ProD3 Overview for Linux Foundation
March 21st, 2022

The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA) under contract HR0011-20-C-0160.

Peraton LABS

USC Viterbi School of Engineering

PRINCETON UNIVERSITY

ProD3: Programmable, Distributed Defense in Depth for 5G Services

Steven Beitzel, Latha Kant, Stuart Wagner

Approved for Public Release – Distribution Unlimited

Team Key Personnel

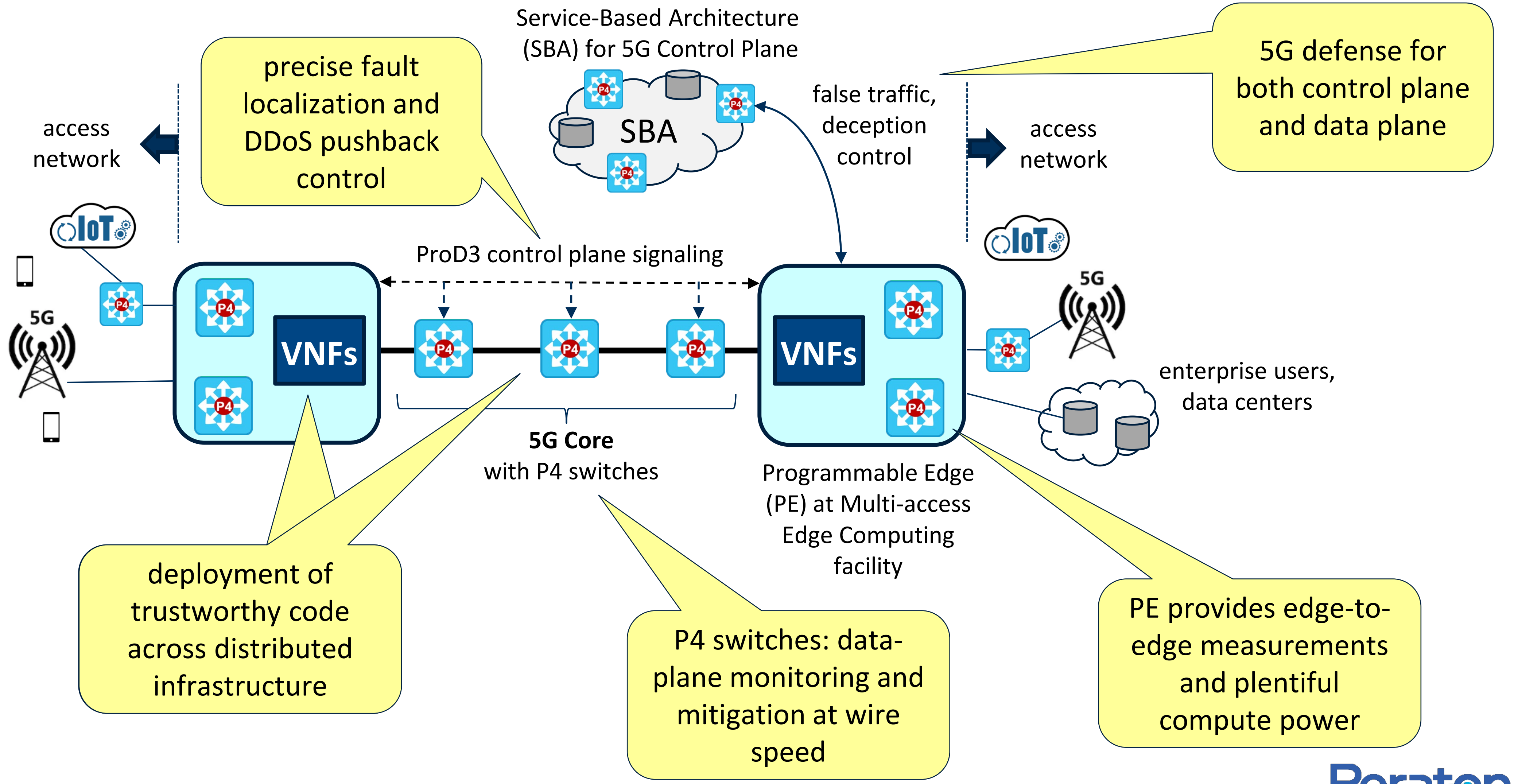
- Steve Beitzel (co-PI)
- Stuart Wagner (co-PI)
- Latha Kant
- Jennifer Rexford (Princeton)
- Andrew Appel (Princeton)
- Salman Avestimehr (USC)
- Bhaskar Krishnamachari (USC)
- Many other SMEs at Peraton Labs, Princeton and USC

Presentation Outline

- ProD3 Project Overview
- DDoS Defense status and recent results
  - Final Phase 1 Mirai DDoS simulations
  - VoIP demo in hardware testbed (video)
- Network Compromise Defense
  - Overview
  - Methodology
- Future Plans
  - Transition activity
  - Phase 2 software development and experimentation

ProD3 Architecture

ProD3 provides distributed programmability for service assurance in 5G



precise fault localization and DDoS pushback control

5G defense for both control plane and data plane

deployment of trustworthy code across distributed infrastructure

P4 switches: data-plane monitoring and mitigation at wire speed

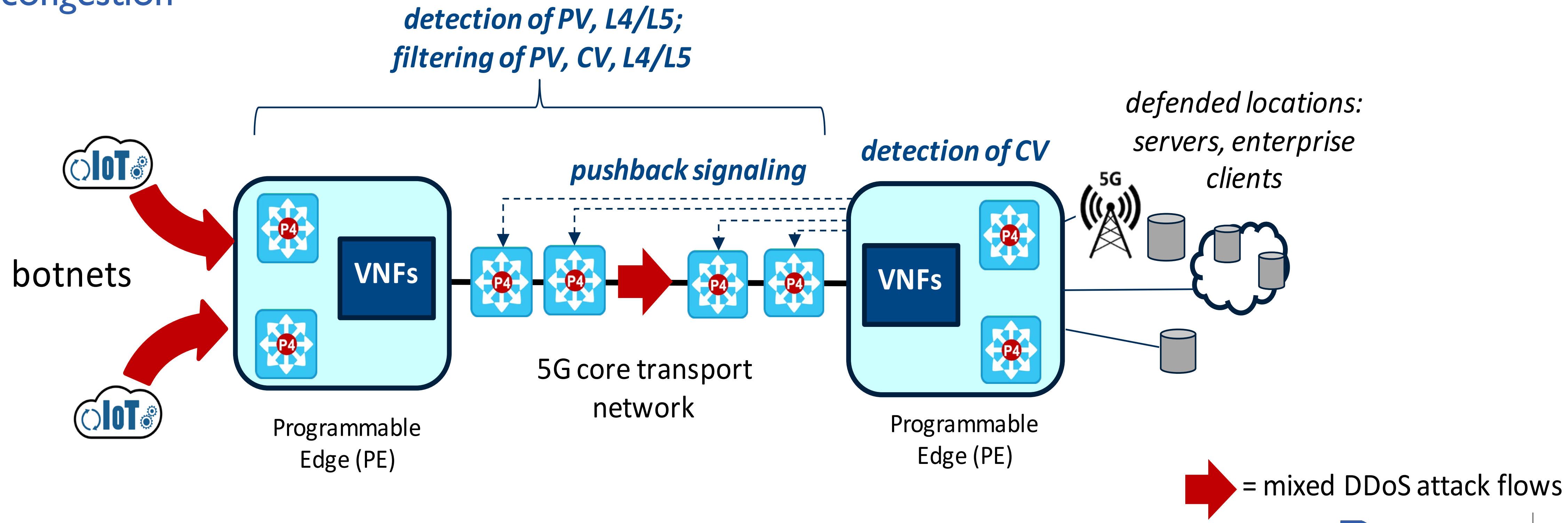
PE provides edge-to-edge measurements and plentiful compute power

Approved for Public Release – Distribution Unlimited

DDoS Defense Technical Approach

ProD3 will detect and defeat four classes of DDoS attacks:

- **Continuous volumetric** (CV), **Layer 4** (TCP state exhaustion), and **Layer 5** (application/protocol logic) attacks modeled after Mirai
- **Pulsed volumetric** (PV) attacks that leverage high peak bandwidths of 5G access to induce congestion



detection of PV, L4/L5; filtering of PV, CV, L4/L5

pushback signaling

detection of CV

defended locations: servers, enterprise clients

botnets

Programmable Edge (PE)

5G core transport network

Programmable Edge (PE)

