EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

FACULTY OF
SCIENCE
Communication Networks

Enhancements to P4TG: Protocols, Performance, and Automation

Fabian Ihle, Etienne Zink, Steffen Lindner, Michael Menth

http://kn.inf.uni-tuebingen.de

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Agenda

- P4TG Overview
- Motivation
- Enhancements to P4TG
  - Protocols
  - Performance
  - Automation
- Conclusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

P4TG – Overview

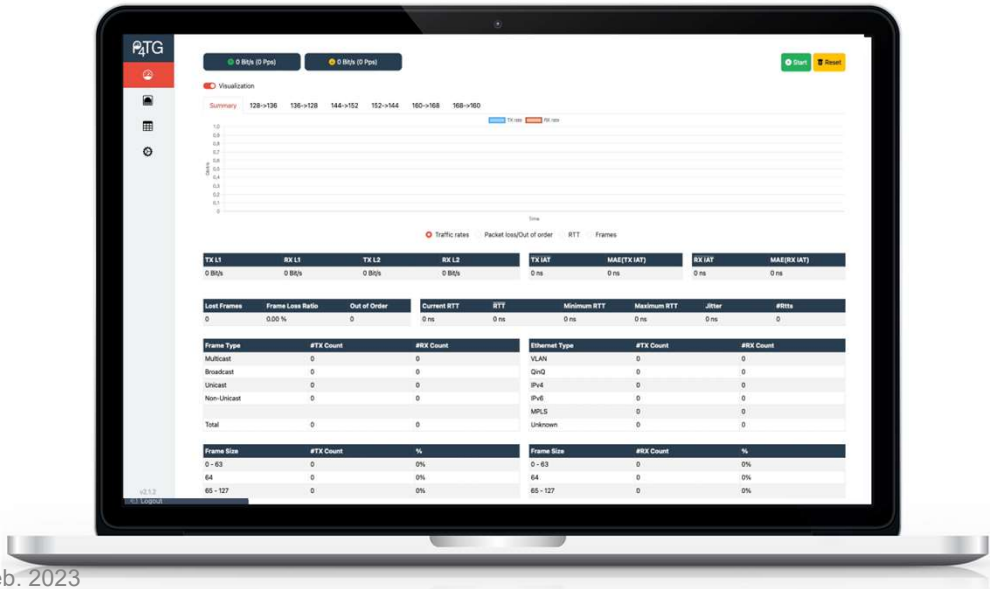
- Traffic generators (TGs) test new protocols & network equipment with realistic traffic rates
  - 100+ Gb/s difficult to generate with software
  - Hardware-based TGs are very expensive ($$$$$)

+

=

- Traffic generation & analysis with P4 and Intel Tofino™ ASIC
  - Intel Tofino™ offers built-in capabilities for traffic generation

- We implement TG configuration and measuring functions
  - Constant bit-rate & poisson traffic
    - Up to 10x 100 Gb/s traffic generation
    - Customizable Ethernet and IPv4 traffic
  - Measure several metrics directly in the data plane (P4) for highest precision
    - L1/L2 TX & RX rates
    - Packet loss, out of order
    - Round-trip-time
    - …



S. Lindner, Marco Häberle, and M. Menth: P4TG: 1 Tb/s Traffic Generation for Ethernet/IP Networks, in IEEE Access, Feb. 2023

