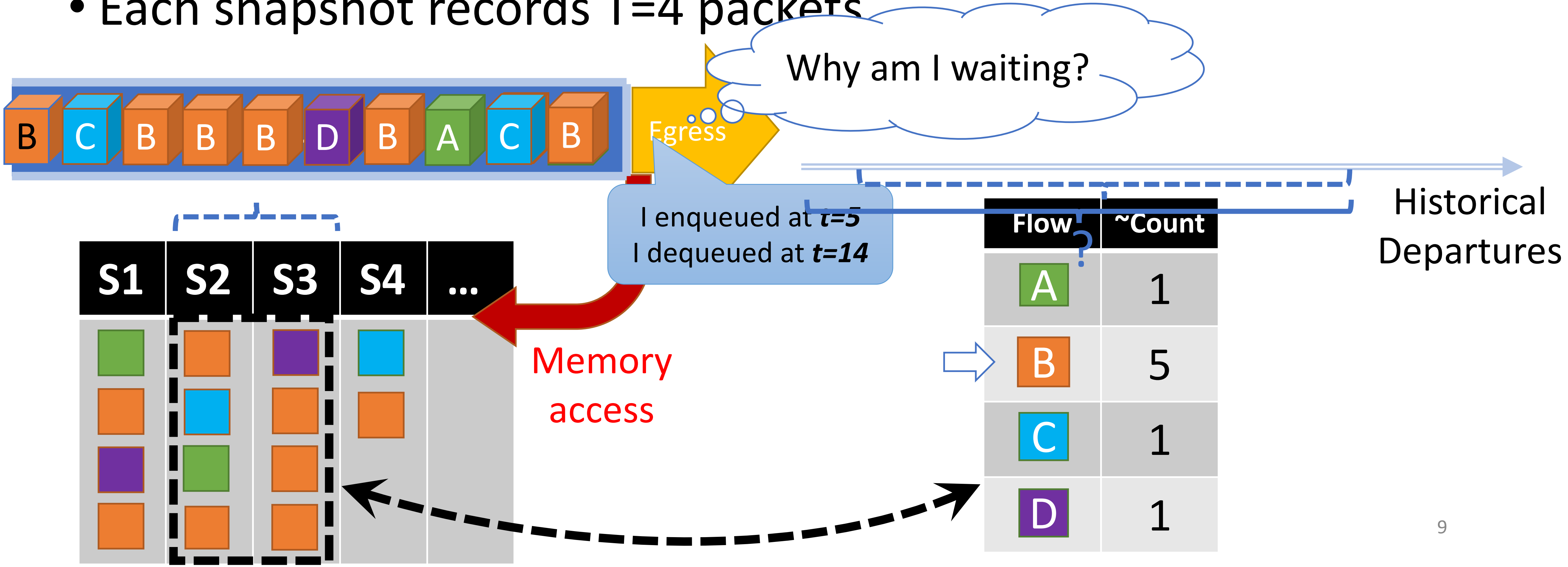


Key	Count
	1
	5
	1
	1
	2
	1

- P4 data structure challenges
  - Updating on packet arrival and departure
  - Per-flow state (key and count)

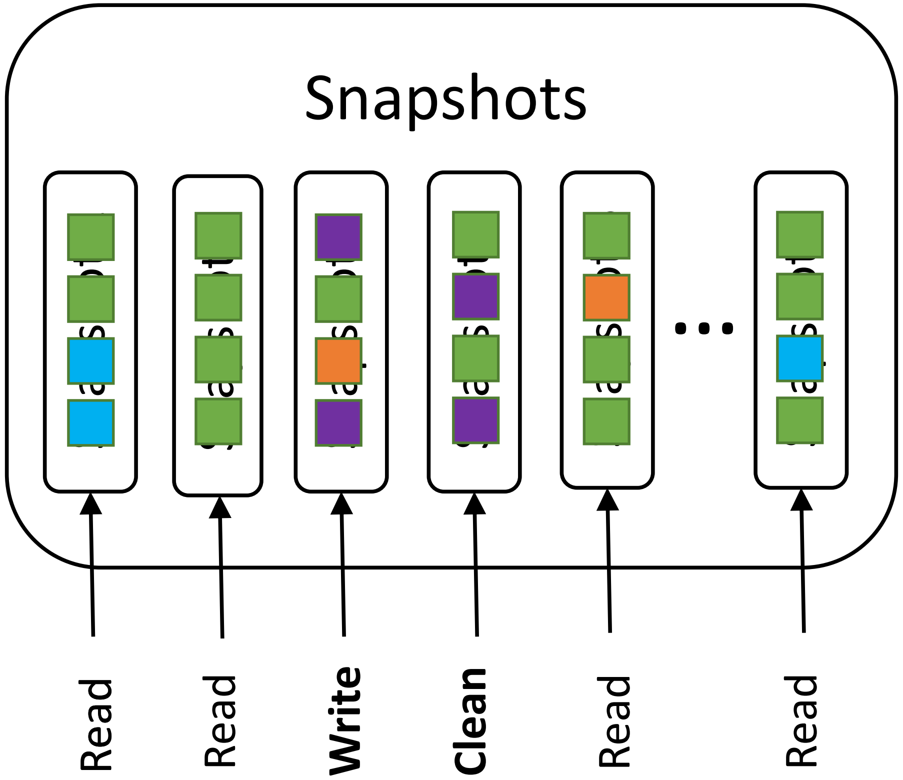
ConQuest: Processing Each Packet Only Once

