Application of the UVA Dependability Assessment Methodology to Digital Reactor Protection Systems

June 22, 2010

For ORNL

Presented by:

Carl R. Elks
Center For Safety Critical Systems
University of Virginia
Department of ECE
Charlottesville, VA.

UNIVERSITY of VIRGINIA ENGINEERING

Overview of the Methodology

Carl Elks

University of Virginia
ENGINEERING

Current Situation: Digital I&C

- Issues
  - New digital technologies and more complex applications are being introduced into the nuclear industry
  - The new technologies can provide both operational improvement and increased reliability and safety, if implemented properly
- The advantages this new technology comes at a price.
- Significant challenges
  - Increased complexity → Impedes understanding
  - Potentially new failure modes→ How do we Know them
  - Fundamentally different behavior from analog system→ Continuous vs Discrete.
- Ever-increasing use of these new digital technologies and complex applications requires effective, objective, and repeatable review processes.
  - Tools and methodologies needed for independently evaluating new applications.

3

UNIVERSITY of VIRGINIA ENGINEERING

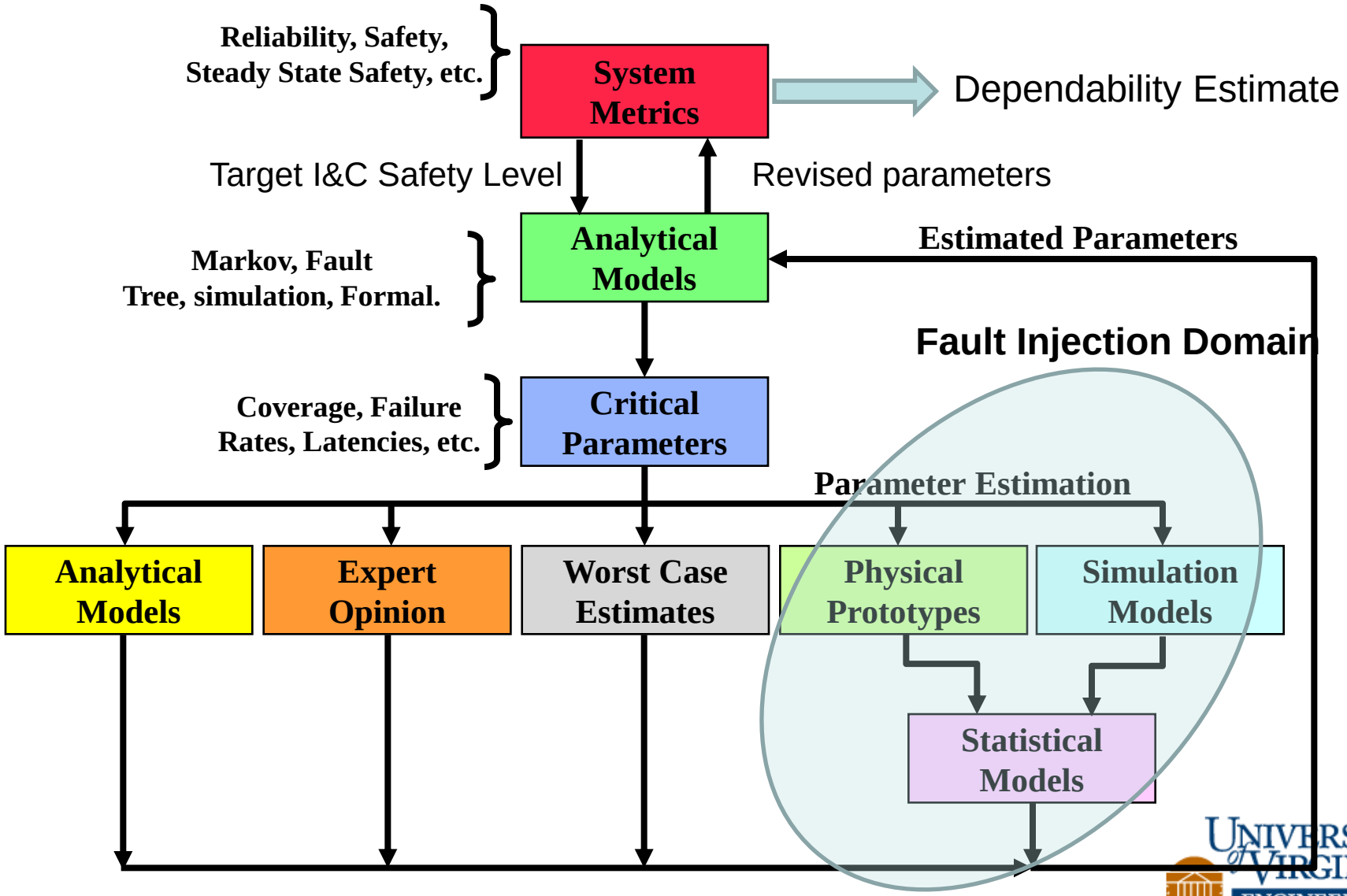
Aim of the Methodology

- A means to validate and verify objectively the dependability attributes of a digital I&C system
  - Support the estimation of quantitative metrics
  - Support the evaluation of qualitative attributes
  - Support modeling activities (PRA, Reliability, and Safety)
- Rigorous in it's foundation
- Strive for practicality and utility.
- Provide useful information to assessors, vendors, and regulators concerning the safety and reliability attributes of a digital I&C system.

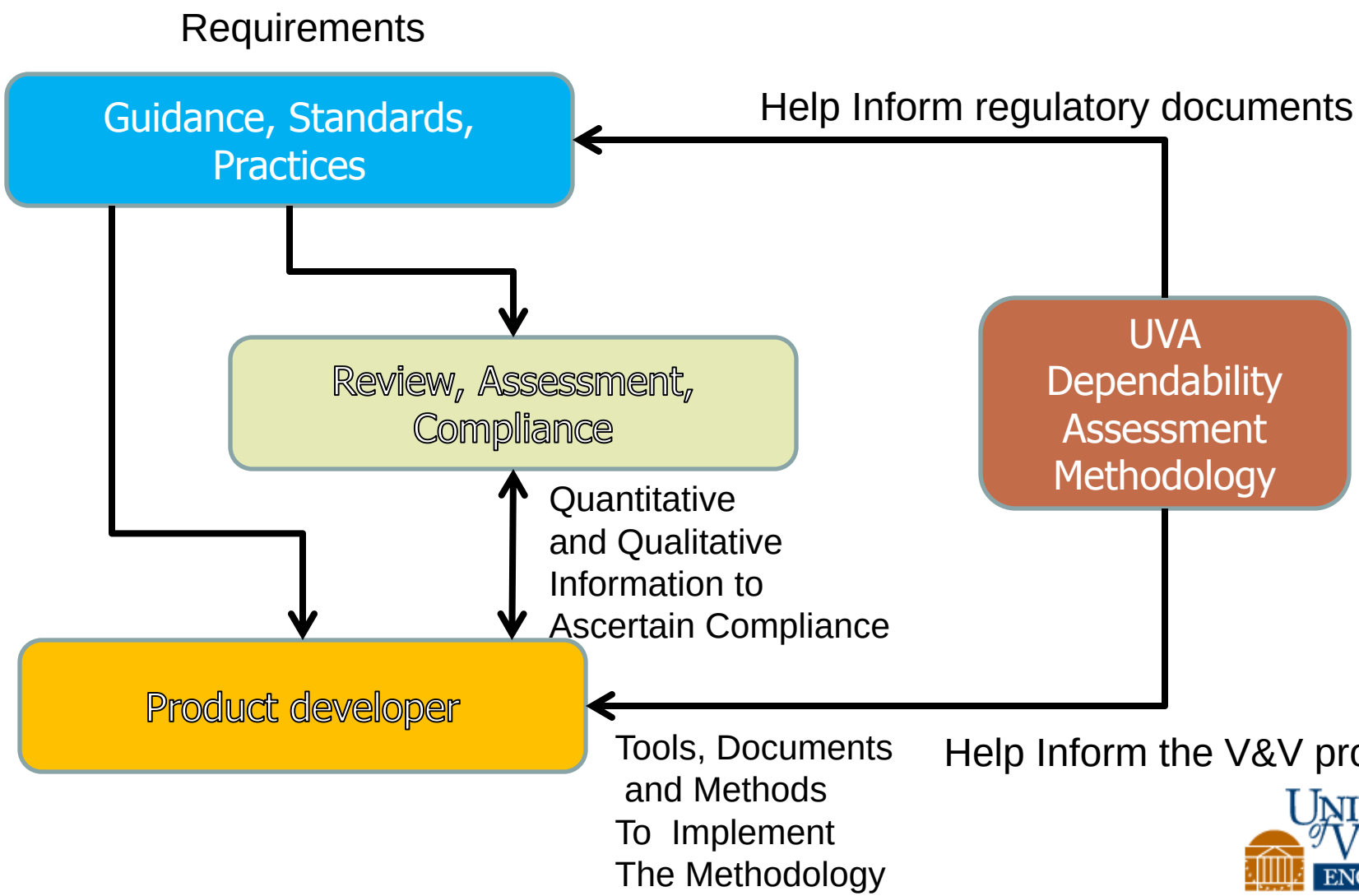
Technical Goals of the Methodology

- Develop, Apply and improve a safety and reliability assessment methodology for digital and computer based I&C systems
  - Consider the integrated hardware/software system
  - Consider bus and network technology issues that may effect safety
  - Consider how different modeling methods can be accommodated.
- Develop modeling, simulation, and fault injection techniques that support the assessment methodology
- Document the quantitative and qualitative results that can be obtained through application of the assessment process, and provide the technical basis upon which NRC can establish the regulatory guidance.
- **Develop tools that support the practical implementation of the assessment methodology**