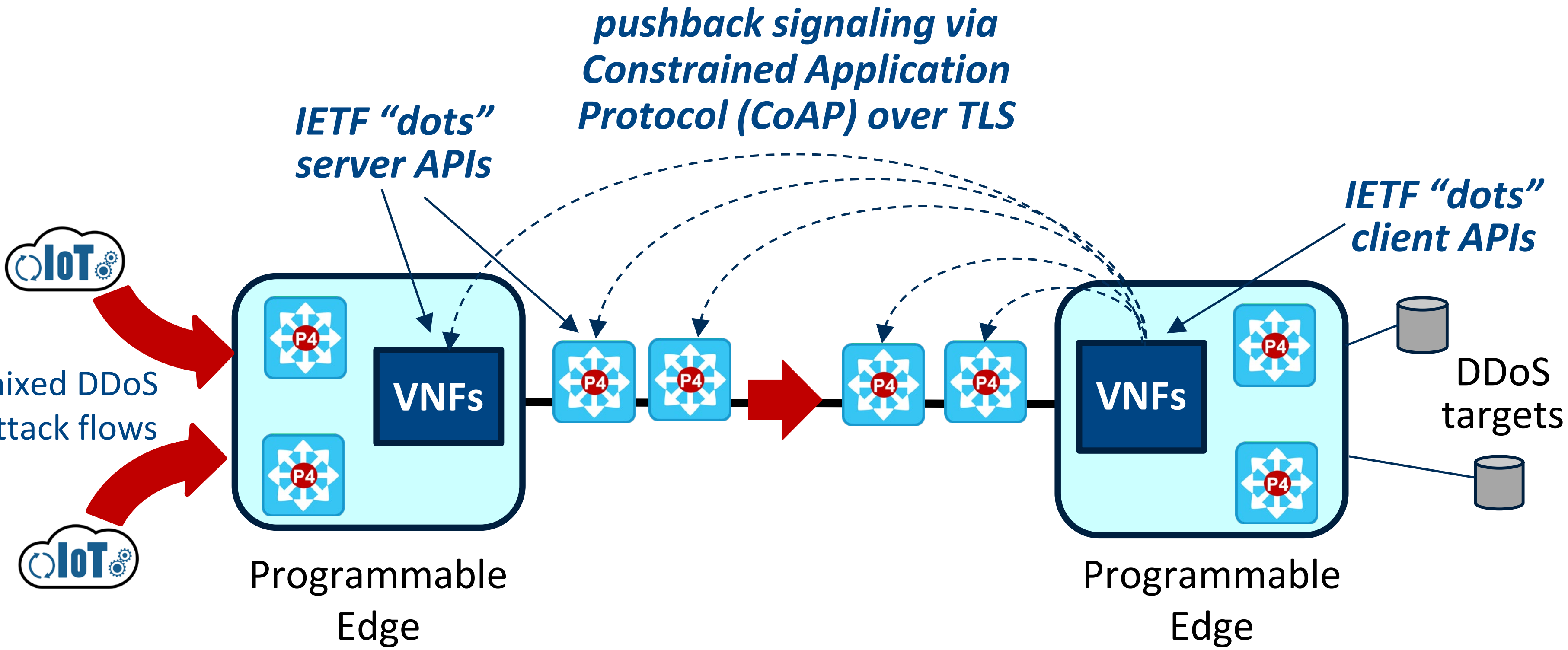
= mixed DDoS attack flows

Challenges in Large-Scale DDoS Defense

Challenge	ProD3 Solution
Ensure secure, **real-time** response to DDoS attacks **at scale**	Distributed, programmable detection and filtering on P4 switches; new pushback signaling protocol design provides scale and guards against tampering
Defeat **continuous volumetric attacks** that use randomized transport addresses	Programmable edges leverage ICMP responses from attacked hosts to identify and block offending flows, while preventing ICMP amplification attacks
Defeat **low-volume L4/L5 attacks** without access to hosts under attack	Migrate host-based algorithms to programmable edges via offline machine learning
Defeat **pulsed volumetric attacks** that current queue-monitoring techniques cannot detect	Data-plane monitoring of queue occupancies on P4 switches reveals fine-grained congestion events and the malicious flows responsible for them

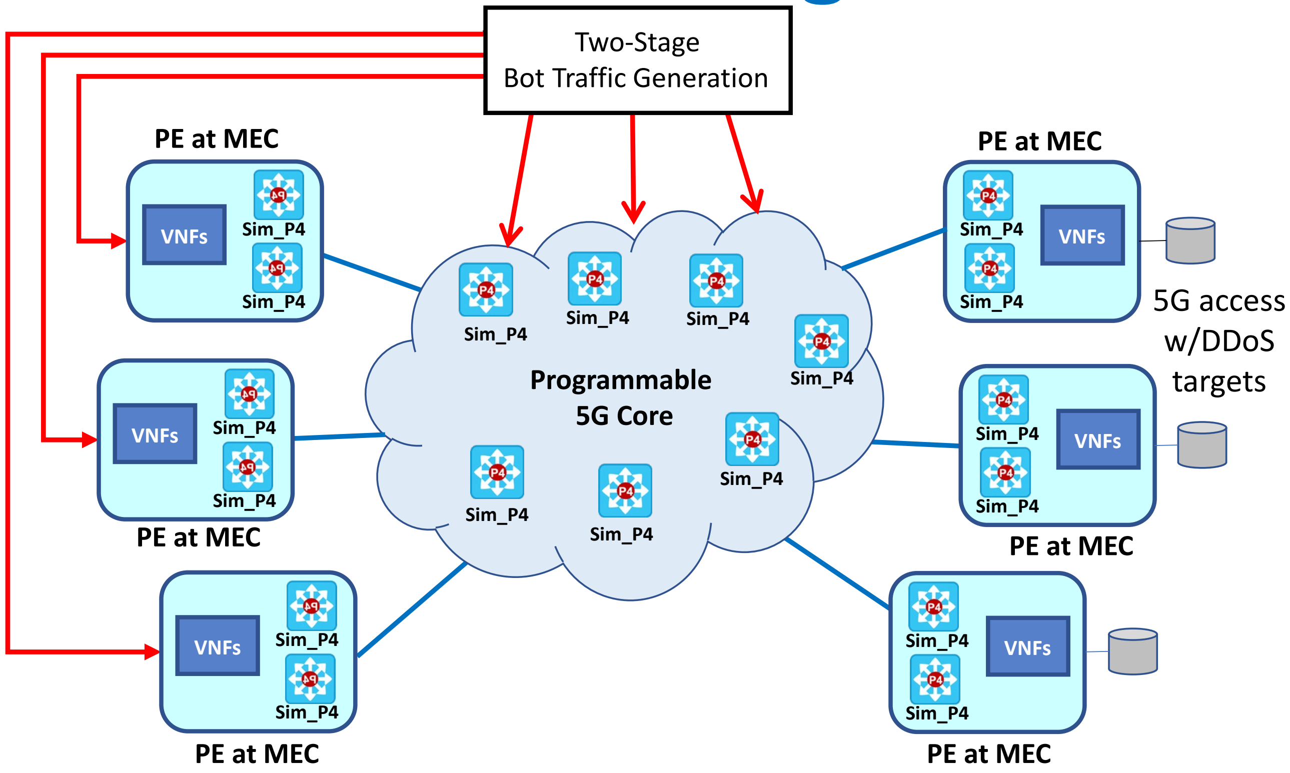
Approved for Public Release – Distribution Unlimited

ProD3 Control Plane – Tying It All Together



- Goal: scalable and secure operation for DDoS pushback, network monitoring, and real-time responses to disruptions across distributed programmable components
- Design will draw from several related efforts in the IETF (e.g., "DDoS Open Threat Signaling")
- Logical star signaling topology differs from prior hop-by-hop pushback designs and employs secure hashing and TLS to prevent forgeries

DDoS Defense Testbed and Testing: Simulation Environment



MEC = multi-access edge computing

- Testbed utilizes custom simulator developed specifically to address OPS-5G scale
- Botnet modeling is source-network agnostic and generates a configurable mixture of DDoS attacks
- Virtualized DDoS attack targets are instrumented to monitor attack impacts and the effectiveness of mitigations

DDoS Defense Solution Development Status

Mirai Attack Type	Simulation w/ 10^9 bots	Software on HW testbed
Volumetric (UDP) Flood	✓	✓
TCP ACK Flood	✓	
TCP SYN Flood	✓	✓
TCP STOMP Flood	✓	
HTTP Flood	✓	

- Successfully repelling volumetric and protocol-based DDoS attacks. Subsequent slides focus on recent HTTP Flood results
- Complete botnet defeat in <1 sec for UDP, TCP ACK, TCP SYN attacks
- STOMP, HTTP Flood: total attack *requires several minutes of network time to play out*; individual bots defeated in <<1 sec once their malicious traffic reaches MECs
- Software for UDP Flood defense employed for Phase 1 Milestone (VoIP) demo

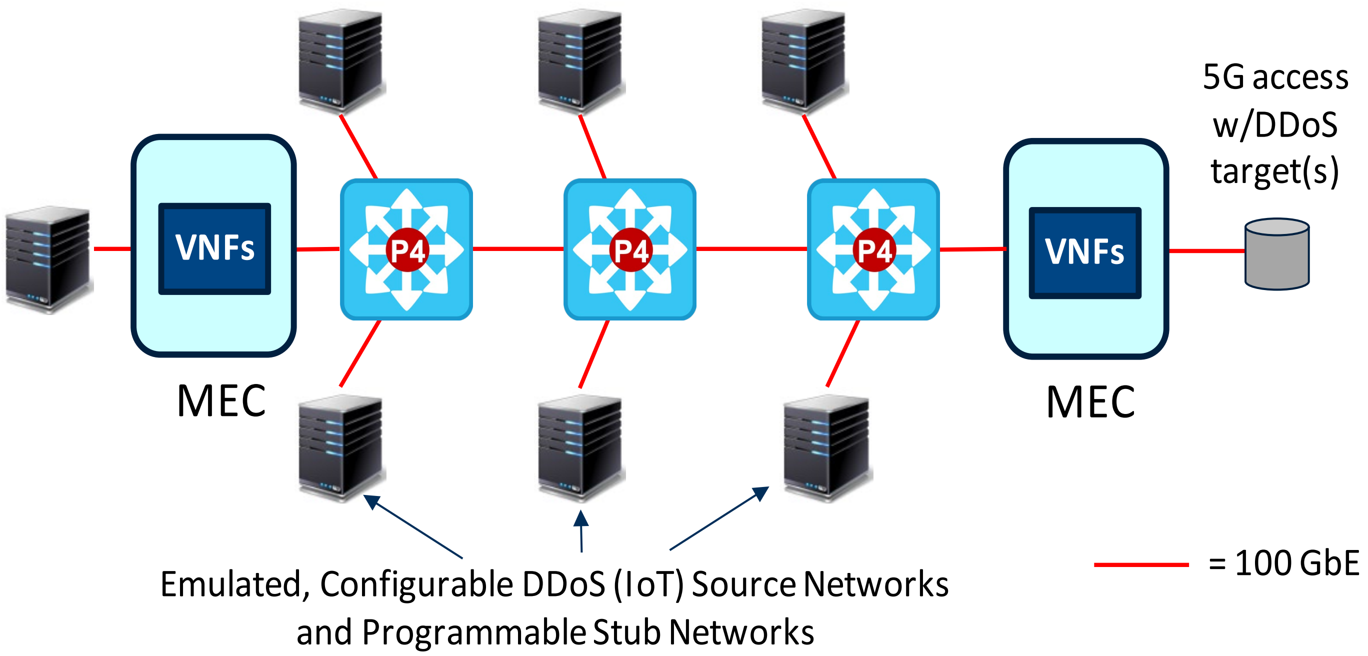
Approved for Public Release – Distribution Unlimited