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

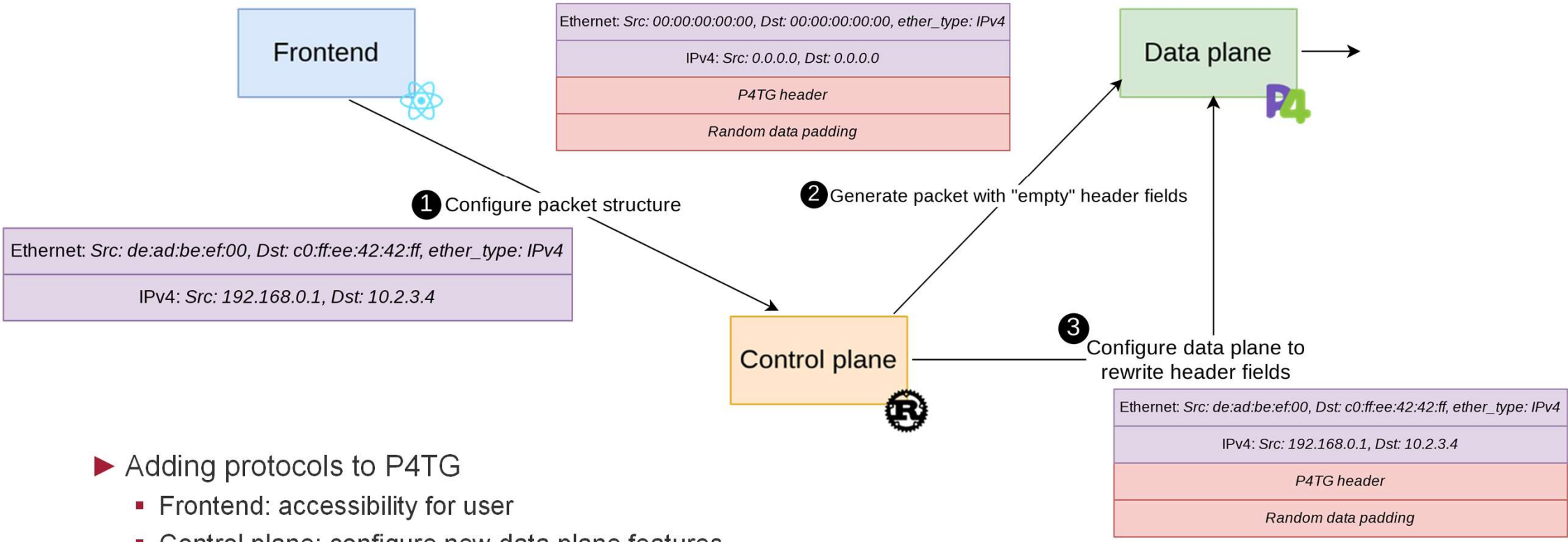
Motivation

- P4TG was published in 2023
  - Many new features have been requested to be added to P4TG
- Enhanced traffic generation capabilities
  - Ethernet and IPv4 traffic is not enough
  - → **Protocols**
- Traffic generation must keep up with increasing traffic rates
  - → **Performance**
- User experience must be improved
  - Ease of operation
  - Visualization
  - Accessibility
  - → **Automation**

S. Lindner, Marco Häberle, and M. Menth: P4TG: 1 Tb/s Traffic Generation for Ethernet/IP Networks, in IEEE Access, Feb. 2023

Enhancements to P4TG

Protocols: Traffic Generation



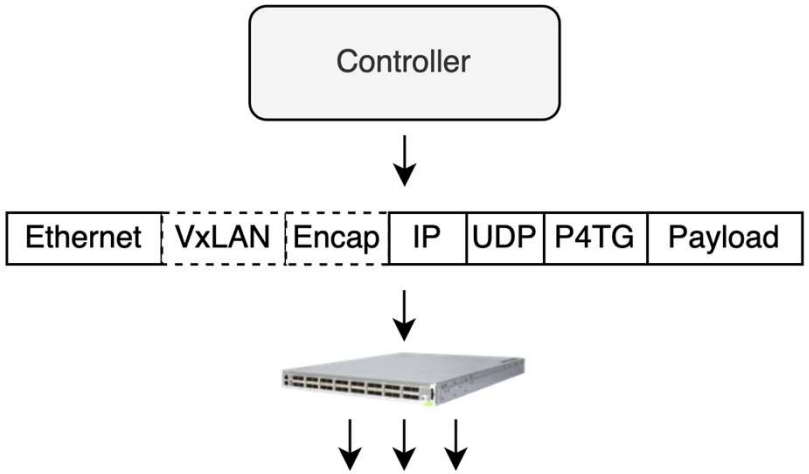
- Adding protocols to P4TG
  - Frontend: accessibility for user
  - Control plane: configure new data plane features
  - Data plane: parsing and processing of new protocol

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Protocols: Traffic Customization