- Slice traffic into time windows
- Each snapshot records T=4 packets



ConQuest: Round-Robin Snapshots

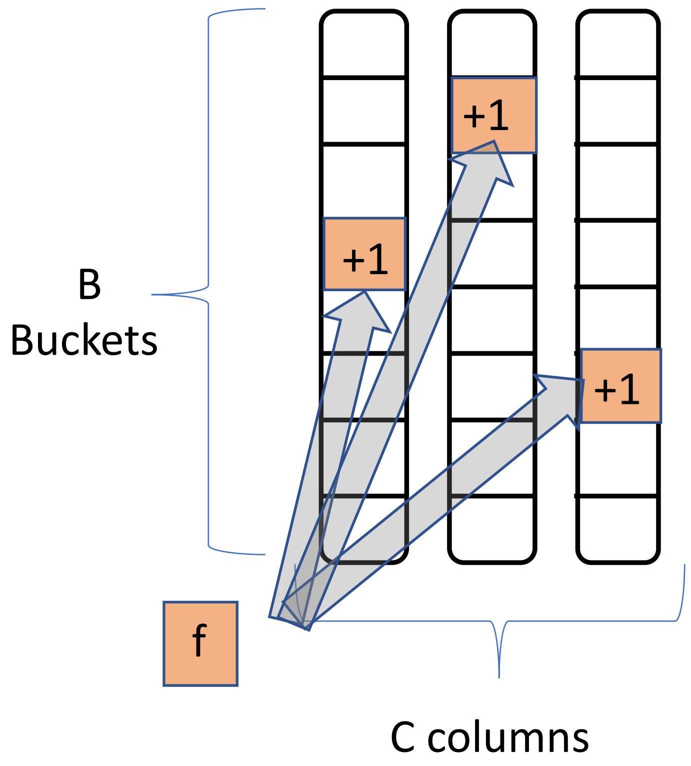
- ***Clean*** and ***Reuse*** the snapshots in the data plane!