Recent DDoS Simulation Results

	Simulated ProD3 Mitigation Time for 10^9 Bots	
Mirai Attack Type	**Attack Target on 10 Gb/s Access Network**	**Attack Target on 50 Gb/s Access Network**
UDP Flood	< 100 msec	< 100 msec
SYN Flood	< 100 msec	< 100 msec
HTTP Flood	185 sec	33 sec

- These results cover the three attack types and test cases requested by the IT&E Team
- All simulations were performed with 10^9 Mirai bots executing the attacks
- Mitigation was measured using VoIP as the test application, and with R-Score and MOS Score being the application metrics, as requested by the IT&E Team (TCP app was also used)
- The lengthy HTTP Flood mitigation time for the lower access bandwidth is due to bots' inability to fight through their own congestion to reach the defended target
  - This limitation also occurs for the TCP STOMP attack

DDoS Defense Testbed and Testing: Experiments in Hardware


5G access w/DDoS target(s)
VNFs
P4
P4
P4
VNFs
MEC
MEC
Emulated, Configurable DDoS (IoT) Source Networks and Programmable Stub Networks
= 100 GbE

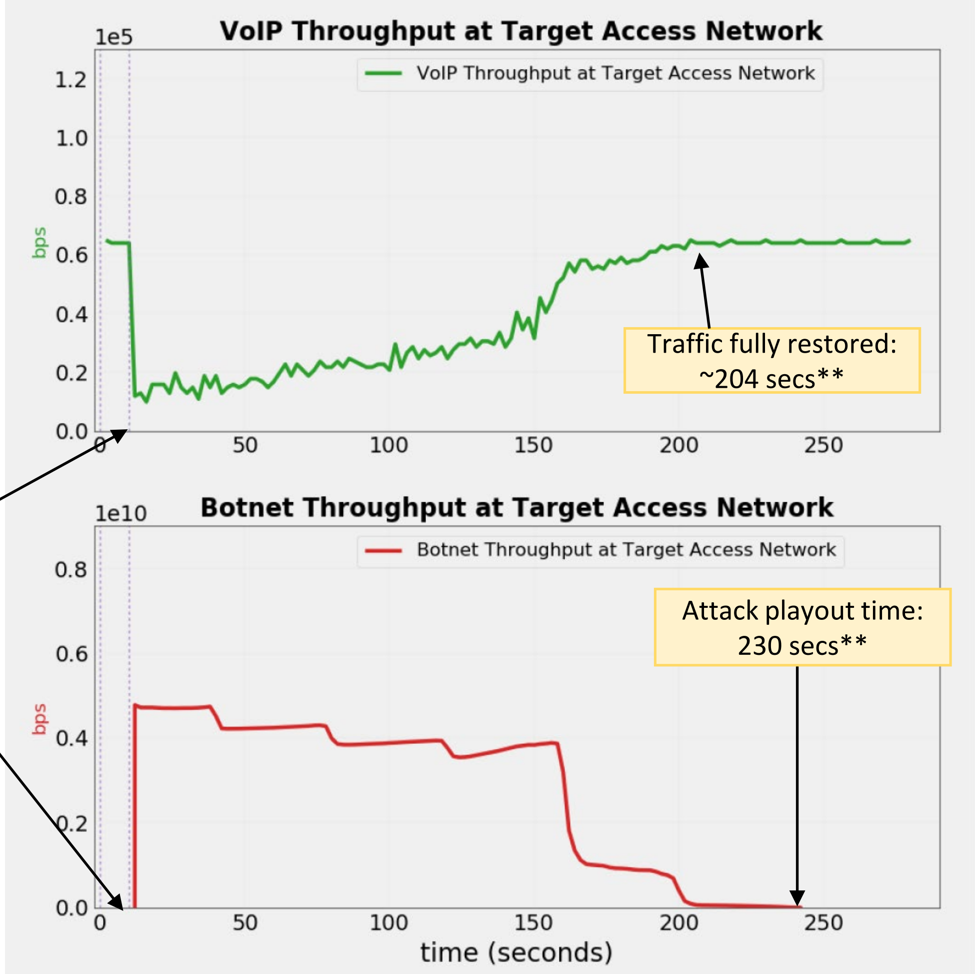
- We use a network of hardware P4 switches to validate simulation results and to substantiate other aspects of performance that simulation may not effectively capture at high throughputs
- Testing includes saturation of 100 GbE interfaces with botnet traffic to quantify performance under worst-case attack conditions
- Hardware testbed supports experiments with other (non-DDoS) ProD3 defenses

Approved for Public Release – Distribution Unlimited

Simulation Data: HTTP Flood (w/ VoIP traffic)

Core & Bot Access Network Bandwidths: 100Gb/s; Target Access Network Bandwidth: 10Gb/s

Attack begins (@ 10sec)

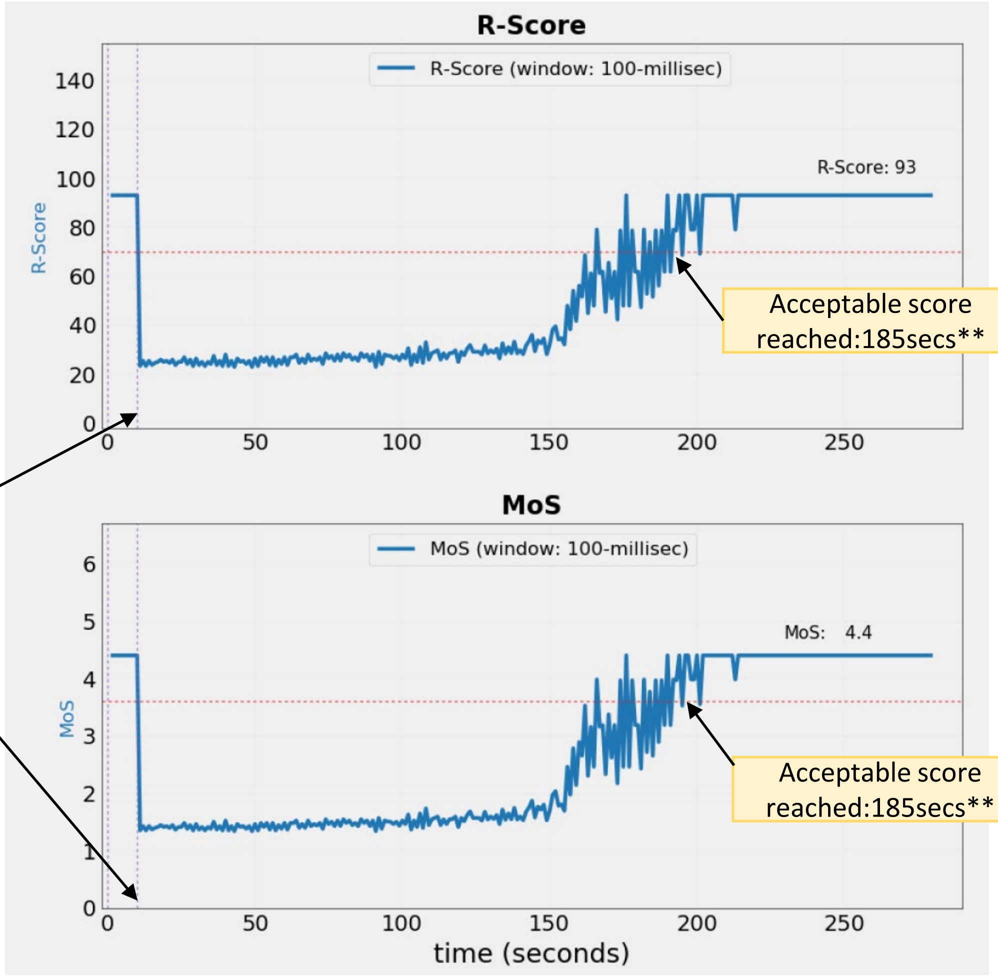

VoIP Throughput at Target Access Network

Traffic fully restored: ~204 secs**

Botnet Throughput at Target Access Network

Attack playout time: 230 secs**

Attack begins (@ 10sec)