Functional View of the Methodology

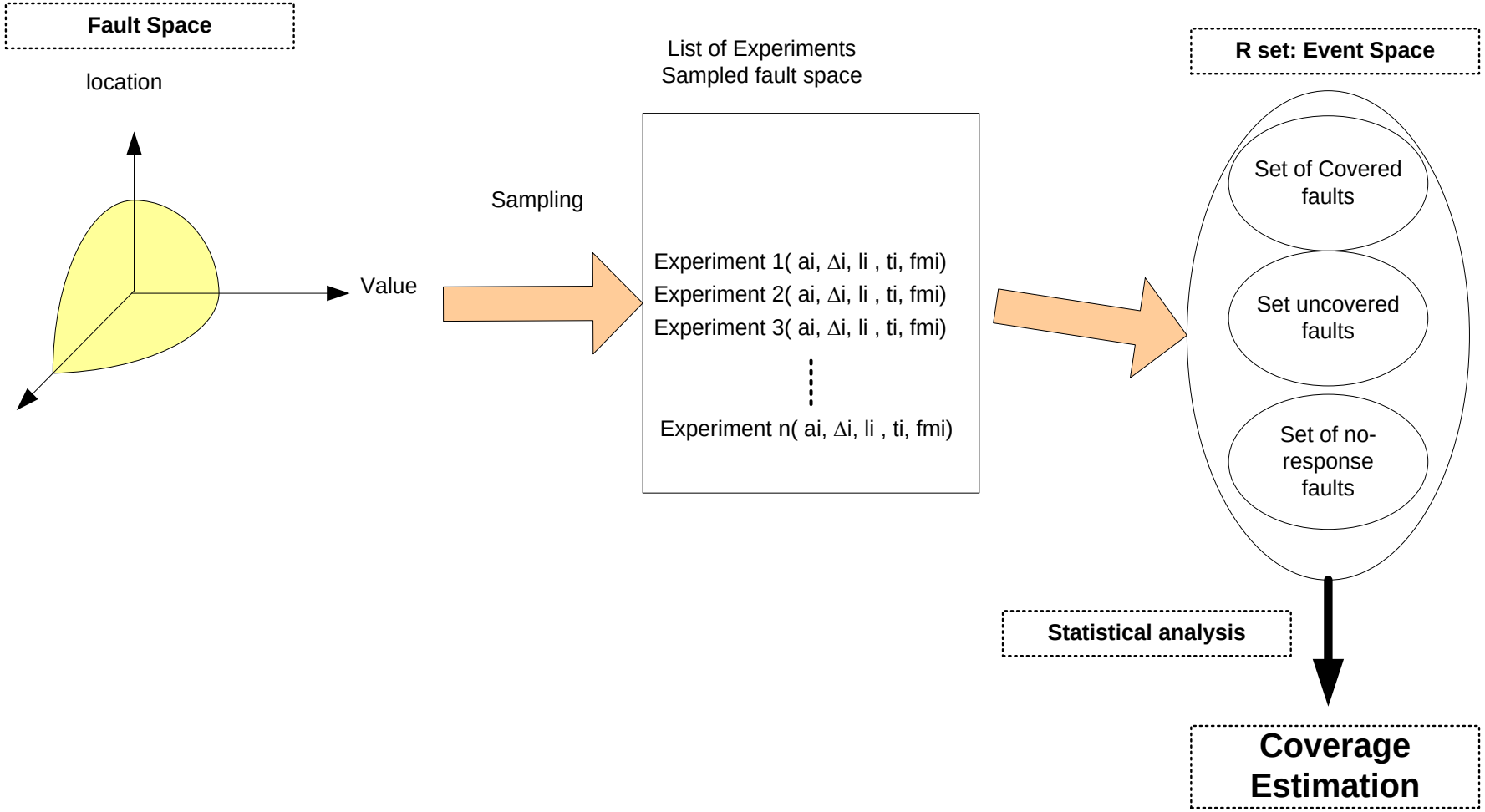


How the Methodology Can Work in a Regulatory Framework



University of Virginia ENGINEERING

Big Picture of the Fault Injection Process



Fault Coverage Basics

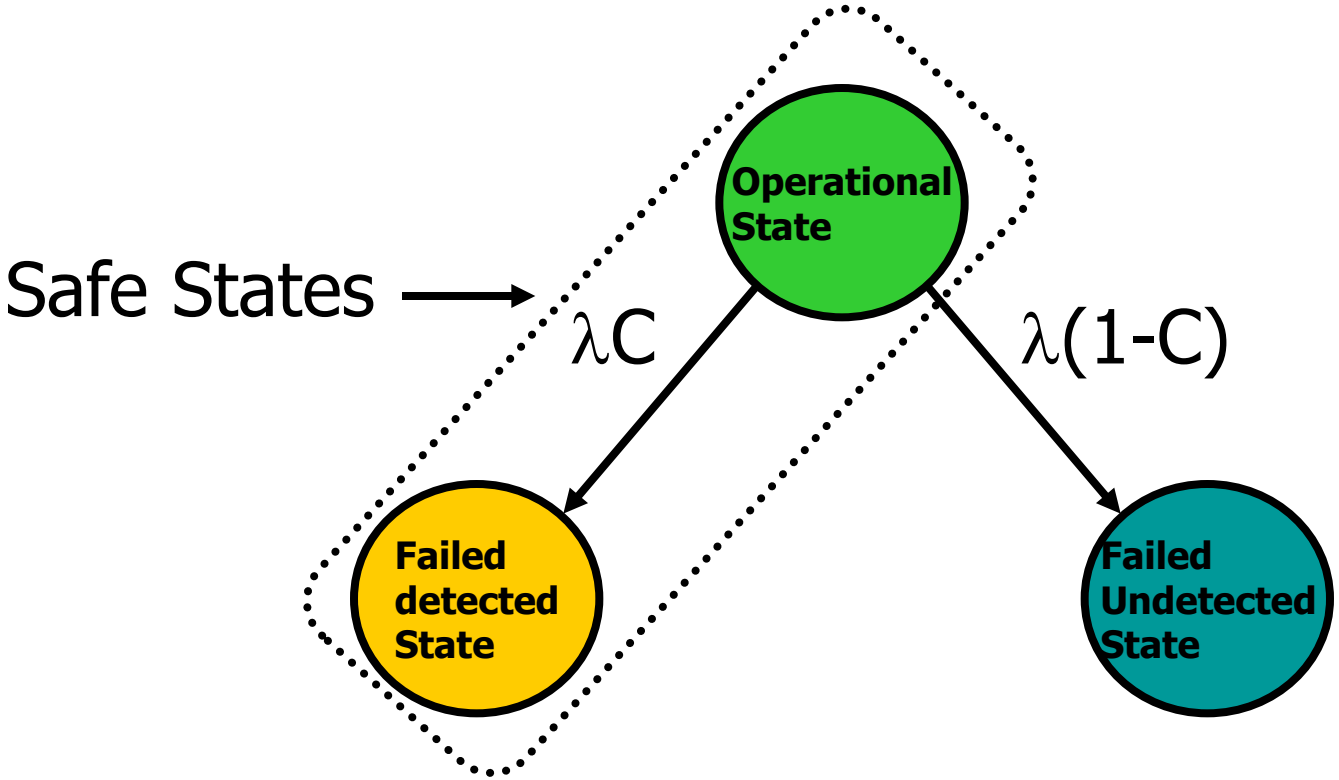
The point estimate for the system fault coverage is obtained

$$\hat{C} = \frac{1}{n}\sum_{i=1}^{n} y(t_i, \Delta_i, \sigma_i, l_i, f_{m_i})$$

- where C^ is the point estimate for the system fault coverage, and *n* is the number of fault injection experiment.
- From Equation 1 we can see that the estimation of fault coverage via fault injection is dependent on five parameters:
  - Fault type, Location, Time, Duration, and Workload profile
- These parameters are dependent experiment attributes of the fault injection process.

University of Virginia ENGINEERING

Coverage Example in a Markov Model



Fault coverage, C , is the conditional probability that a system correctly handles a fault, given that a fault has occurred.