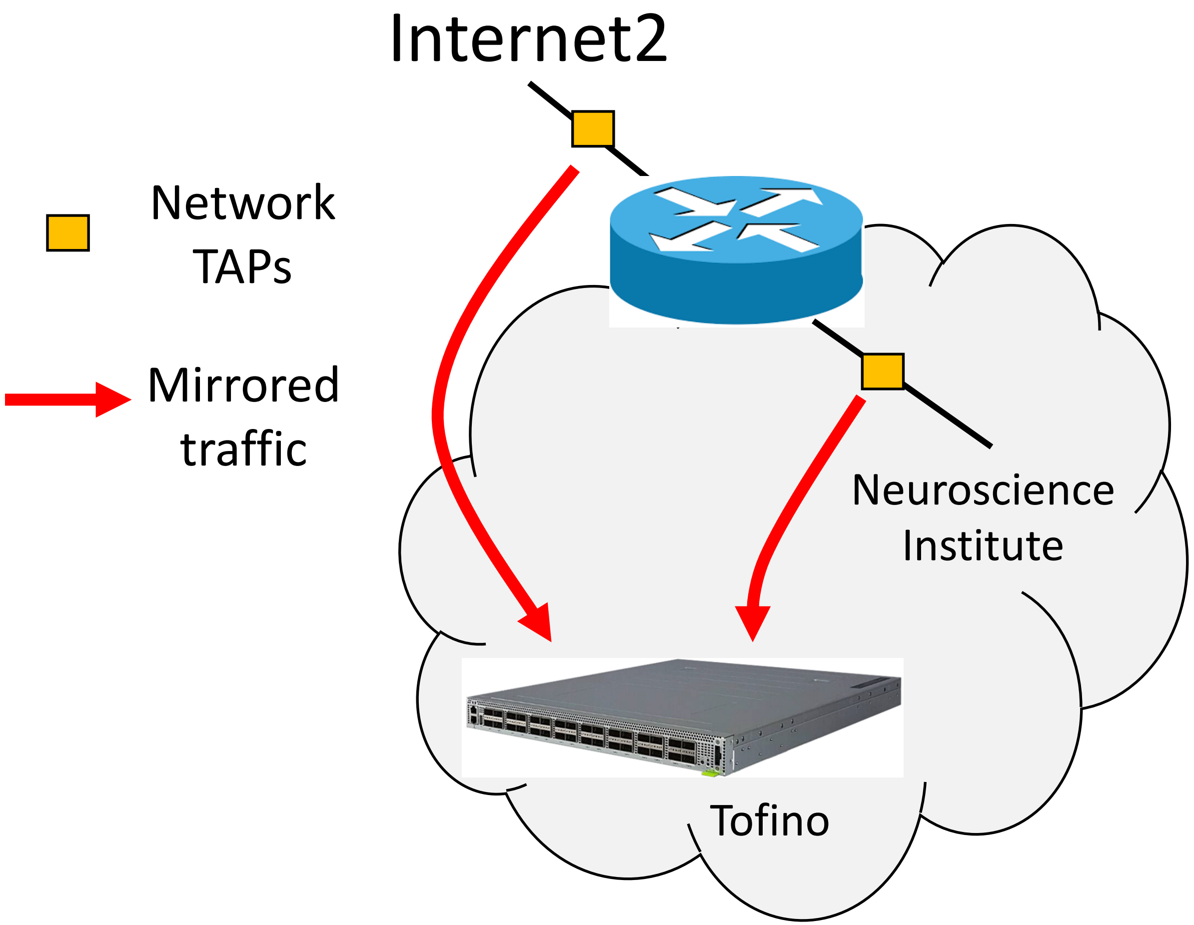
ConQuest: Avoiding Per-Flow State

- Compact data structures
  - With limited memory and processing
  - … at the expense of lower accuracy
- Estimate per-flow counts per snapshot
  - With accurate estimates for large flows
- Count-Min Sketch per snapshot
  - C columns indexed by hash functions
  - Increment $hash_i$(flowid) in column I
  - Estimate is the min of the C counts

Count-Min Sketch [CM ‘05]



ConQuest in Action on the Princeton Campus

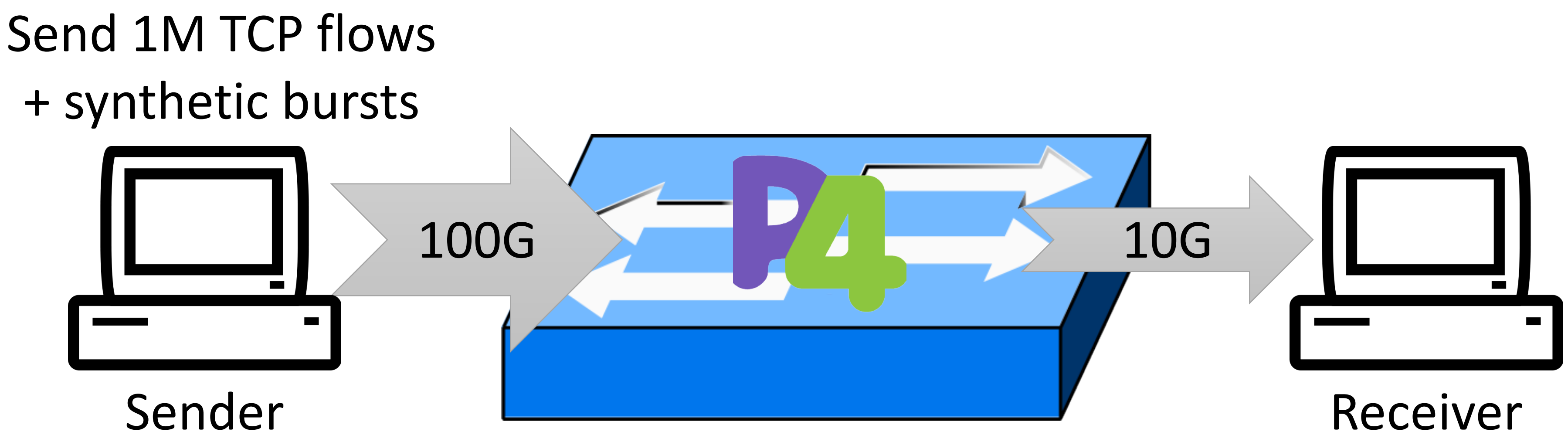


- Performance symptoms
  - Big neuroscience data transfers
  - High loss with low average load
  - Router with limited measurements
- Microburst analytics on Tofino
  - Microbursts on a small timescale
  - Caused by PerfSONAR active probes
- Recent deployment with AT&T

See https://p4campus.cs.princeton.edu/ web site for more details!