R-Score

Acceptable score reached:185secs**

MoS

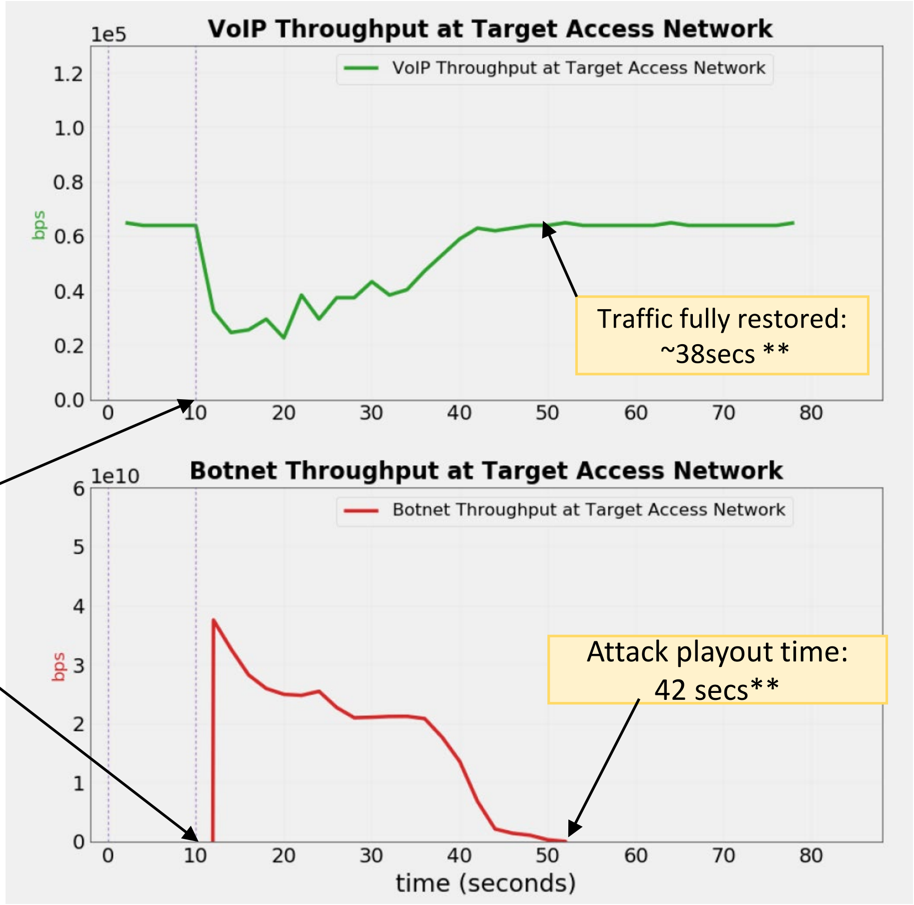
Acceptable score reached:185secs**

** Takes into account the 10sec offset (attack begin time)

Simulation Data: HTTP Flood (w/ VoIP traffic)

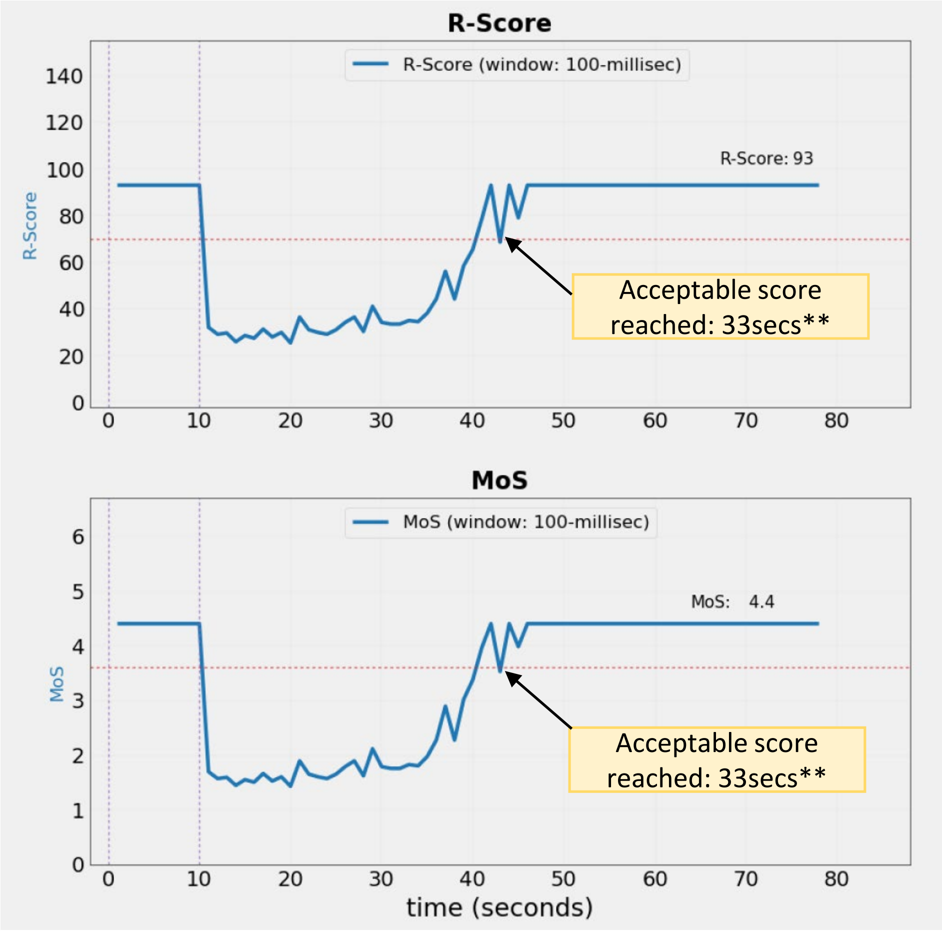
Core & Bot Access Network Bandwidths: 300Gb/s; Target Access Network Bandwidth: 50Gb/s

Attack begins (@ 10sec)



Traffic fully restored: ~38secs **

Attack playout time: 42 secs**



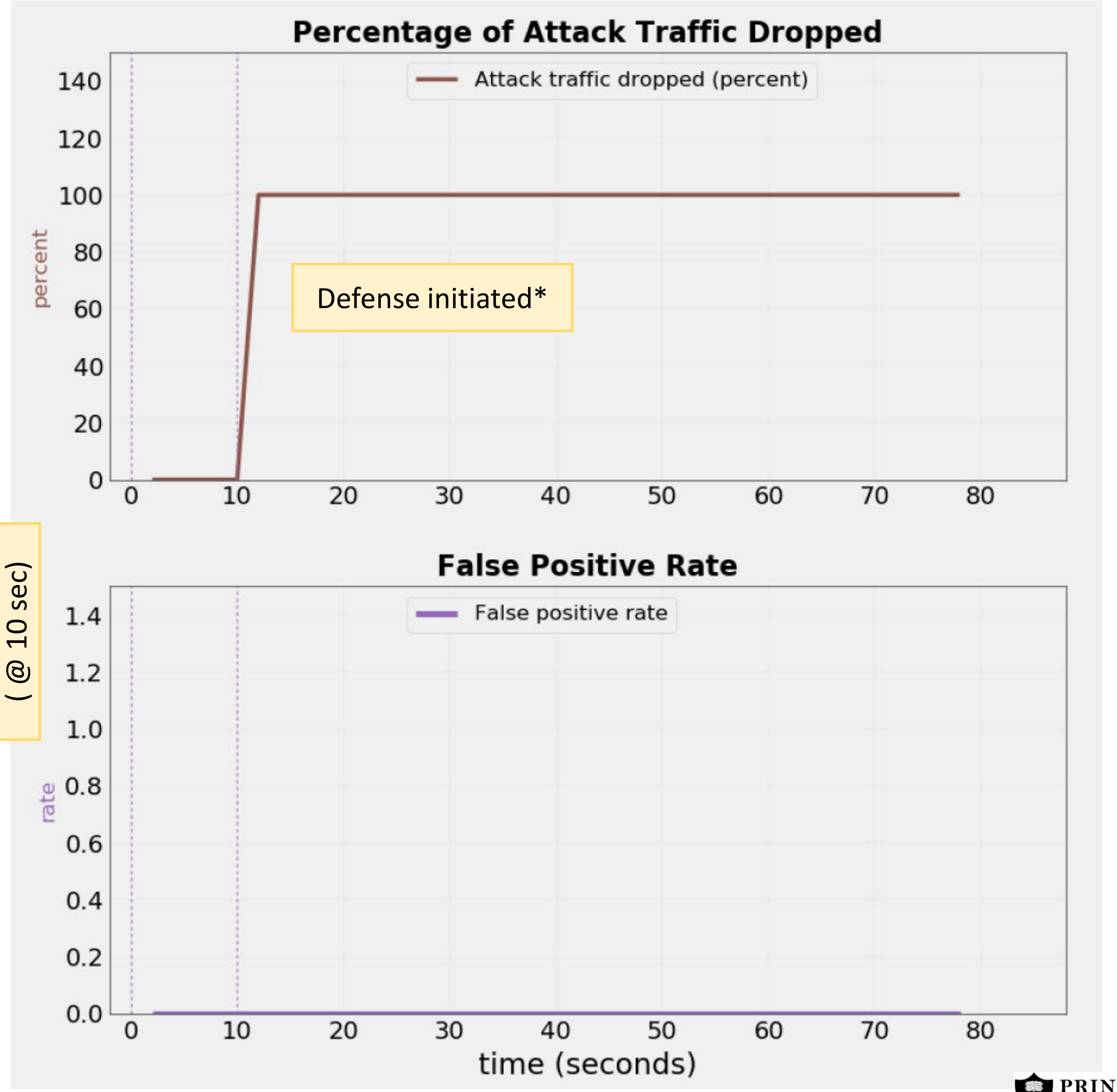
Acceptable score reached: 33secs**

Acceptable score reached: 33secs**

** Takes into account the 10sec offset (attack begin time)

HTTP Flood (w/VoIP traffic): Other Metrics Specified by IT&E Team

Core & Bot Access Network Bandwidths: 300Gbps; Target Access Network Bandwidth: 50Gbps