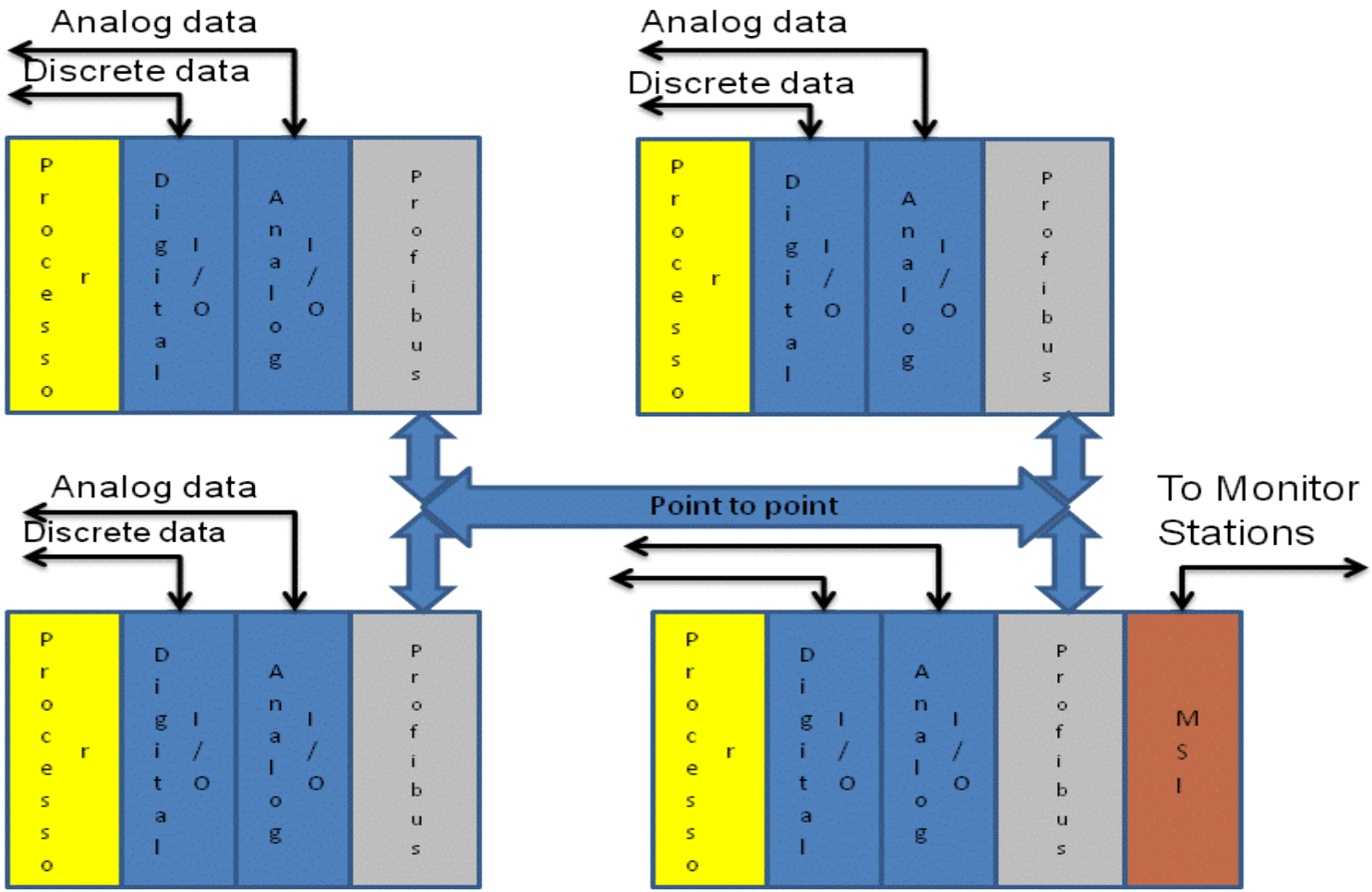
Overview of the Benchmark Architecture

Basic architectural principles.

- A federated distributed fault tolerant system
  - Channelized, symmetric, quasi asynchronous operation.
  - Cyclic prioritized static real time scheduler OS.
  - Safety functions utilize both application dependent and application independent fault tolerance features.
  - Highly configurable for different applications.
- Channels are coupled by Profibus DP network.
  - Event triggered, token passing network.
  - Worst case bounded operation wrt token management.
  - Non deterministic messaging service.
- Fault tolerance is achieved through several forms of redundancy and self monitoring.
- Fault containment and isolation is typically at the channel level.

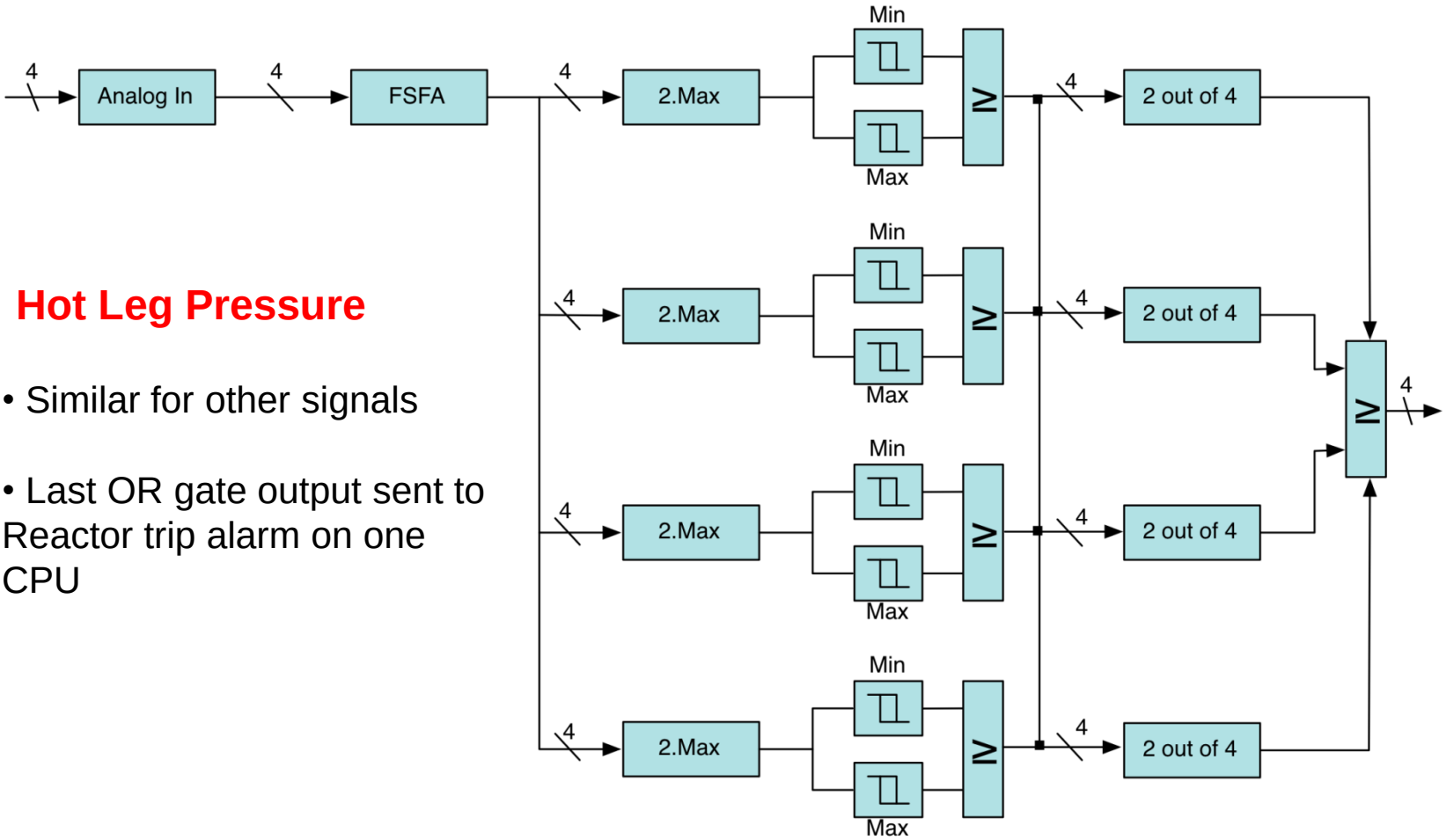
UNIVERSITY of VIRGINIA ENGINEERING

Platform Architecture



UNIVERSITY of VIRGINIA ENGINEERING

RPS Functional Diagram



Hot Leg Pressure

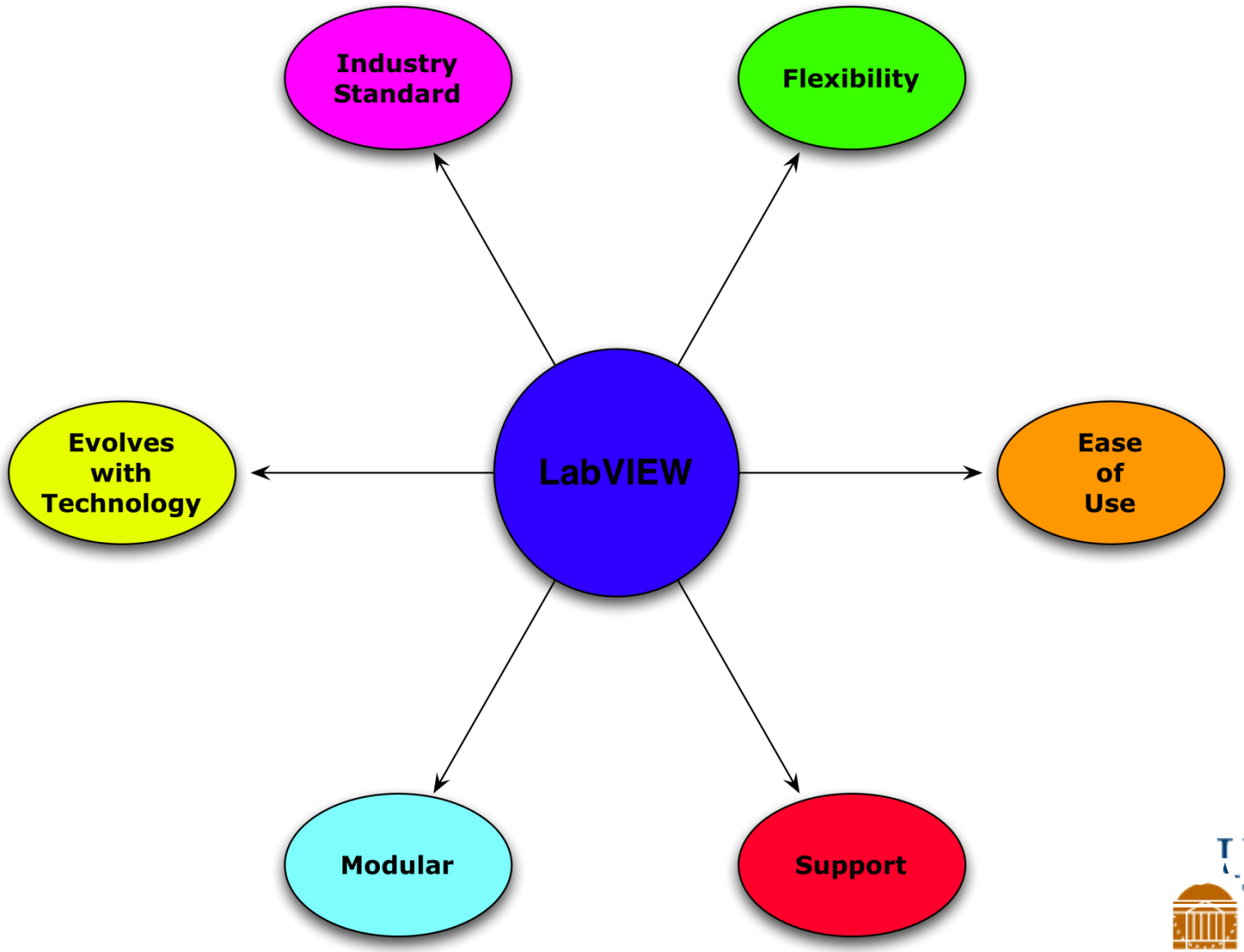
- Similar for other signals
- Last OR gate output sent to Reactor trip alarm on one CPU