ConQuest: Closing the Control Loop

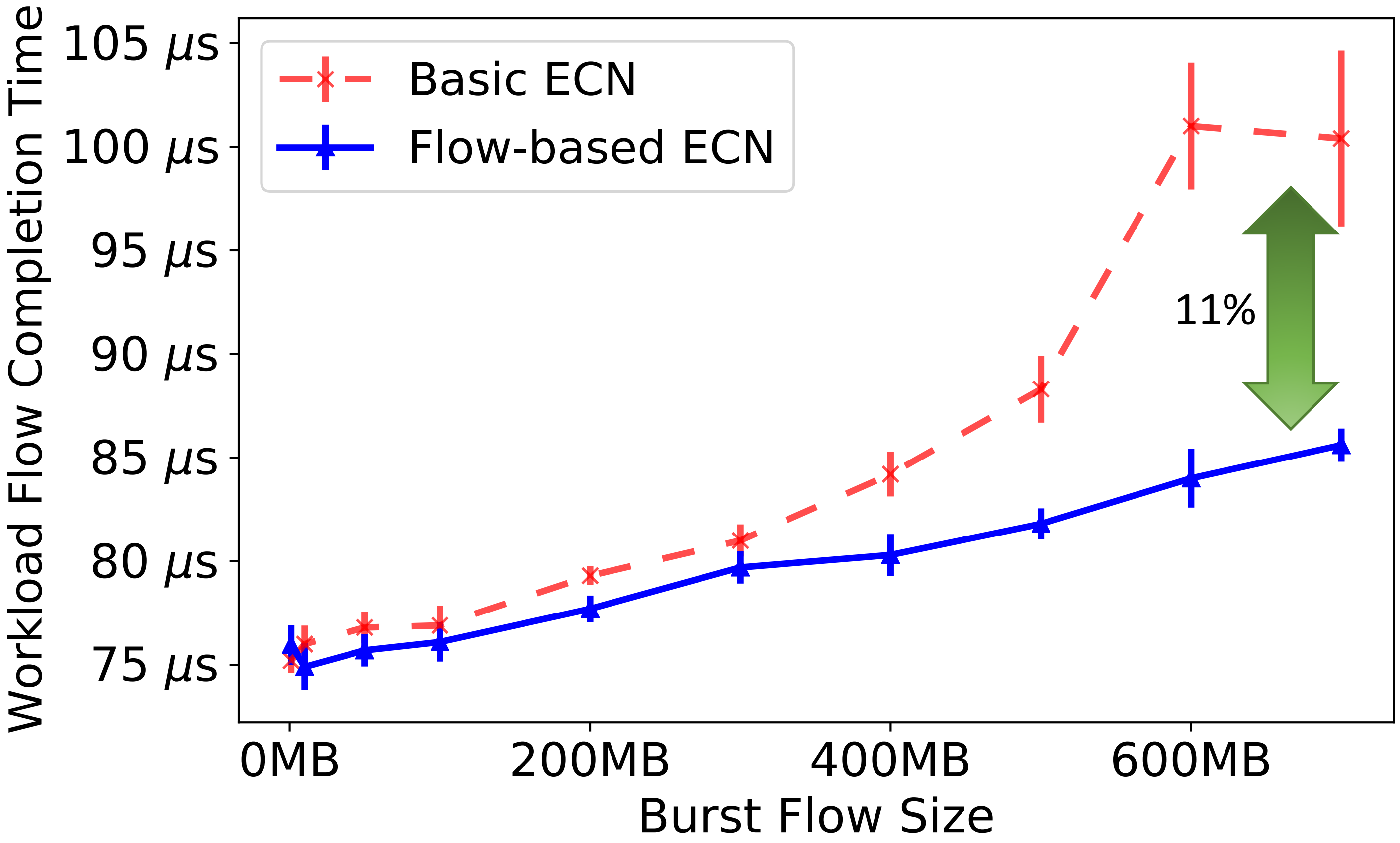
- Testbed with P4-enabled Barefoot Tofino, 4 snapshots
- Smart early congestion notification (ECN)
  - Baseline: mark all packets when the queue is long
  - ConQuest: flow-based ECN to mark flows causing others to wait