Defense initiated*

Attack begins (@ 10 sec)

Zero FP rate is an inherent property of how we perform the bot detection and subsequent blocking

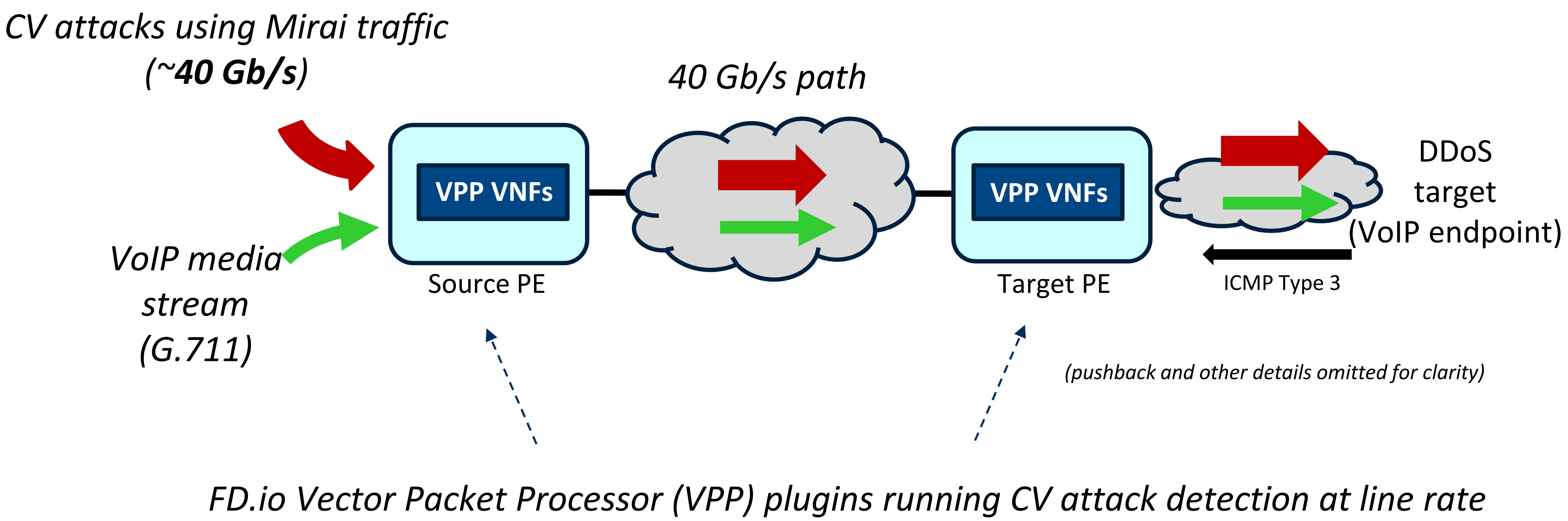
Simulation Run Times for 10^9 bots

Attack Type	Simulation Run Time (includes time for collecting all the metrics)	
	VoIP User App	TCP User App
UDP Flood*	27 mins	31 mins
TCP SYN Flood*	1 hr 42 min	2 hr 5 min
HTTP Flood*	18.8 hrs	34.9 hrs
HTTP Flood**	21.1 hrs	87.1 hrs

*** Using bandwidth configuration 1:**
Core & bot access network bandwidths: 100Gbps; target access network bandwidth: 10Gbps
**** Using bandwidth Configuration 2:**
Core & bot access network bandwidths: 300Gbps; target access network bandwidth: 50Gbps

Through code refinement, we have reduced simulation runtimes for the HTTP Flood by more than 10x compared with earlier versions, for multiple access bandwidths

VoIP Demonstration in Hardware Testbed


CV attacks using Mirai traffic
(~**40 Gb/s**)
40 Gb/s path
VPP VNFs
VPP VNFs
DDoS
target
(VoIP endpoint)
VoIP media
stream
(G.711)
Source PE
Target PE
ICMP Type 3
(pushback and other details omitted for clarity)
FD.io Vector Packet Processor (VPP) plugins running CV attack detection at line rate

- ‘sipp’ test program generates VoIP media stream using G.711 codec
- Mirai bots generate application traffic with random spoofed source IPs and destination ports
- R-score for VoIP call measured on the receiver
- Demonstration includes replay of captured audio as heard by the victim during the test
- Attack effects are completely mitigated within ~ one second

VoIP Demo in Hardware Testbed

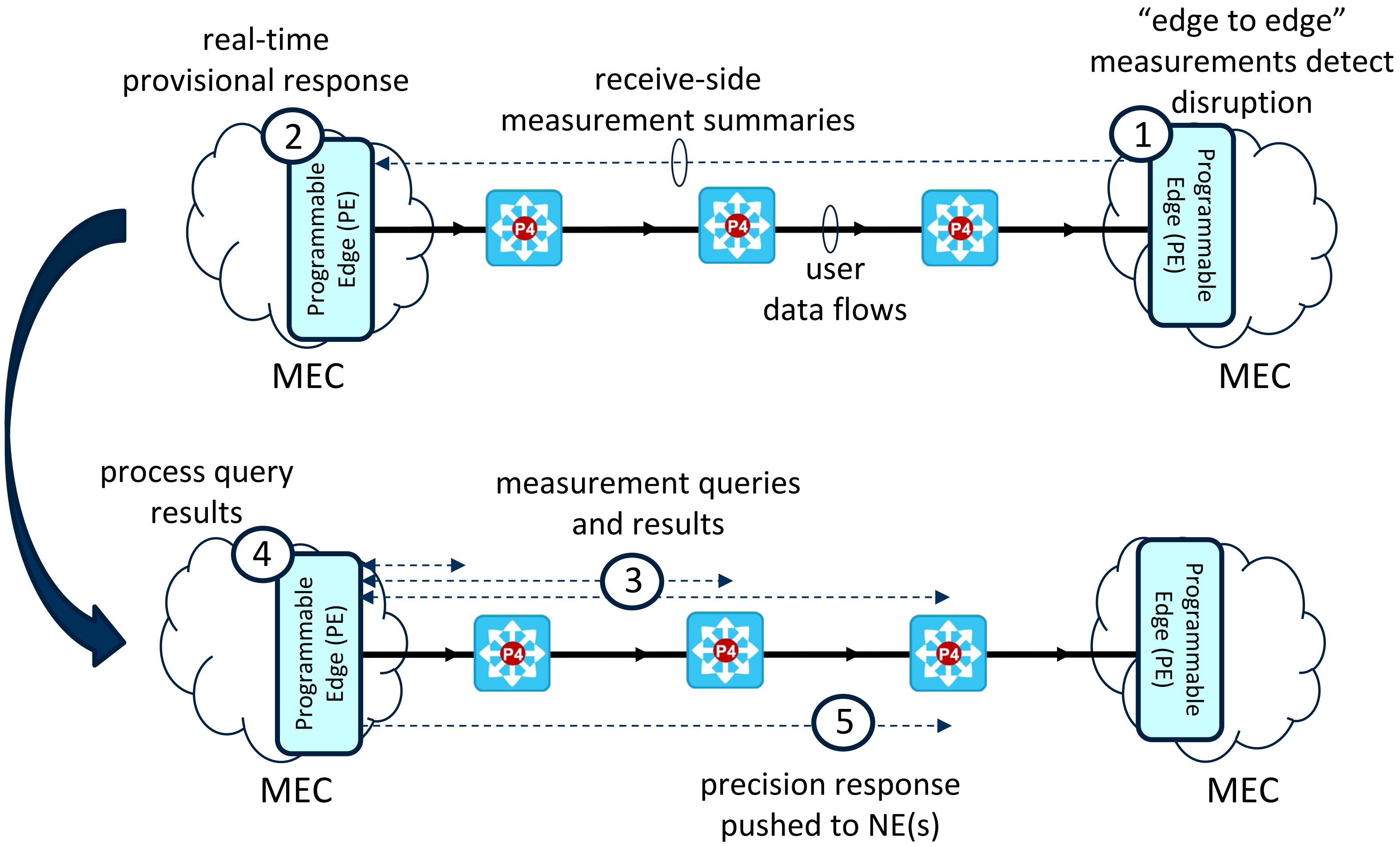
(video)

Network Compromise Defense Update

- This portion of ProD3 uses distributed, programmable network elements (MECs, switches) to detect, isolate, and (where possible) mitigate the following disruptions:
  - Packet dropping
  - Packet corruption
  - Packet reordering