Fault Injection Environment

- Uses LabVIEW 8.6 Development Tool
- Coordinates the following functions
  - analog and digital data acquisition
  - external system reset
  - initiation of fault injection campaign
  - signal-level corruption
  - data logging and analysis

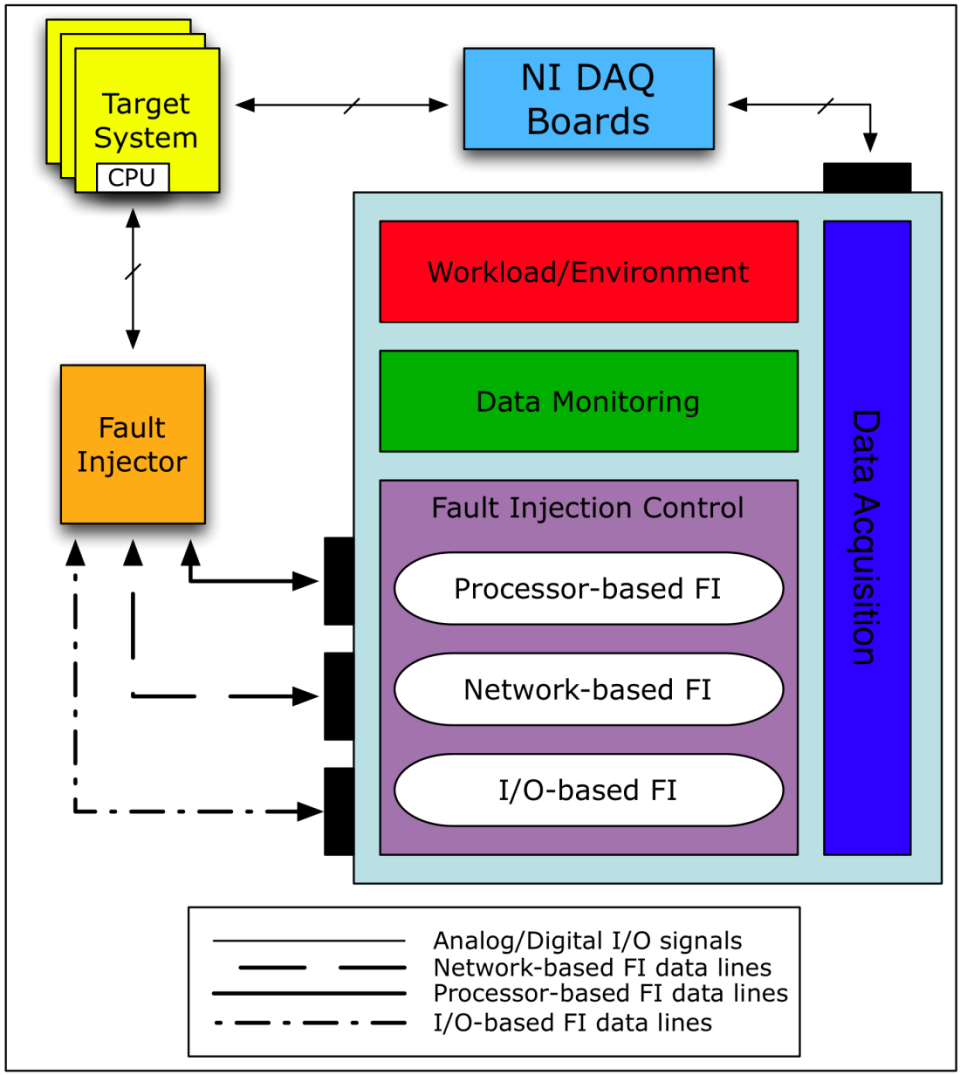
University of Virginia ENGINEERING

FIE Design Goals



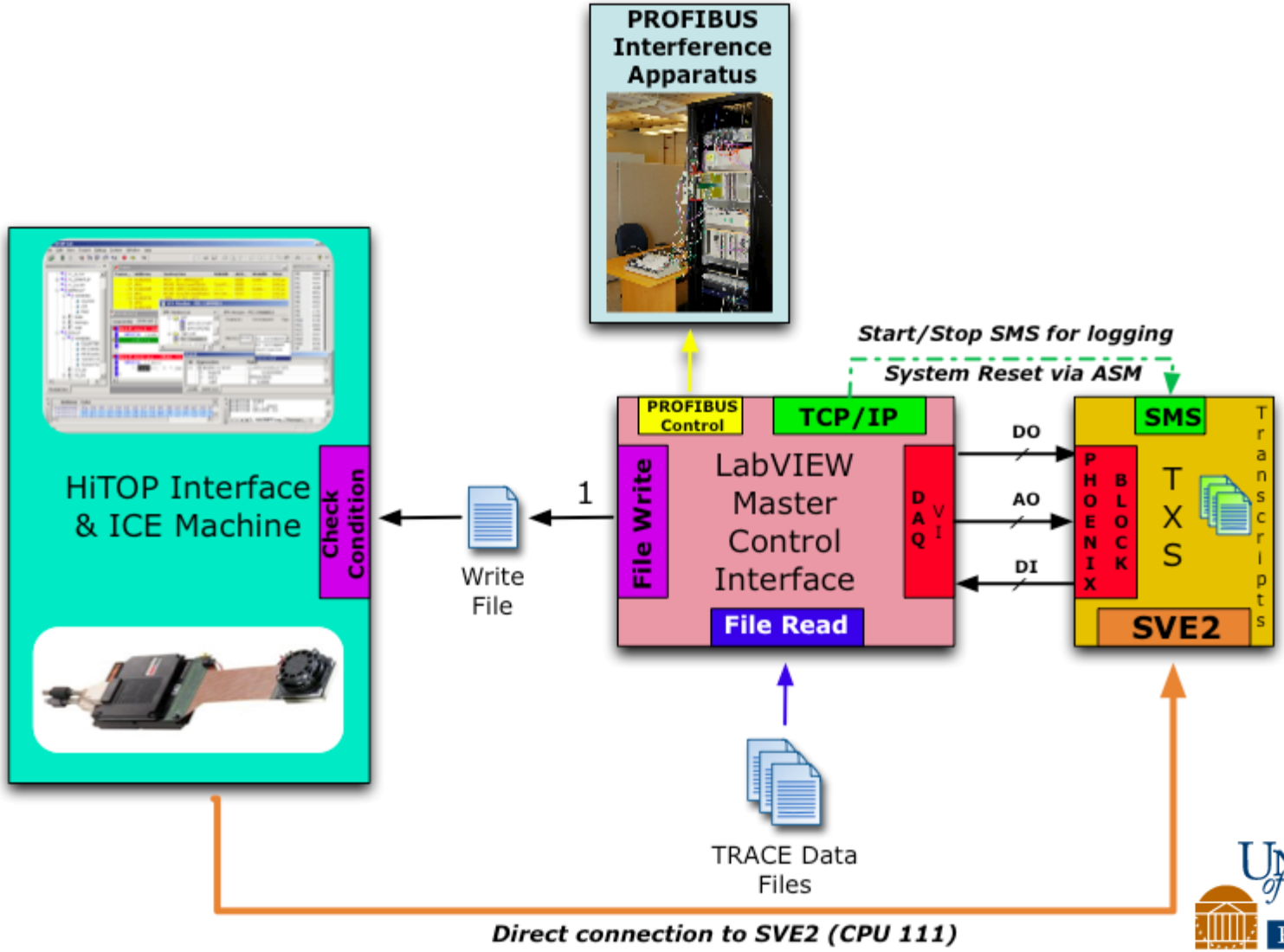
UNIVERSITY of VIRGINIA ENGINEERING

FIE Architecture



UNIVERSITY of VIRGINIA ENGINEERING

Fault Injection Environment for Benchmark Testing



UNIVERSITY of VIRGINIA ENGINEERING

FAULT INJECTION ENVIRONMENT

ENVIRONMENT PARAMETERS

RESET PERIOD: 30
Amplitude of Noise: 1
Number Of Faults: 531
PERIOD OF FILE READ: 0.05
Kill Port 1: 6452
SMS TCP/IP PORT: 6432