- The initial P4TG version only supports Ethernet and IPv4 traffic

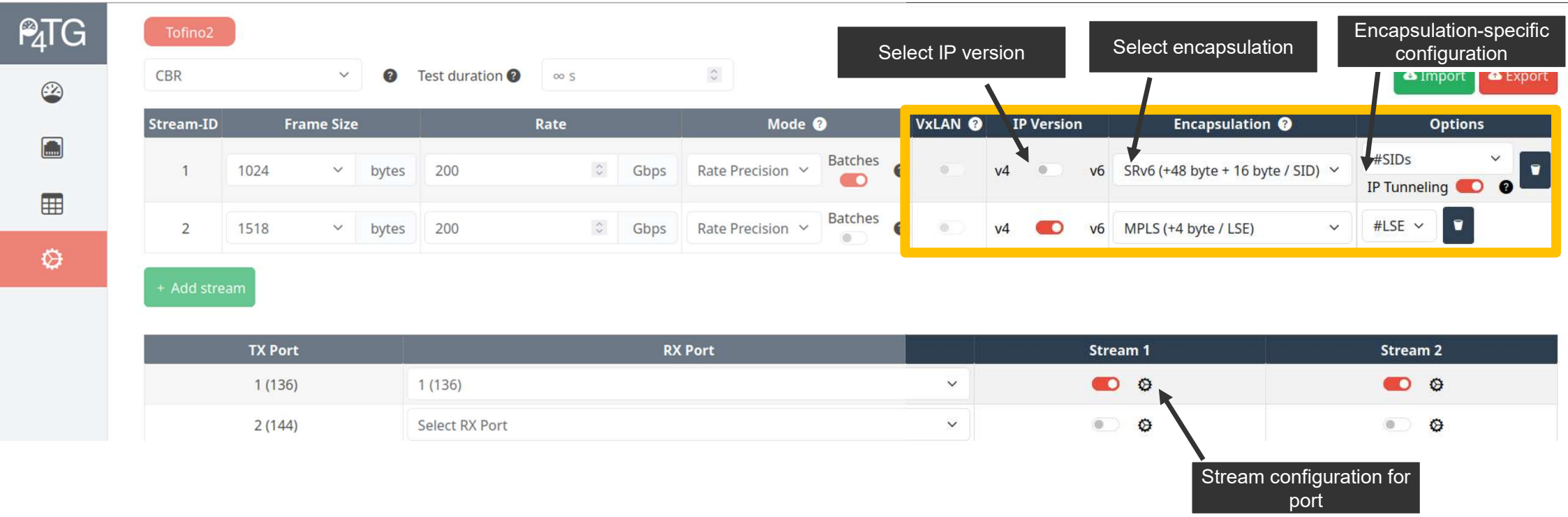
- P4TG is extended with various protocols
  - IPv6 traffic generation (48 bits of address randomization)
  - Traffic encapsulation protocols
    - VLAN
    - QinQ
    - MPLS (up to 15 LSEs)
    - SRv6 (up to 3 SIDs)
    - VxLAN