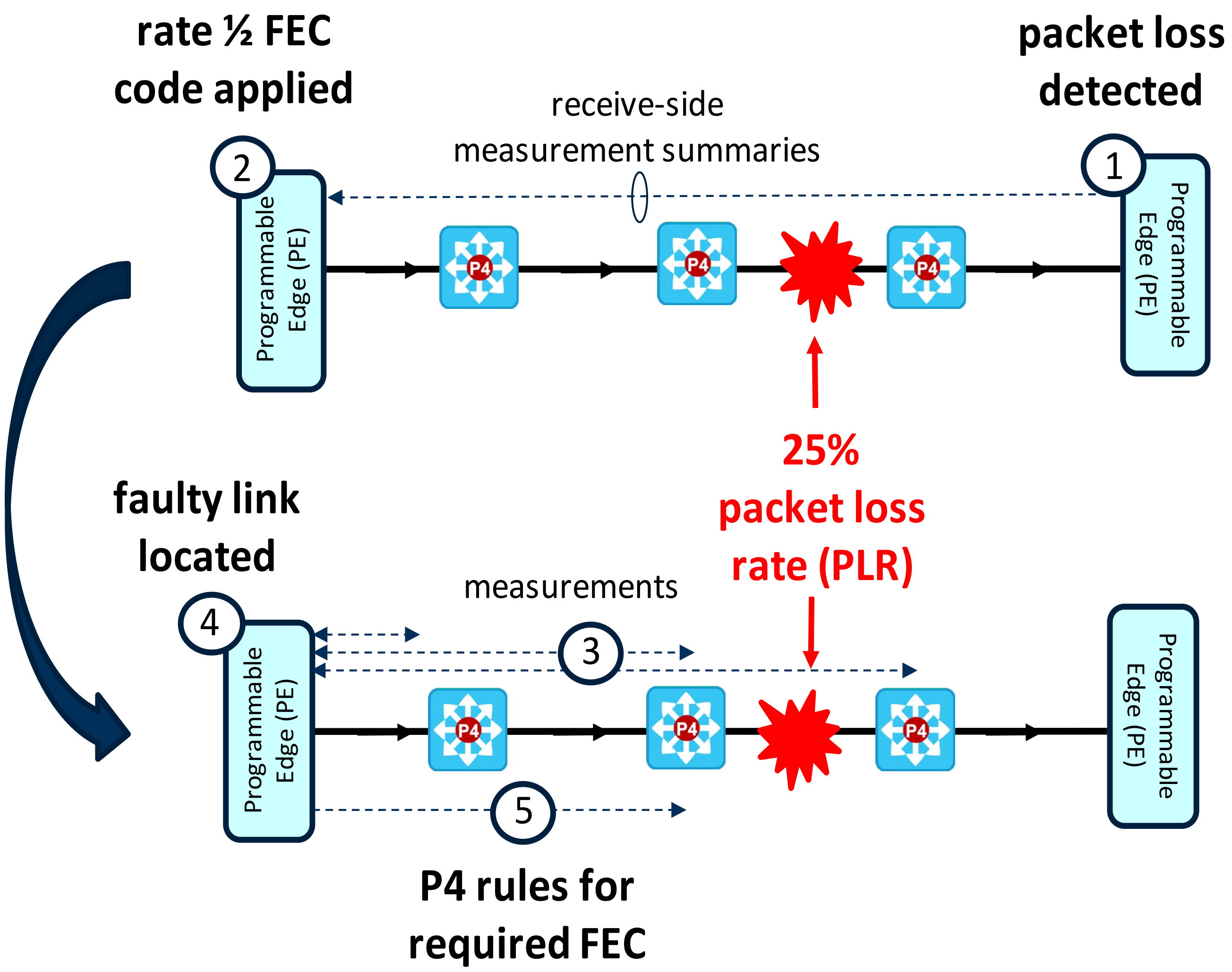
ProD3 Technical Approach

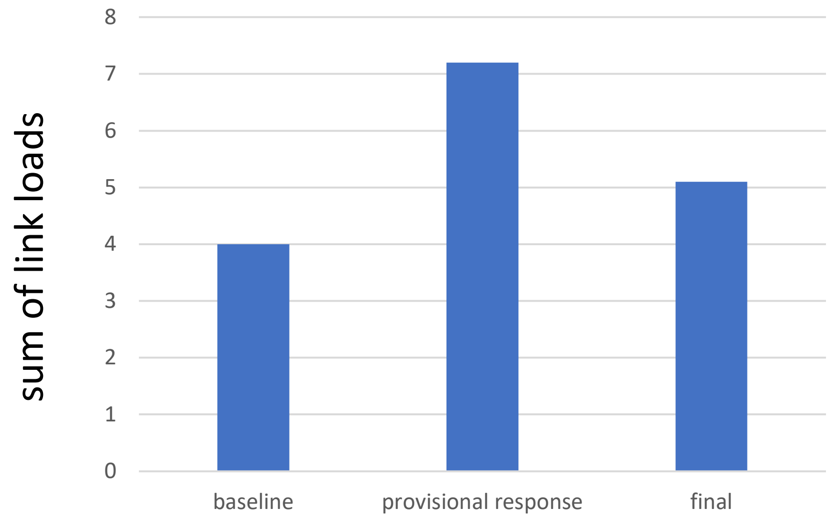
Edge and core programmable elements work in tandem to detect, locate, and mitigate disruption


real-time provisional response
receive-side measurement summaries
"edge to edge" measurements detect disruption
2
1
Programmable Edge (PE)
P4
user data flows
MEC
MEC
process query results
measurement queries and results
4
3
5
precision response pushed to NE(s)
MEC
MEC

progressive refinement of diagnosis and response

Experiment: Real-Time Mitigation of a Packet-Dropping Attack



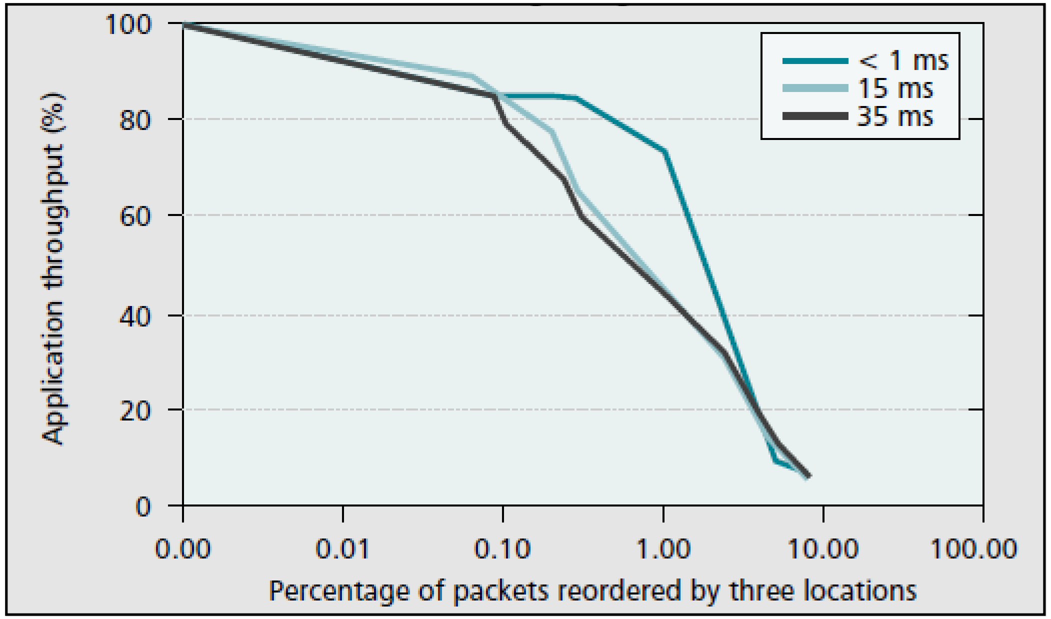


- **FEC overhead minimized** compared with provisional edge-to-edge mitigation
- The network operator now knows **where to focus its repair efforts** and exactly which flow(s) were targeted ("**service mapping**")
- To date, we have implemented steps 1-3 of this capability and are working with Princeton to continue development.

Approved for Public Release – Distribution Unlimited

Why We Care About Packet Reordering

- Significant reordering of packets in the network can severely impair many applications (both TCP-based and UDP-based)

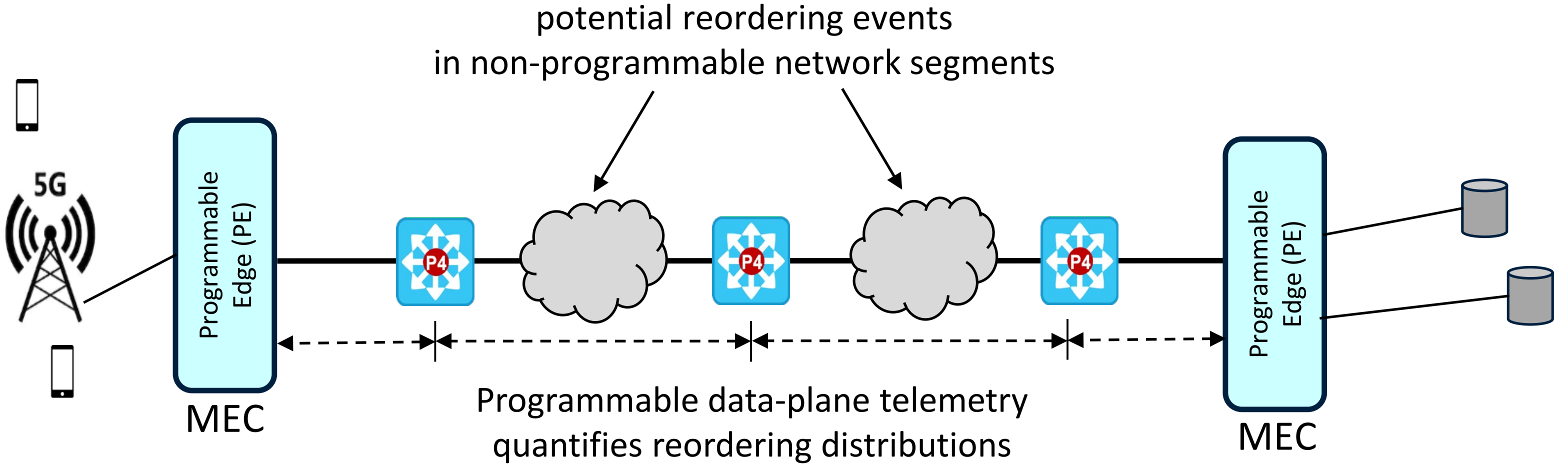


From Laor and Gendel, *IEEE Network*, 9/2002

Impact on a TCP flow

- Reordering poses a potentially critical issue for 5G and beyond, due to greater expectations for service reliability and faster network speeds
- The possibility and ease of malicious reordering are not well-recognized