PROFIBUS INJECTION CONTROLS

EXPERIMENT SELECTOR
Profibus File Read Time (ms): 1

MISCELLANEOUS

FAULT ITERATION: 282
EXTRA INPUT (UNUSED)
TIME INDEX: 13
COUNTDOWN TO INJECTION: -40

SENSOR VALUES

COOLANT FLOW

CF A: 0
CF B: 0
CF C: 0
CF D: 0

HOT LEG PRESSURE

HLP A: 0
HLP B: 0
HLP C: 0
HLP D: 0

STEAM GENERATOR PRESSURE

SGP A: 0
SGP B: 0
SGP C: 0
SGP D: 0

OUTPUTS AND ALARMS

COOLANT FLOW

COOLANT FLOW TRIP: 0

PRESSURIZER PRESSURE

PRESSURIZER PRESSURE TRIP: 0

STEAM GENERATOR PRESSURE

STEAM GEN. PRESSURE TRIP: 0

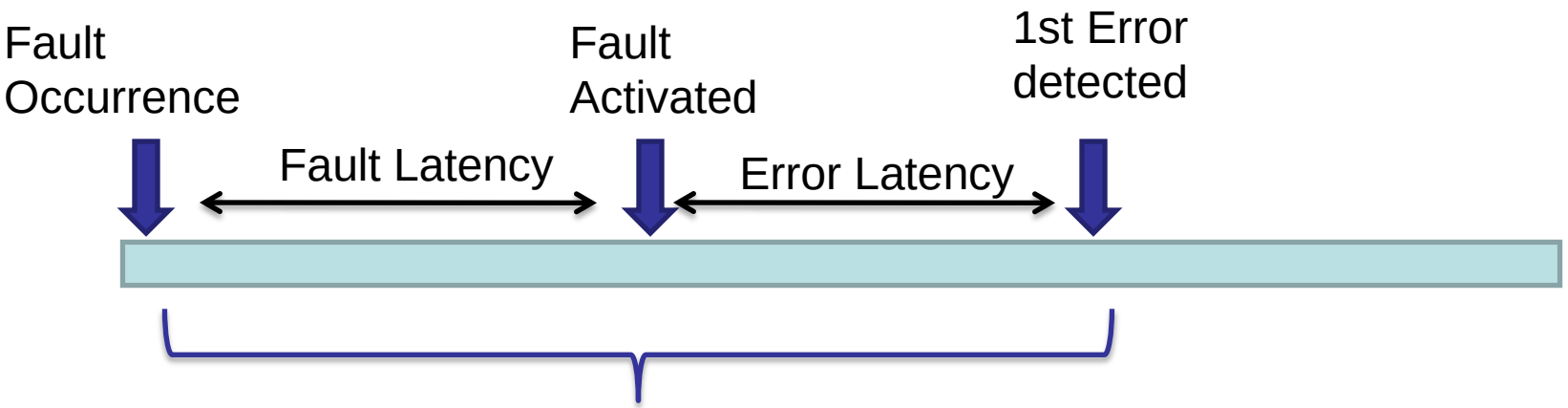
UNIVERSITY of VIRGINIA ENGINEERING

Processor Fault Injection Analysis and Results.

Carl Elks and Nishant George

University of Virginia
ENGINEERING

Fault and Error Latency

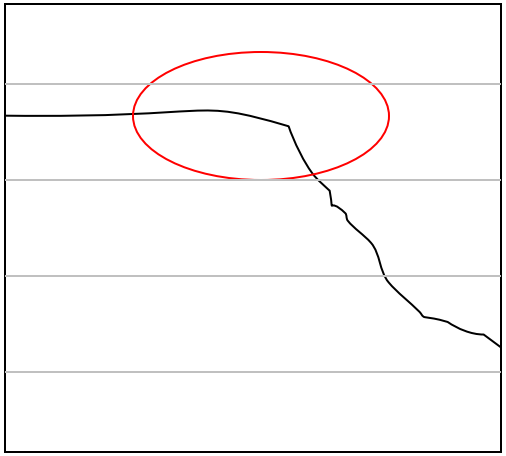


The total time is observed from a fault injection experiment

University of Virginia ENGINEERING

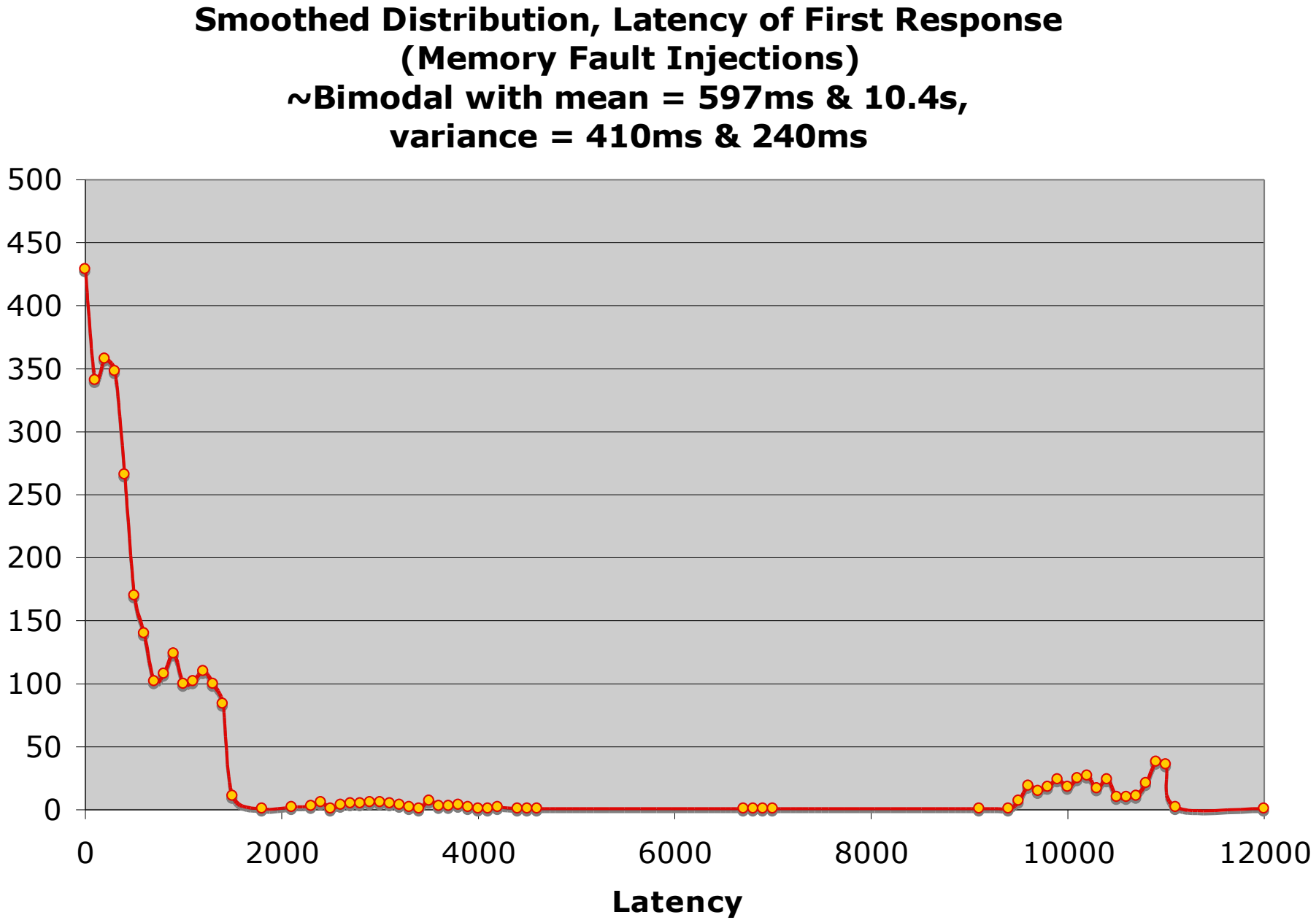
Nature of faults injected

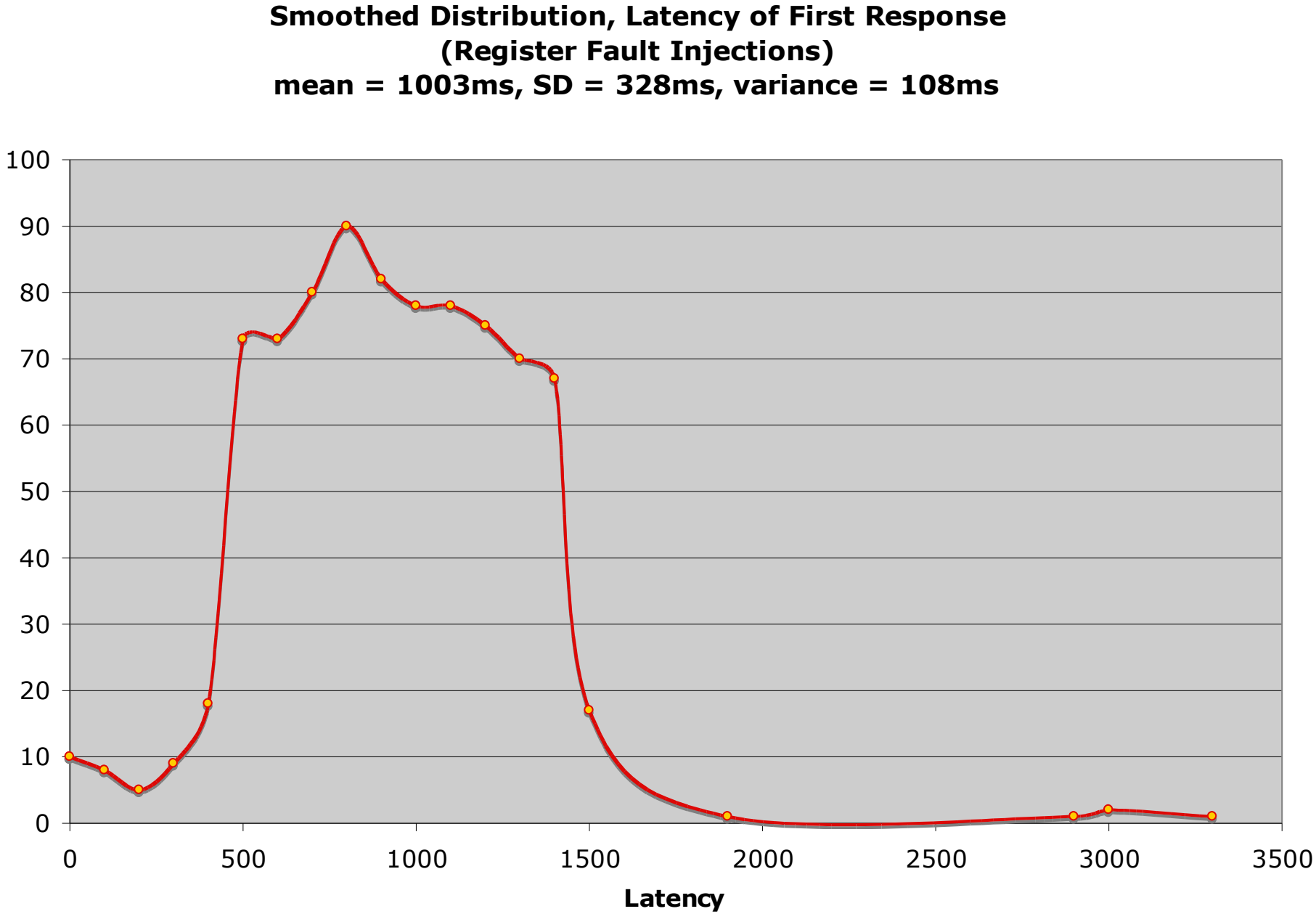
- Faults injected into neighborhood of possible alarm trip events
- Faults injected into
  - Locations in app. map
  - Locations from analyses