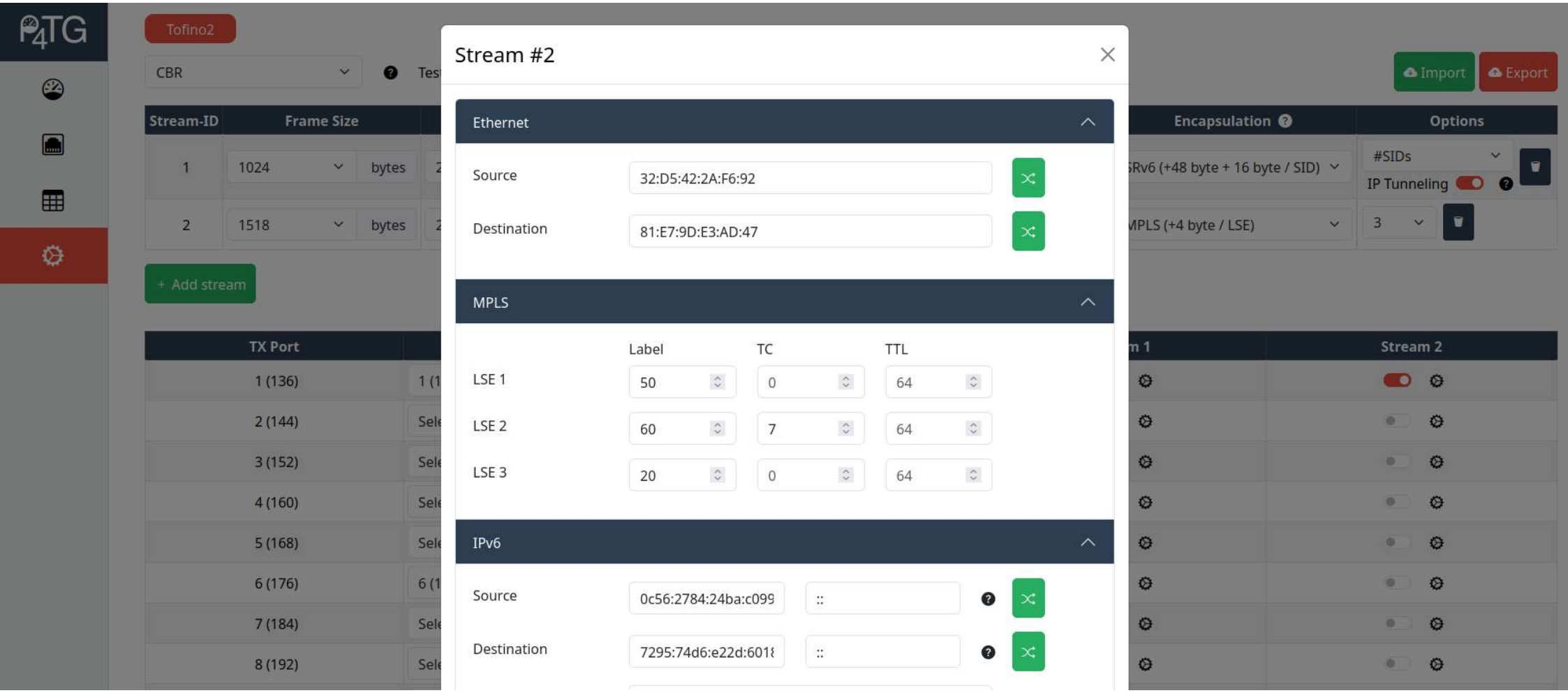
- Traffic generation is fully customizable in the web-based GUI or via the REST API

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Protocols: Traffic Customization



Protocols: Traffic Customization



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Performance: Tofino 2 Support

- P4TG was only available for the Intel Tofino 1
  - Up to 100 Gb/s per port
- Intel Tofino 2 hardware is more powerful
  - Up to 400 Gb/s per port
  - Extended pipeline size
- P4TG is ported to the Intel Tofino 2 platform
  - Up to 4 Tb/s (10 x 400 Gb/s) traffic generation
  - More sophisticated traffic encapsulation (SRv6)
- Generation accuracy
  - P4TG does not achieve 400 Gb/s per port for small frame sizes
    - 256 B frames or larger are required for that
    - Encapsulation headers increase the frame size and alleviate this effect
  - Close to 400 Gb/s achieved for larger frame sizes

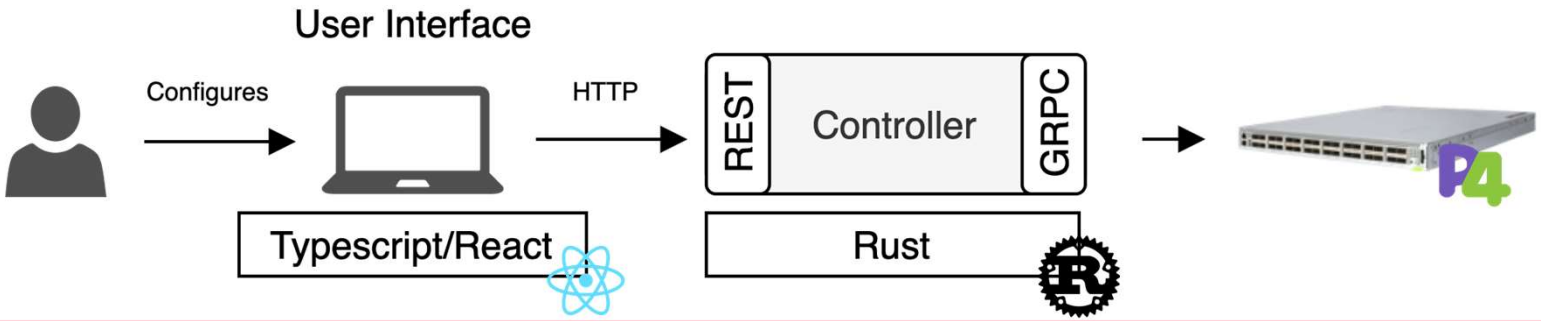
TABLE II
MAXIMUM TRAFFIC GENERATION RATE OF P4TG PER PORT ON TOFINO™1 AND 2 FOR DIFFERENT FRAME SIZES.