Elements of ProD3 Reordering Diagnostics



potential reordering events
in non-programmable network segments
5G
Programmable Edge (PE)
P4
P4
P4
Programmable Edge (PE)
MEC
MEC
Programmable data-plane telemetry
quantifies reordering distributions

Goal: Diagnostics to isolate and characterize reordering events in near real time via:

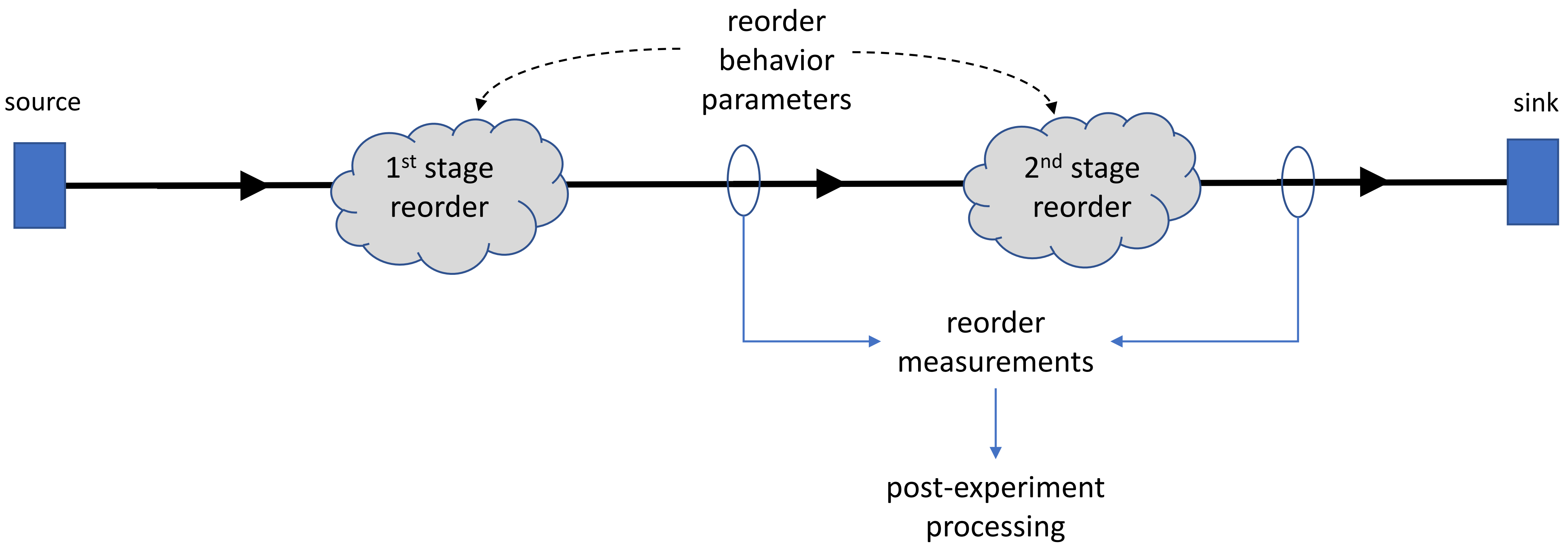
- *Programmable telemetry in the data plane* (leverages prior Princeton research)
- *IETF RFC-based conventions for reordering measurement,* to facilitate transition
- *Convolution and z-transform analysis* of telemetry measurements (at Programmable Edges) to characterize reordering within each segment

Efficient mitigation becomes feasible once these measurements are available

Simulations of Reordering

Key questions:

- To what extent is the theory accurate? (Are the reorder effects additive?)
- How does the accuracy depend on the reordering behavior?
- How can we best leverage the techniques for diagnostic purposes?

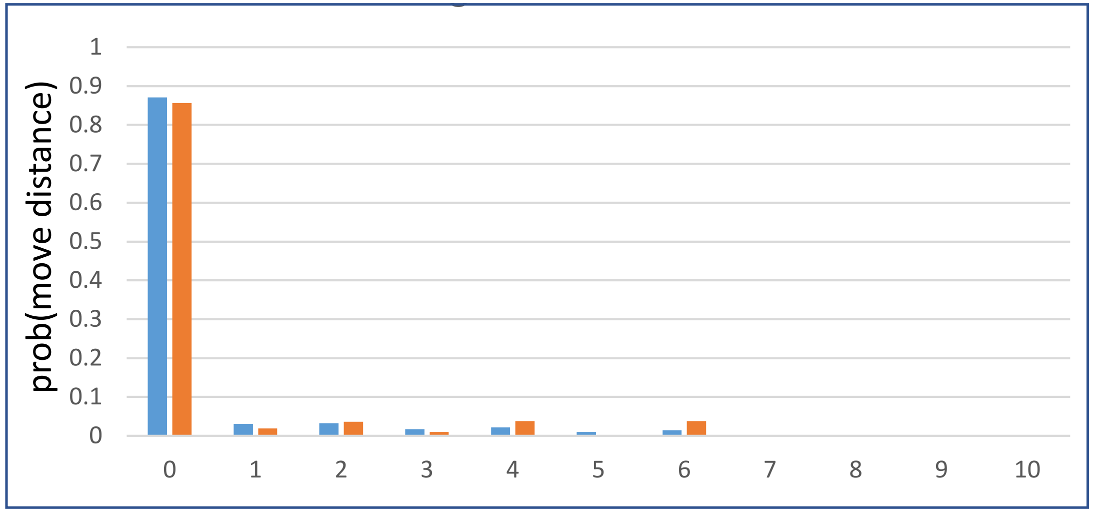


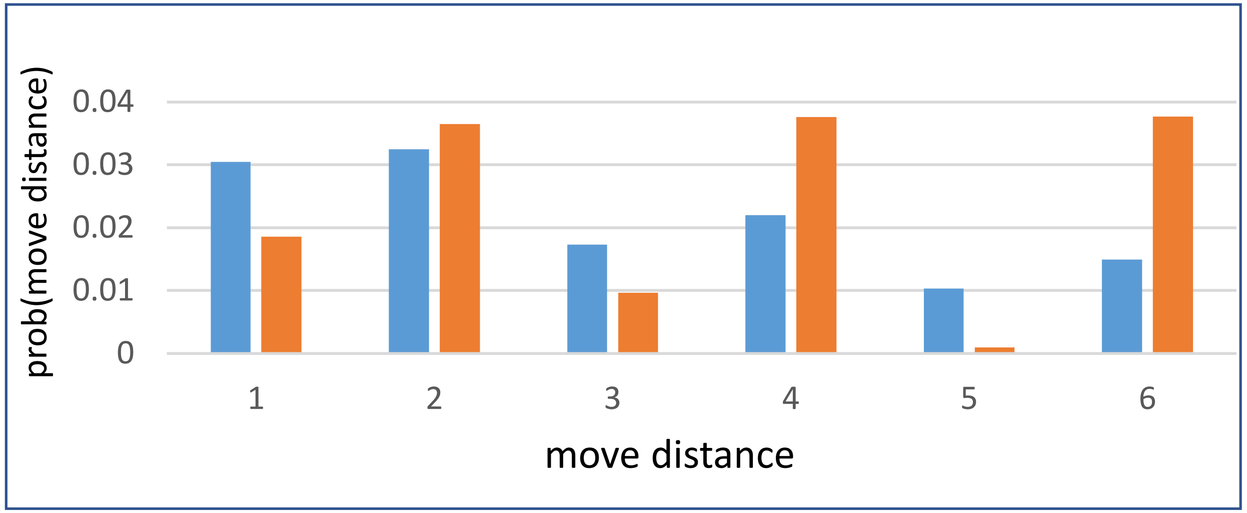
Approved for Public Release – Distribution Unlimited