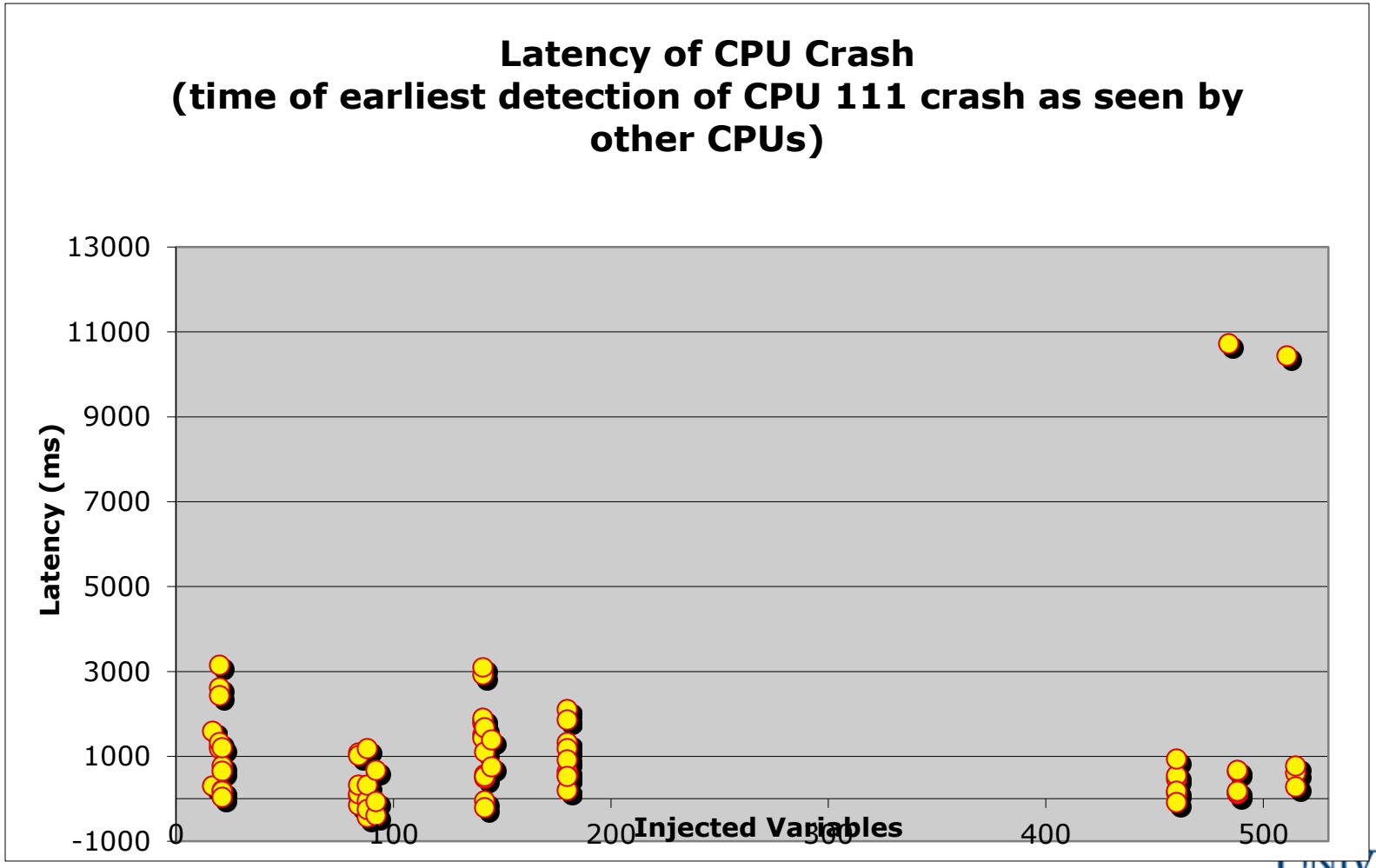
LOCA Event

University of Virginia ENGINEERING


Smoothed Distribution, Latency of First Response
(Memory Fault Injections)
~Bimodal with mean = 597ms & 10.4s,
variance = 410ms & 240ms
500
450
400
350
300
250
200
150
100
50
0
0
2000
4000
6000
8000
10000
12000
Latency

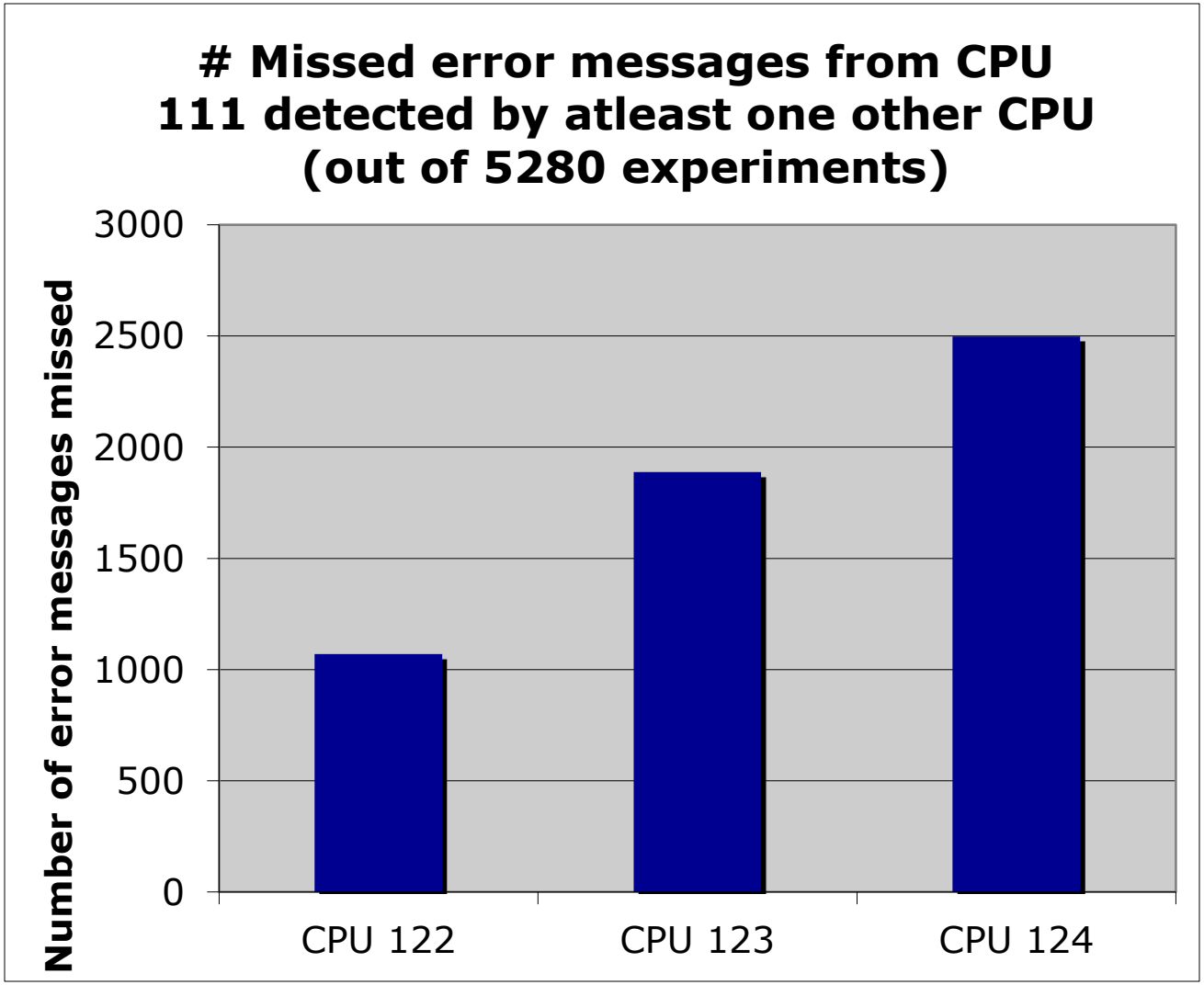

Smoothed Distribution, Latency of First Response
(Register Fault Injections)
mean = 1003ms, SD = 328ms, variance = 108ms
100
90
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
2500
3000
3500
Latency

Crash latency



UNIVERSITY of VIRGINIA ENGINEERING

Inconsistency of Error Detection



UNIVERSITY of VIRGINIA ENGINEERING

Coverage and Error Detection Latency Statistics for Processor Fault Injections

Location	**Point Estimate of C**	**Variance Of C**	**Confidence Bounds of C**	**Mean Error latency**	**Total fault Injections**	**No response faults**
Memory	.999697	9.1883×10^{-8}	$.999102 < C < 1.0$	Bimodal Mean with 597ms and 10.4 s.	5274	1975
Registers	.9998	1.39×10^{-6}	$.9965 < C < 1.0$	1003 ms	1470[1]	622

1- partial set of register fault injections.