Frame size	Tofino 1	Tofino 2
64 B	99.37 Gb/s	294.00 Gb/s
128 B	98.56 Gb/s	388.50 Gb/s
256 B	99.77 Gb/s	396.73 Gb/s
512 B	99.55 Gb/s	399.83 Gb/s
1024 B	99.41 Gb/s	399.72 Gb/s
1518 B	99.62 Gb/s	398.47 Gb/s

Performance: Improved Control Plane

- Control plane exposes REST API to configure traffic generation
  - Leveraged by user interface to interact with control plane via HTTP (REST) calls
    - written in Typescript/React
  - Configures Tofino via BFRT-GRPC interface
  - Initially written in Python
  - Entirely redesigned in Rust
    - Byproduct: the Rust Barefoot Runtime (rbfrt) library

- Makes the control plane more robust through the strict typing system of Rust



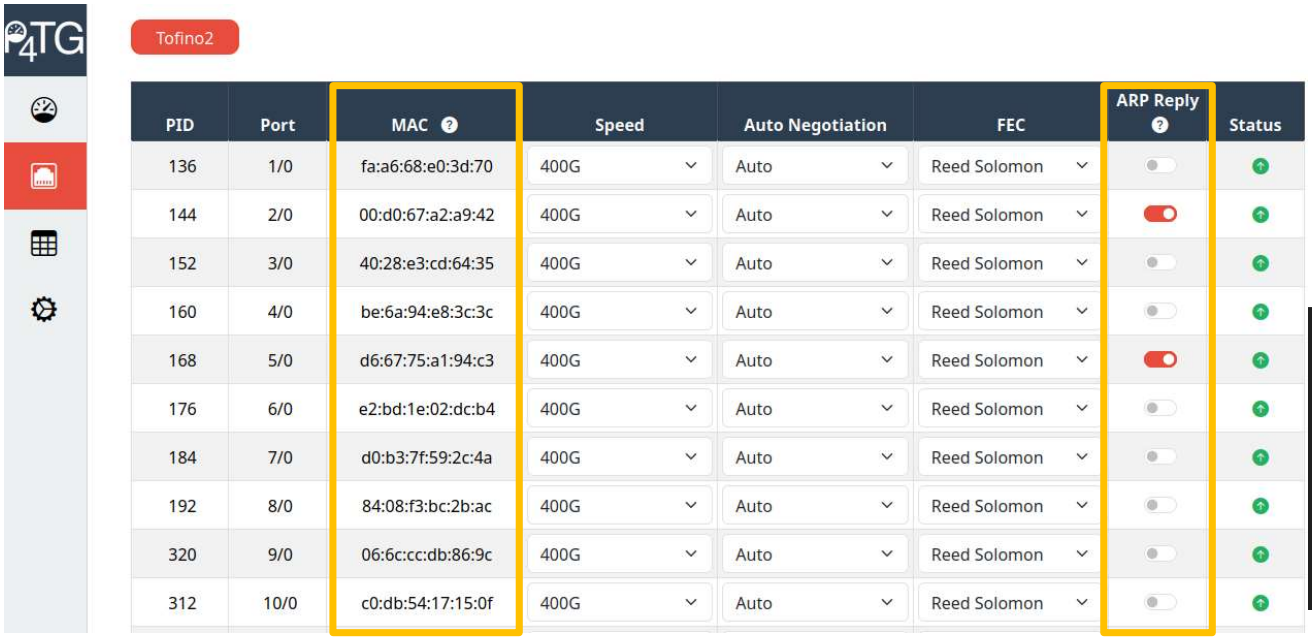
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Automation: Improved User Experience