ConQuest: Evaluating the Control Loop

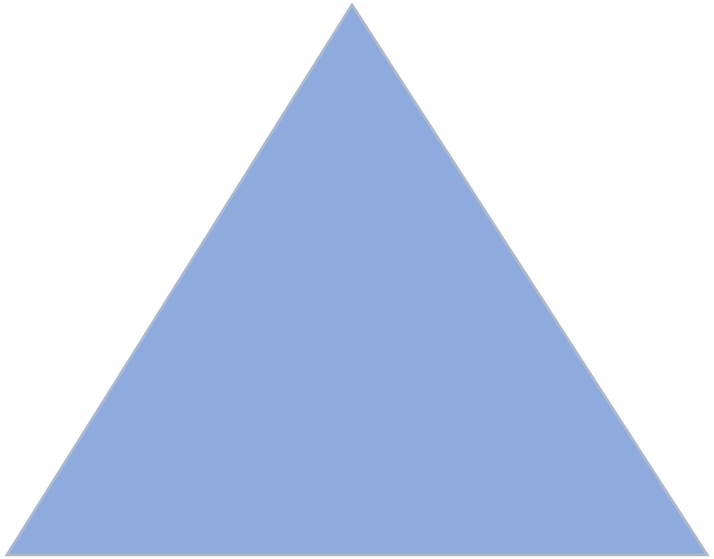
✓Flow-based ECN reduces Flow Completion Time

✓Queue remains short, bursty flow effectively suppressed



Performance-Aware Routing: Contra [NSDI'20]

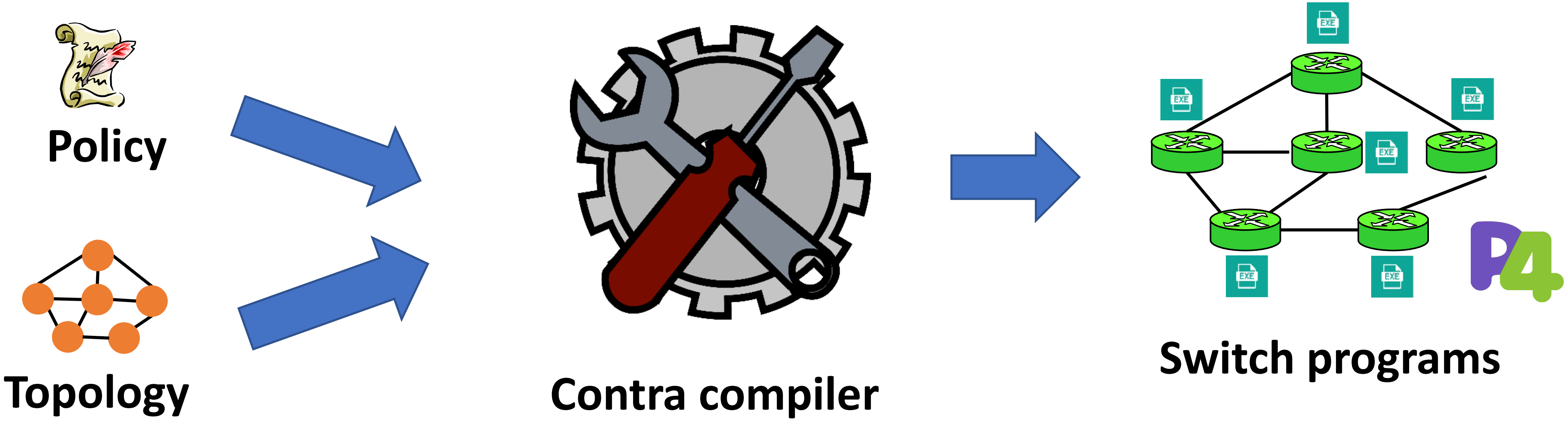
G1: Traffic Engineering
e.g., prefer least utilized paths



G2: Routing Constraints
e.g., middlebox traversal

G3: Fast Adaptation
e.g., update path choices upon performance changes

Contra: Performance-Aware Routing



- **Language + Compiler**: Compiles rich, high-level policies
- **Runtime**: Performance-aware routing protocol in the data plane

Contra Policy Language

- Routing policy: a function that ranks network paths
  - Matching on paths using regular expressions
  - Computing and comparing path metrics

Waypoint W with min utilization

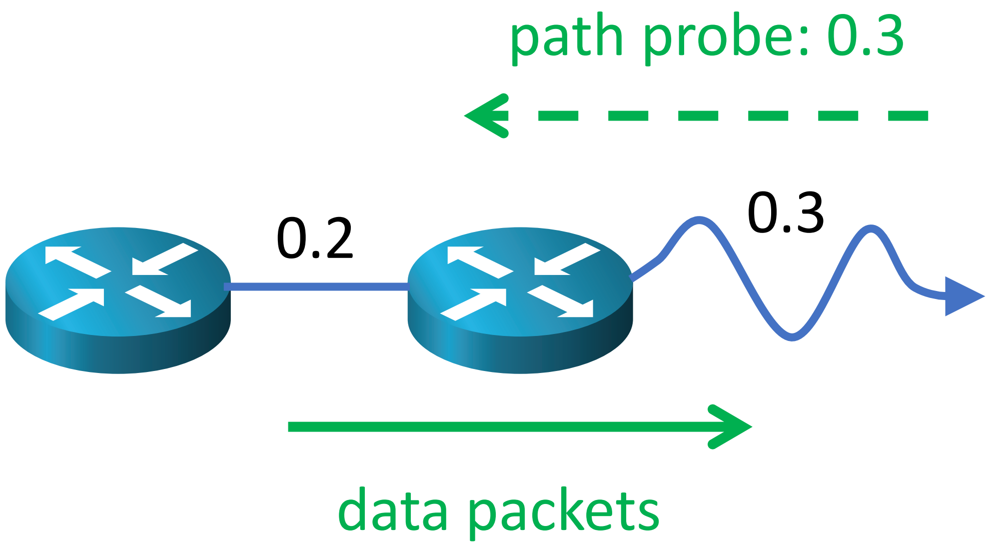
```
if (.* W .*)
  then path.util
  else ∞
```