UNIVERSITY of VIRGINIA ENGINEERING

Profibus Fault Injection and Results

Marko Miklo, Carl Elks, and Ron Williams

University of Virginia
ENGINEERING

Fault Injection Approach

- Inject faults to induce specific failure modes – failure mode driven fault injection.
  - Known faults that will cause Token to be corrupted and rejected by the receiving Master station
  - Data Message fault injection increases the "Length" field of the message
    - Preserves all the data, tricking the receiving controller
    - Data passed to the next layer -> we don't know what happens then
  - Observe behavior and response of the network
  - Measure the timeout after each Token loss
- These are the problems that we simulated in our fault injections

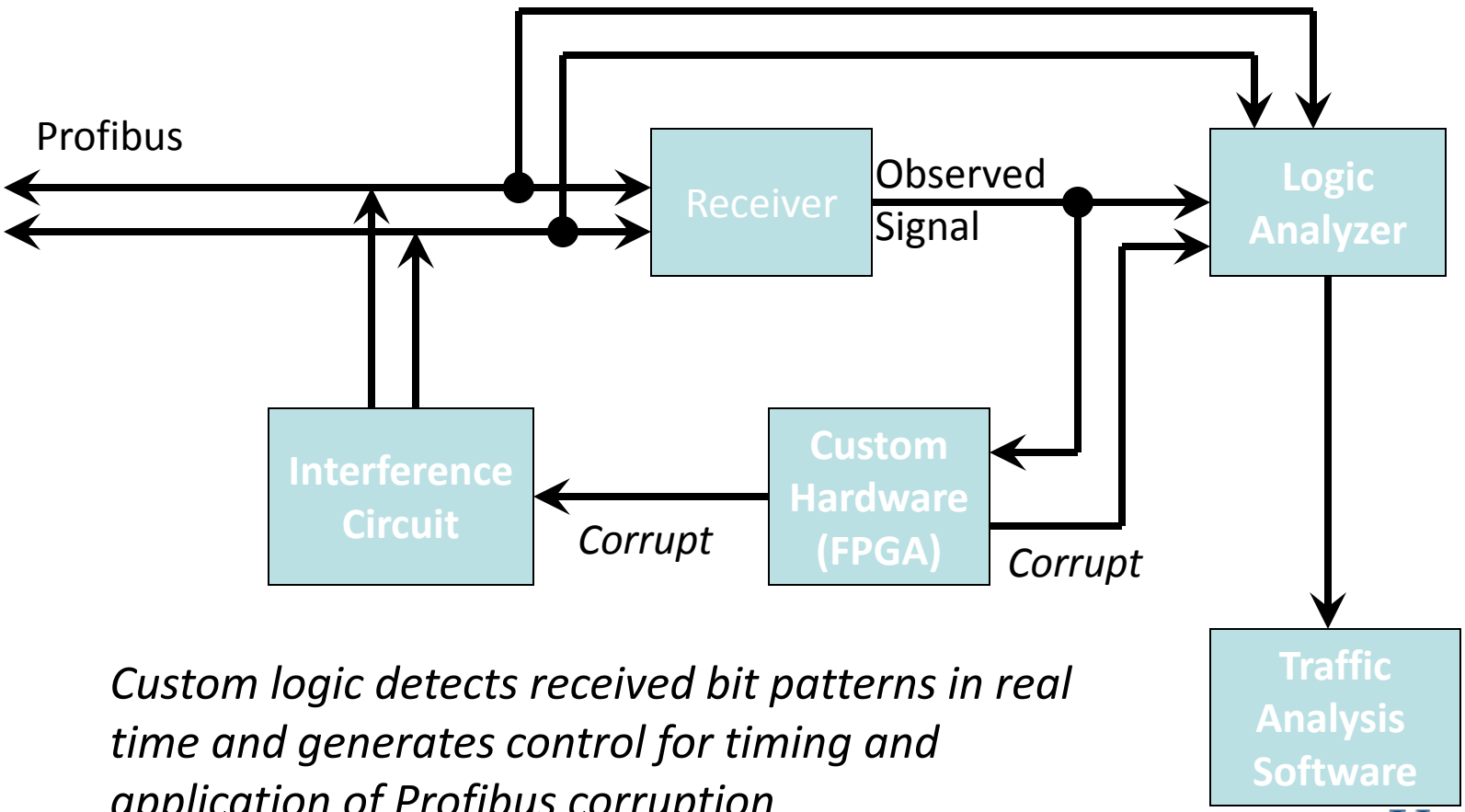
University of Virginia ENGINEERING

Covering Faults

- Fault/Error Model encompasses
  - Misbehaving Profibus controller
  - Faulty Profibus transceivers
  - Faulty links

University of Virginia ENGINEERING

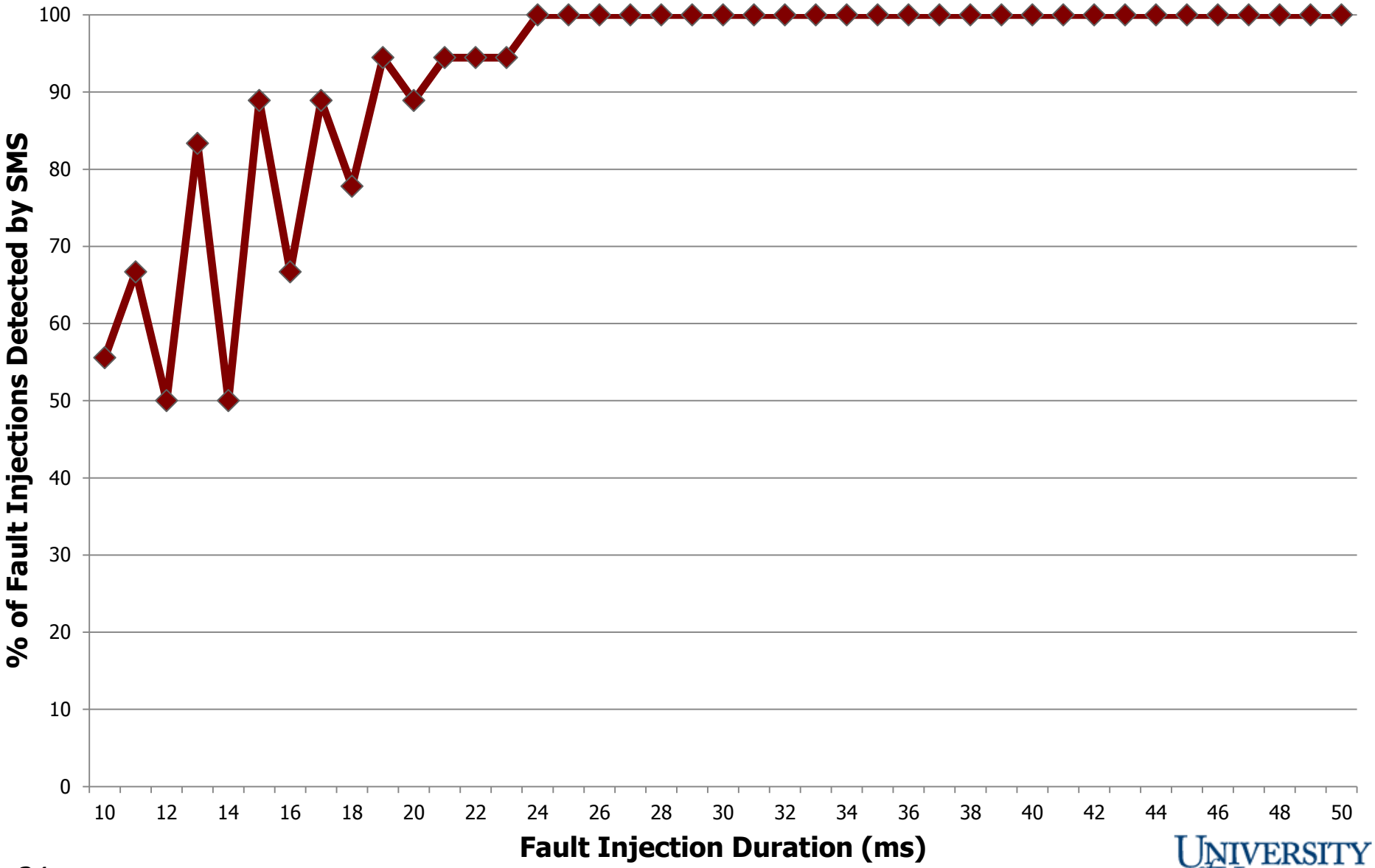
Profibus Fault Injection Functional Representation



Custom logic detects received bit patterns in real time and generates control for timing and application of Profibus corruption