- Live visualization of generated and measured traffic rates and metrics

- Better accessibility
  - Localization
  - Dark mode

- Report generation
  - Export measurement data in a .pdf or .csv file format

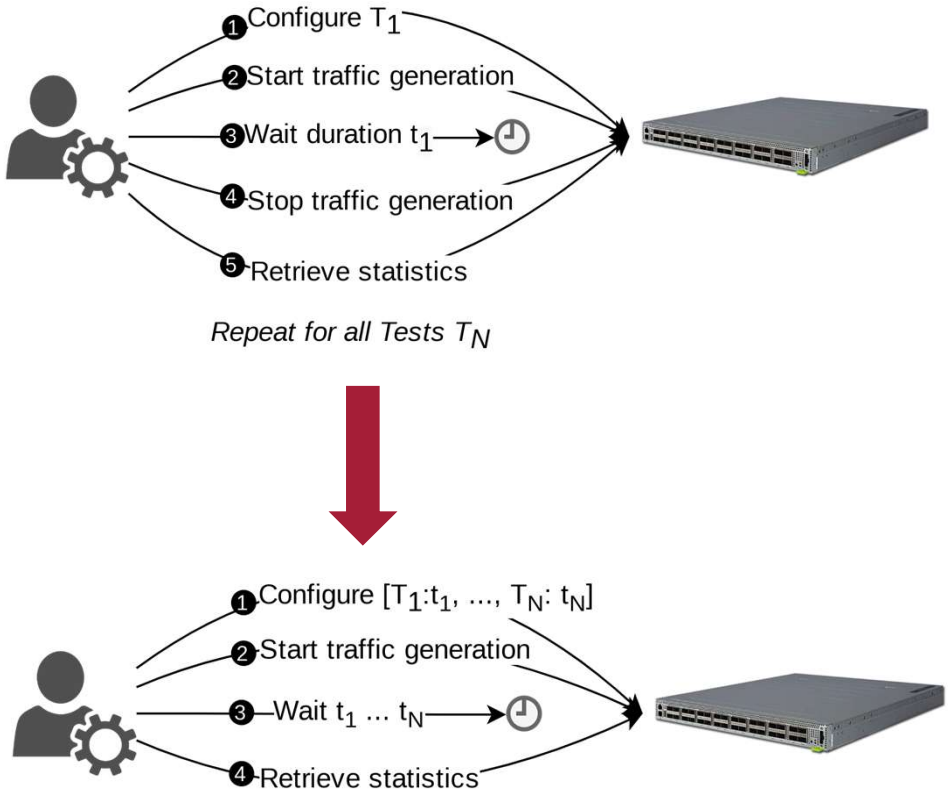


PID	Port	MAC	Speed	Auto Negotiation	FEC	ARP Reply	Status
136	1/0	fa:a6:68:e0:3d:70	400G	Auto	Reed Solomon		
144	2/0	00:d0:67:a2:a9:42	400G	Auto	Reed Solomon		
152	3/0	40:28:e3:cd:64:35	400G	Auto	Reed Solomon		
160	4/0	be:6a:94:e8:3c:3c	400G	Auto	Reed Solomon		
168	5/0	d6:67:75:a1:94:c3	400G	Auto	Reed Solomon		
176	6/0	e2:bd:1e:02:dc:b4	400G	Auto	Reed Solomon		
184	7/0	d0:b3:7f:59:2c:4a	400G	Auto	Reed Solomon		
192	8/0	84:08:f3:bc:2b:ac	400G	Auto	Reed Solomon		
320	9/0	06:6c:cc:db:86:9c	400G	Auto	Reed Solomon		
312	10/0	c0:db:54:17:15:0f	400G	Auto	Reed Solomon		

- Automated ARP replies
  - ARP table of a DUT must be filled by ARP replies and requests
  - → P4TG now responds to ARP requests with pre-configurd MAC addresses to facilitate automated testing