Example Simulation Results

Reordering Distributions for Packets Passing Through Both Segments

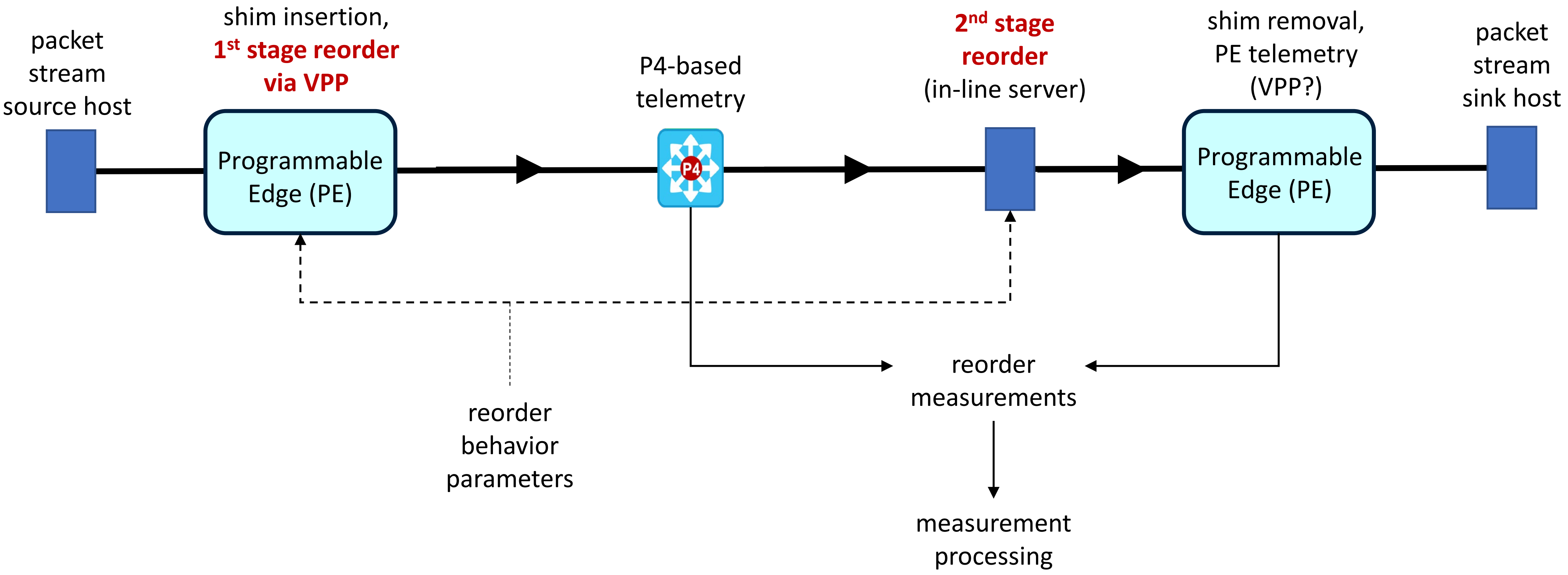
🟦 = Full-Path Simulation Output 🟧 = Prediction from Convolution





- Graphs are from the same experiment, but right-hand graph expands detail for move distances 1-6
- The convolution prediction has limited point-by-point accuracy
  - So, the effects of cascaded reordering stages are not strictly additive
- Effective diagnostics are still feasible (the sums across all n > 0 data points are within 1%)
- More experiments are necessary to explore these techniques in greater detail

HW Testbed for Reordering Experiments (in progress)


packet stream source host
shim insertion, **1st stage reorder via VPP**
Programmable Edge (PE)
P4-based telemetry
P4
2nd stage reorder (in-line server)
shim removal, PE telemetry (VPP?)
Programmable Edge (PE)
packet stream sink host
reorder behavior parameters
reorder measurements
measurement processing

- Reordering behaviors (**red**) configured for each experiment, and across a wide range
- Measurements taken at PEs and in programmable switch (via P4)
- ProD3 shim sequence numbers provide basis for reorder detection and characterization

Approved for Public Release – Distribution Unlimited

Recent Papers

- (Princeton) J.M. Cohen, Q. Wang, and A. Appel, “Verified Erasure Correction in Coq with MathComp and VST,” submitted for publication 1/21/22.
- (Princeton) M. Pan *et al.*, “A Mechanized Formalization of P4: Language Specification and Program Verification,” submitted for publication 2/2/22.
- (USC) T. Zhang *et al.*, “Federated Learning for Internet of Things: Applications, Challenges, and Opportunities,” https://arxiv.org/pdf/2111.07494.pdf.
- (USC) A. Hekmati, E. Grippo, B. Krishnamachari, "Dataset: Large-scale Urban IoT Activity Data for DDoS Attack Emulation" *4th ACM DATA Workshop on Data: Acquisition To Analysis (DATA'21), in conjunction with ACM BuildSys*, November 2021.
- (USC) A. Hekmati, E. Grippo, B. Krishnamachari, "Neural Networks for DDoS Attack Detection using an Enhanced Urban IoT Dataset", *Communication & Information System Security Symposium, IEEE ICC 2022*, in submission.

(multiple other papers on DDoS defense in preparation)

Plans for Phase 2

- Software development and testing for defense against Mirai HTTP Flood, leveraging insights that we have gained from Phase 1 simulation
- Further refinement of existing defensive software for UDP and SYN Floods
- Completion of software for defeat of packet dropping and reordering disruptions; demos in hardware testbed
- USC: continue FedML/FedIoT research in support of DDoS defenses
  - Leverage their new urban IoT dataset for the FedML platform
  - Optimize time-series forecasting algorithms to work on smaller, IoT-like footprints
  - Demonstration of a lightweight, on-device training engine for the IoT edge
- Princeton:
  - Move software verification beyond C-based VNFs and into P4
  - Continue development of DDoS defenses leveraging data-plane programming

ProD3 Transition Activity (work in progress)

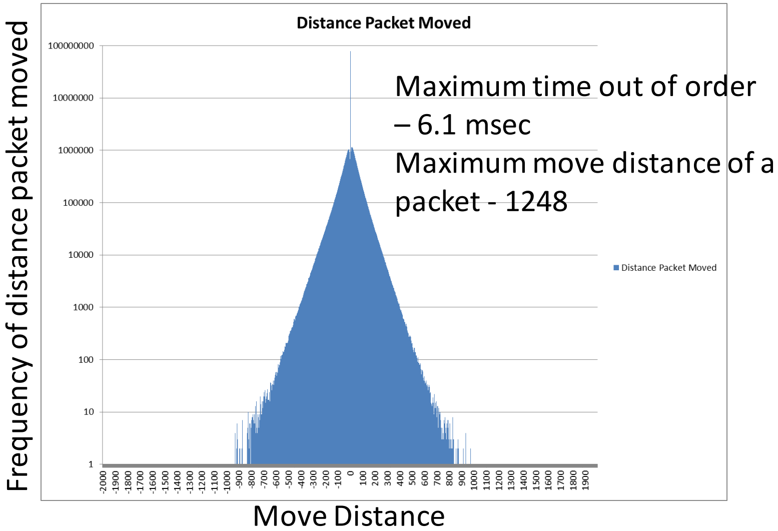
- Continuing discussions with Linux Foundation regarding open source plans
- Intent: work with TA3 team and Lumen to identify useful features and requirements for its core network, to drive specific components and interfaces for inclusion in open source software development pipeline
- Additionally, DREN could be a useful venue for prove-in and transition of selected ProD3 components

Approved for Public Release – Distribution Unlimited

Backup

Characterization of Reordering Behavior

- Model the reordering behavior as a random process *r*
- Form a histogram showing the relative frequencies of each move distance *n*



Maximum time out of order – 6.1 msec

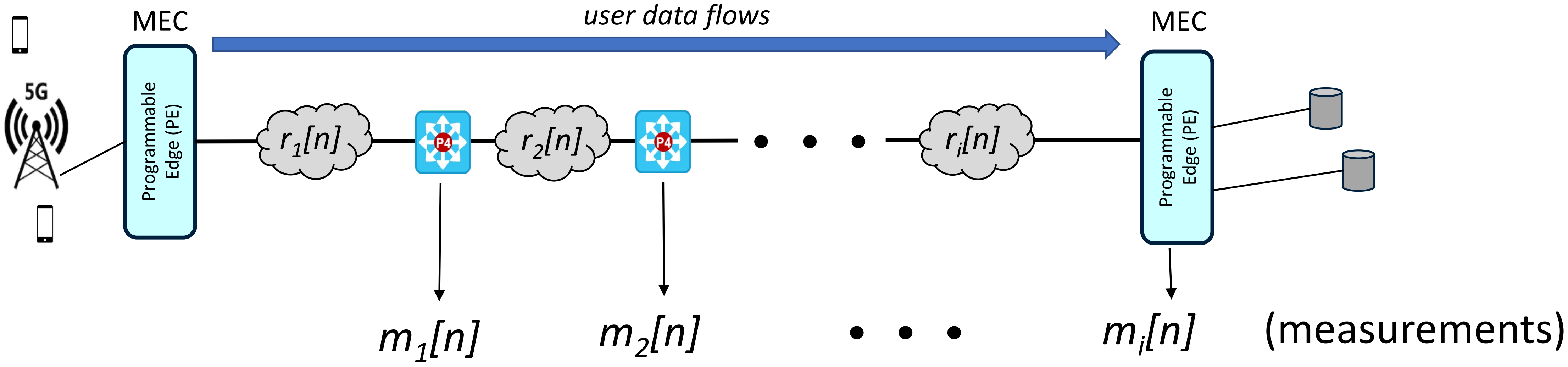
Maximum move distance of a packet - 1248

Diagnostic data from a DISA network (courtesy Doug Wood, ca. 2015)

- *r[n]* is then the (discrete) probability distribution for the random process *r*
- We want to know *r[n]* for each distinct segment of a network path

Approved for Public Release – Distribution Unlimited

Analytics: Isolating Segment Characteristics



$$m_i[n] \approx r_1[n] * r_2[n] * \cdots * r_i[n]$$ (convolution)

$$M_i(z) \approx \prod_{j=1}^{i} R_j(z)$$ (z transform)

$$r_j[n] = Z^{-1}\left(\frac{M_j(z)}{M_{j-1}(z)}\right)$$ (inverse z transform)

Approved for Public Release – Distribution Unlimited