Min utilization under light load, otherwise shortest

```
if (path.util  < 0.8)
  then (1, 0, path.util)
  else  (2, path.len, path.util)
```

Contra Family of P4 Routing Programs

- Distance vector routing
  - Flexible path constraints and metrics
  - Implementable in modern P4 data planes

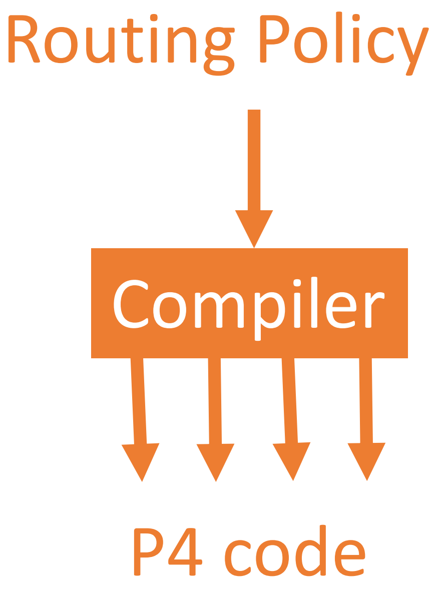


Contra P4 Building Blocks

- Monitor path performance
  - Reverse-path probes collect and accumulate statistics
- Enforce path constraints
  - Controlling the propagation of probes
- Compare and select paths
  - Best-path table updated as new probes arrive
- Avoid out-of-order packets
  - Policy-aware flowlet switching table
- Prevent forwarding loops
  - Version numbers of probes (like DSDV and Babel)

Contra Prototype and Evaluation

- Contra compiler
  - Written in 7485 lines of F#
  - Generates the switch-local P4 programs
- Experimental setup
  - Topologies: data centers, random graphs, ISPs
  - Performance metric: flow completion time (FCT)
  - Comparisons: equal-cost multipath, Hula, and SPAIN
- Simulation (in ns-3) and testbed (in CloudLab)
  - Outperforms shortest-path routing and static load balancing



Toward Verified Closed-Loop Control

- Explore more control-loop examples
  - DDoS mitigation, blocking unwanted OSes, etc.
- Evaluate under realistic conditions
  - Hardware switches and operational networks
- Identify unifying language constructs
  - Traffic queries integrated with control actions
- Verify the compiler
  - Ensure the P4 programs are ``correct by construction”