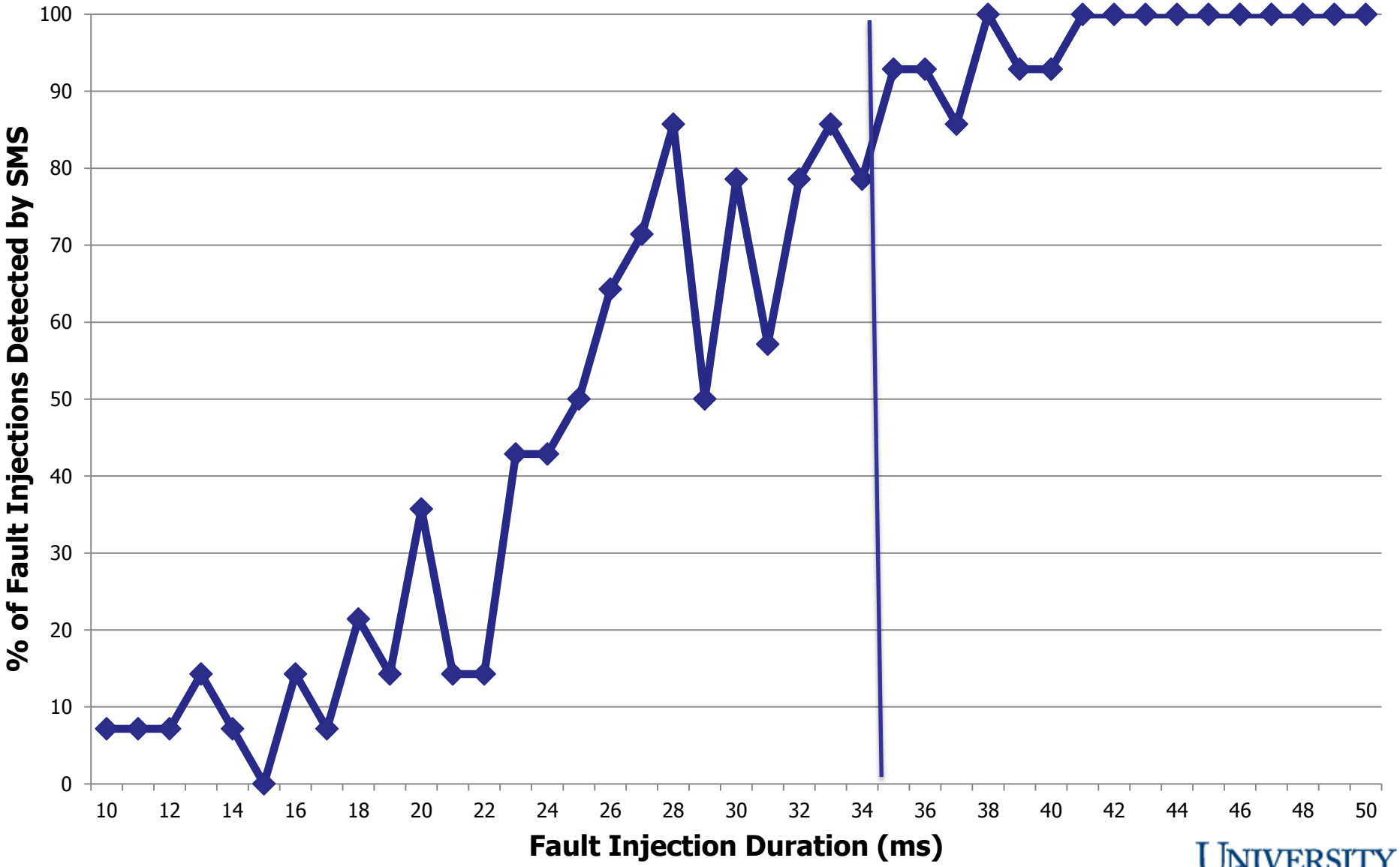
University of Virginia Engineering

Token Fault Injection Coverage



UNIVERSITY of VIRGINIA ENGINEERING

Data Message Fault Injection Coverage



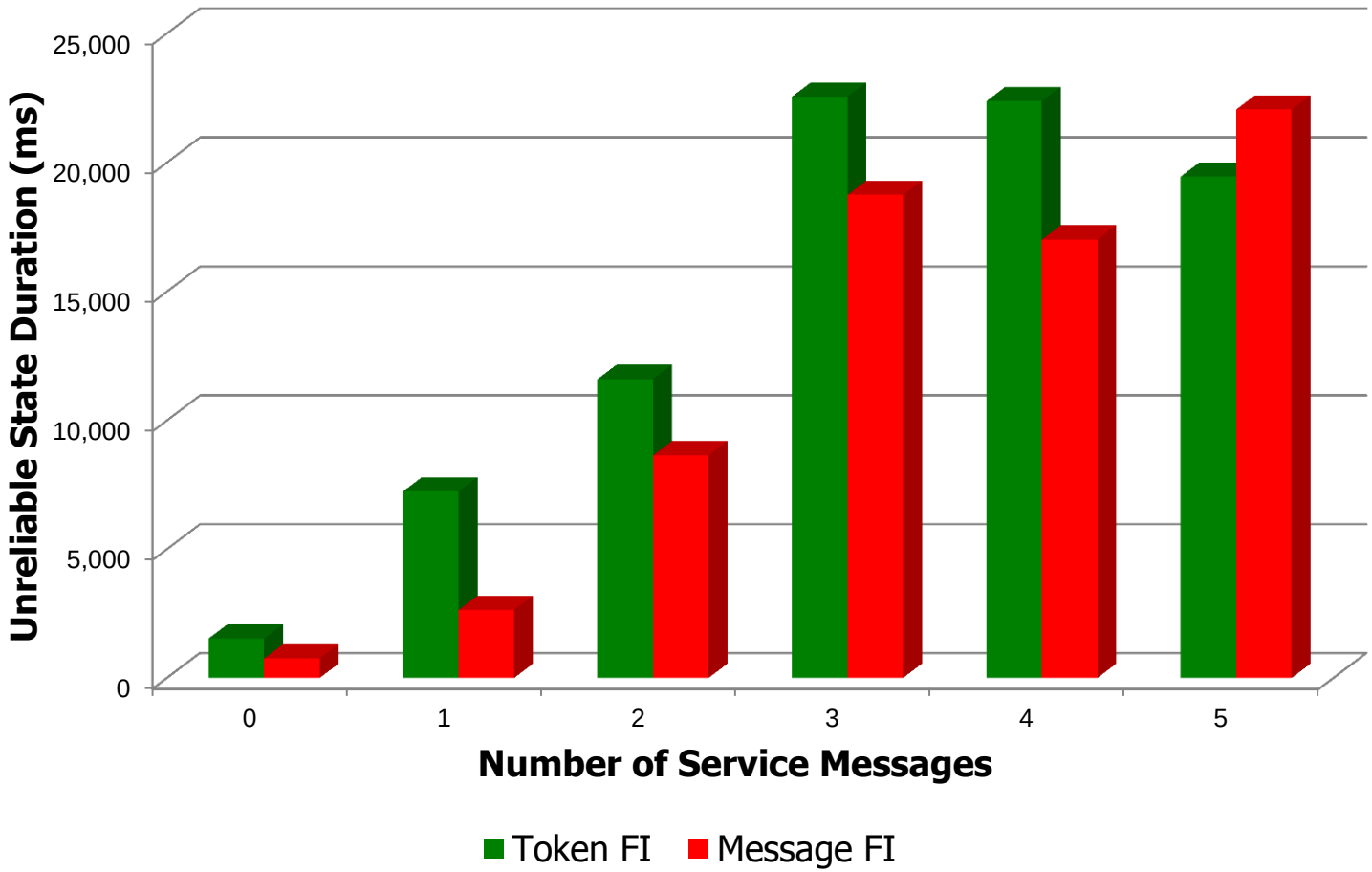
UNIVERSITY of VIRGINIA ENGINEERING

Profibus Controller Outage Coverage

- We can measure this exactly
- From observations, it is highly dependent on number "Service Messages" in the system
  - We did not control this parameter, just observation from the data.
- As expected, Token FI have much longer "*Unreliability time"* than Data Message FI
  - This is due to the time required to reinsert the Master that was taken out of the Logical Ring

University of Virginia ENGINEERING

Token FI vs Message FI Outage Coverage



# Service Msgs	Token FI (ms)	Message FI (ms)
0	1,531.51	756.68
1	7,244.86	2,639.03
2	11,607.72	8,644.00
3	22,548.86	18,745.29
4	22,376.83	17,014.75
5	19,454.00	22,053.00

UNIVERSITY of VIRGINIA ENGINEERING

Significant Findings to Date

- Methodology was successfully deployed.
- Methodology aided us (significantly) in understanding how the Benchmark fault tolerance and safety features work in real time.
- Fault injection campaigns produced results that were used to estimate critical parameters in the availability, reliability and safety models.
- Some of the results could not have been predicted or anticipated ahead of time with traditional testing. (results are still preliminary)
  - Bimodal error latency distribution
  - Service message impact on profibus controller outage (still trying to figure why this is).
  - Error perception inconsistencies
- Benchmark main fault tolerance features appear to be robust to the faults we injected into the system.
- A few bugs found in SPACE development tool.

Significant Findings to Date

- Benchmark system is relatively complex and intricate in its functionality and programming.
  - Assessors, Plant I&C engineers would be better served to take engineering training (as opposed to operator training).
  - Documentation is operator oriented rather than system functionality oriented.
  - Runtime and system fault tolerant operation is somewhat difficult to piece together from the documentation set.
- Guidelines on how to configure the system in a safety application would be helpful. I&C Vendor has some documentation on this, we now have it.
- Faults injected into Profibus suggested that faulted profibus communication can affect system availability and reliability.
- Some profibus failure modes could be malicious (still looking into this).

What's next?

- Triconex system is on deck.
  - A different system architecture, will stress the methodology in different ways.
- NUREG is top priority.
- Have I&C Vendor's feedback on the preliminary testing results.
- Many lessons learned along the way.

Long term

- It's clear that open source tools to enable both vendors and regulators to leverage the methodology are paramount.
  - A set of tools to allow assessors and regulators to analyze, trend and access fault injection data from the vendors.
  - A set of open-source tools to allow the vendors to implement and integrate the methodology into their testing and V&V processes.
- Invensys and I&C Vendor both seem to agree that this would be helpful.
- We are already moving toward a direction of "tool boxes" in our methodology.

University of Virginia ENGINEERING