Automation: Testing

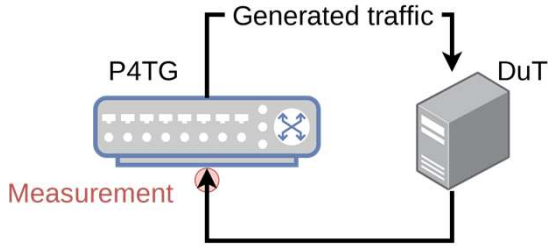
- The REST API provides endpoints to start, stop, and configure traffic generation
  - Multiple individual network tests in sequence must be scripted using those endpoints
- P4TG is extended to accept a list of pre-defined tests
  - Stream description
  - Duration
- P4TG executes list of tests with a configured duration
  - → Developer can pre-define multiple tests which are applied sequentially



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

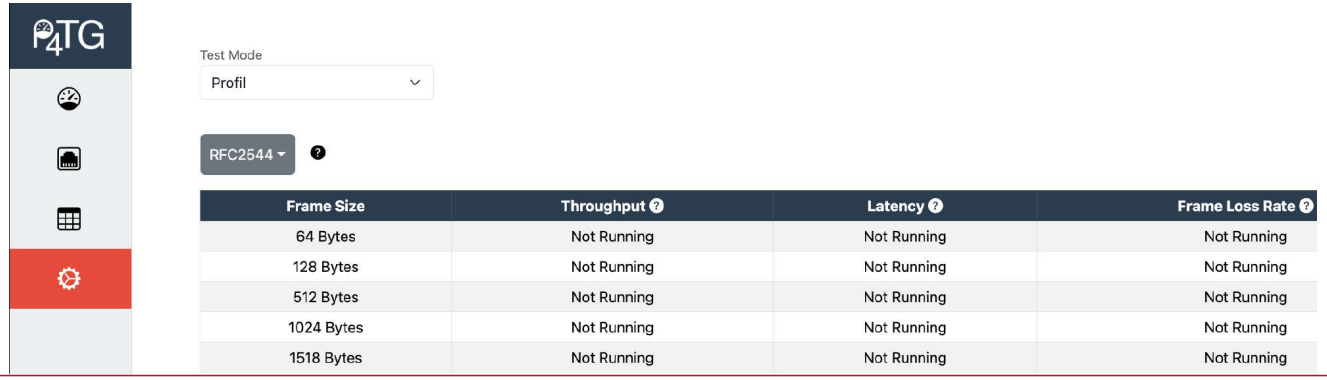
Automation: Test Profiles

- RFC 2544: Benchmarking Methodology for Network Interconnect Devices
  - Defines tests to describe the performance characteristics of a device under test
    - *Maximum throughput* test
    - *Latency* test
    - *Frame loss rate* test
    - *Reset time* test



- Each benchmark consists of multiple tests with multiple frame sizes
  - → Those tests must be configured manually ☹

- P4TG is extended with a profile mode that applies pre-defined profiles such as RFC 2544 benchmarking



Frame Size	Throughput	Latency	Frame Loss Rate
64 Bytes	Not Running	Not Running	Not Running
128 Bytes	Not Running	Not Running	Not Running
512 Bytes	Not Running	Not Running	Not Running
1024 Bytes	Not Running	Not Running	Not Running
1518 Bytes	Not Running	Not Running	Not Running

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Conclusion

- P4TG is extended with various protocols
  - IPv6, VLAN, QinQ, VxLAN, SRv6, MPLS
- P4TG is ported to the Tofino 2 platform
  - Up to 10x 400 Gb/s of traffic generation
- Enhanced user experience
  - Localization, dark mode, report generation, automated network testing, ARP replies, test profiles
- Future work
  - Neighbor Discovery Protocol (NDP), NETCONF API
- Open for contribution
  - Missing feature/protocol? → Welcome to open an issue / PR
  - https://github.com/uni-tue-kn/P4TG

Any Questions?

Enhancements to P4TG: Protocols, Performance, and Automation

https://github.com/uni-tue-kn/P4TG

Fabian Ihle
University of Tübingen
fabian.ihle@uni-tuebingen.de