P4 Expert Roundtable Series

April 28-29, 2020

Hosted by: ONF

Sponsored By

Google intel STORDIS The Open Networking Expert

Thank You

https://www.cs.princeton.edu/~jrex
https://p4campus.cs.princeton.edu

P4 Expert Roundtable Series

April 28-29, 2020

Hosted by: ONF

Sponsored By

Google intel STORDIS The Open Networking Expert

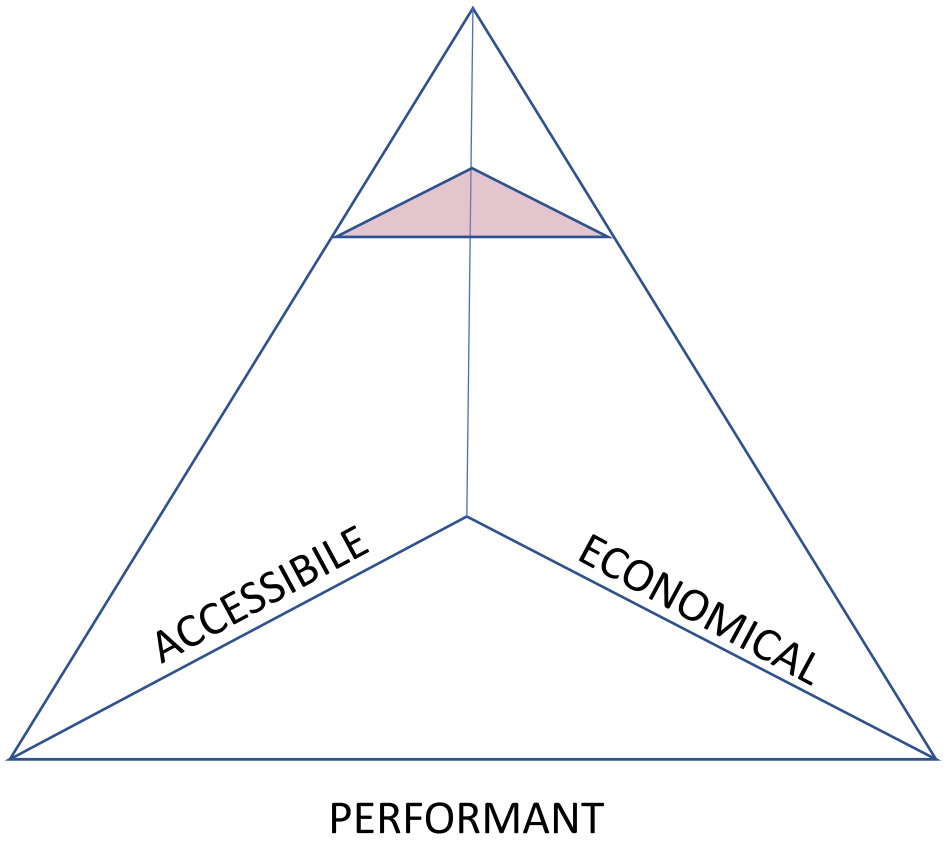
Programmability, Data and Actionable Network Analytics

Daniel Alvarez
Barefoot Networks (Intel company)

First a Detour…

The GOAL of any TECHNOLOGY is to become THE PLATFORM

Reaching PLATFORM status



- Three core parameters
- Requirements are dynamic, F(t)
- Need to MAX() all core parameters to gain and retain status
- Any dramatic shift in requirements can topple the platform

Perspective: 30+ years of IP/TCP

Its been a long cycle

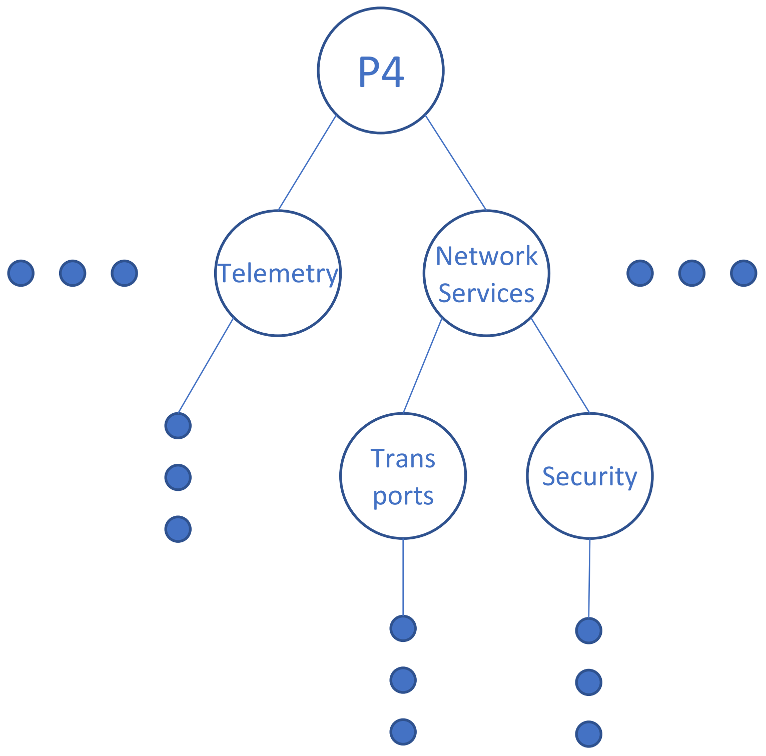
- We have been here before, 25 years ago the dataplane was fully programmable (SW based switching)
- Platform status results in reduced innovation
  - Relative slow pace of innovation in L3 and L4 layers
    - Luckily TCP has been a very versatile and adaptable transport
  - Above Layers benefit from programmability (linux, user space)
  - Below Layers driven by speed and cost
- New workloads and frameworks are pushing the capabilities of TCP
  - Need more efficient, framework specific transports
  - Advanced congestion control
  - New Reliable Transports enabled by dataplane programmability (TCP will also adapt)

The Next Platform (L3 -> L5) Programmable DataPlane + P4

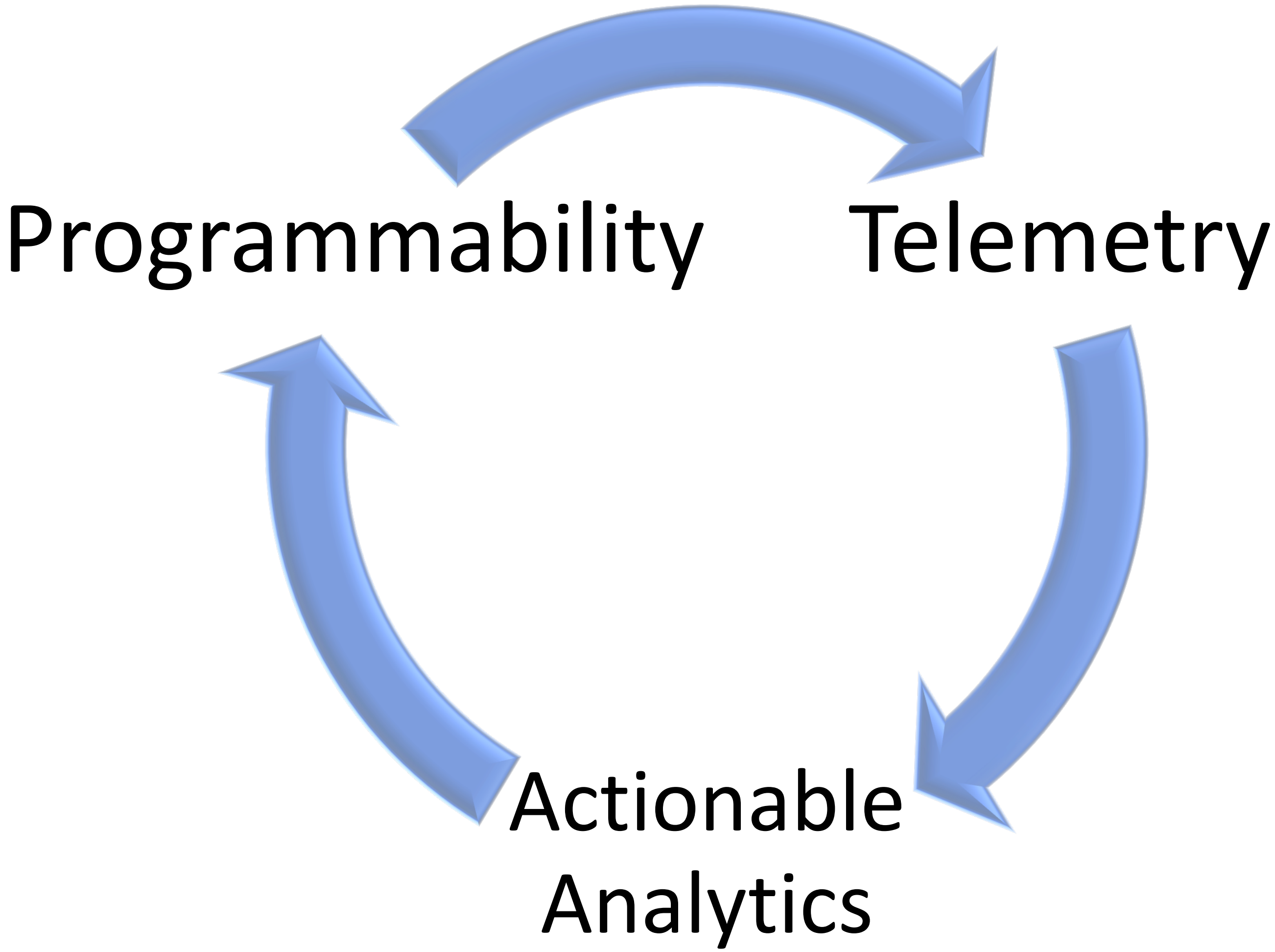
- Accessibility
  - P4 and P4.org
  - Smartnics, Programmable switches, SW dataplanes
- Performance and Economy
  - Programmable network chips at par in cost, speeds and feeds with fixed function

First round of innovations are leading indicators for a potential Cambrian explosion

Any one of these threads can revolutionize the industry



TELEMETRY: Networking as a Data Problem



- We are behind, BUT..
  - New network data sources
  - Real-time Analytics
  - Leverage Data Analytics and ML industry knowledge
  - End to End
- Telemetry
  - Visibility
  - Intrinsic Insights
  - Actionable
- Feed the loop
  - New Discoveries
  - New Applications

TELEMETRY: Path to True Intent

Partial Intent – Today's Intent Based Networking Solutions provide static policy and provisioning validation of the Network

Driving DevOps best practices into networking

Adaptive Intent

- Data Plane feedback
- Control Plane feedback
- Load/Performance feedback
- Security feedback

Measurement and Data - Telemetry and Analytics are the enablers for True Intent based networking