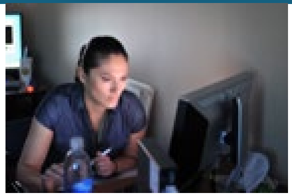
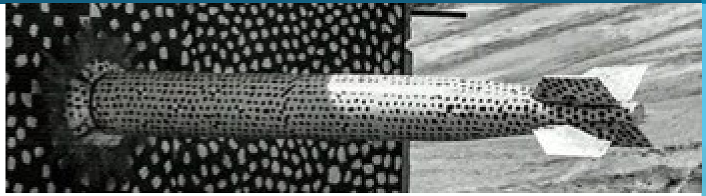
Sandia National Laboratories

Moving Target Defense for Space Systems





SNL: Chris Jenkins (PI), Eric Vugrin, Indu Manickum, Sarah Krakowiak, Richard "Grant" Brown, Jacob Hazelbaker, Nicholas Troutman, Josh Maxwell

Purdue: Prof. Bharat Bhargava, Marina Haliem, Ganapathy Mani

DATAWorks 2024

April 17, 2024

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND2024-04196PE

Agenda

Intro
- BLUF
- Who is Sandia?

Background
- Motivation
- Moving target defense
- MIL-STD-1553

Algorithm
- State Generation
- Usage
- Randomness Characterization
- Unpredictability Quantification

Experimentation
- Description
- Challenges
- Results

Machine Learning Attacks
- Methodology
- Results

Future Work
- IPv4 random port generator
- 2FA Passphrase Generator
- Transparent filesystem

Intro

BLUF

Accomplishments

- Patent awarded, 4 publications, multiple (invited) talks
- Obtained GUN copyright for MTD algorithm software
- 2023 R&D 100 Finalist

Key Results

- **Reduced adversarial knowledge by 97% during exfiltration cyber resilience experiment**
- Identified hopping frequency requirements to defeat exfiltration adversary capable of learning
- Quantified randomness and unpredictability of MTD algorithm
- Enhanced MTD's random sequence generator to defeat machine learning methods

Future Work

- IPv4 random port generator
- 2FA Passphrase Generator
- Transparent filesystem

U.S. National Laboratories

Sandia Has Two Main Locations

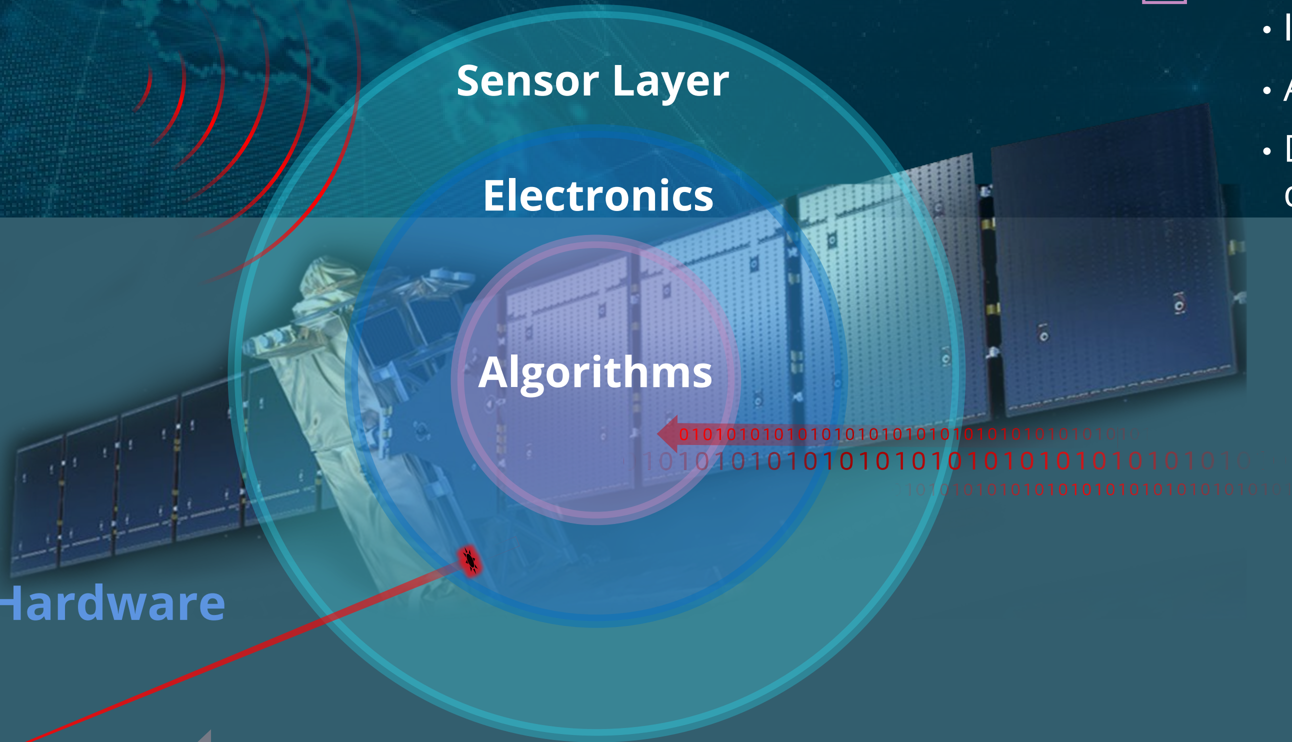
Science and Technology Advancing Resilience in Contested Space

| *STARCS Mission Campaign*

It's not Artificial Intelligence: *It's Artificial Instinct*

Sensor Protection

- Defensive Materials
- Data security/recovery
- Demonstrate protection

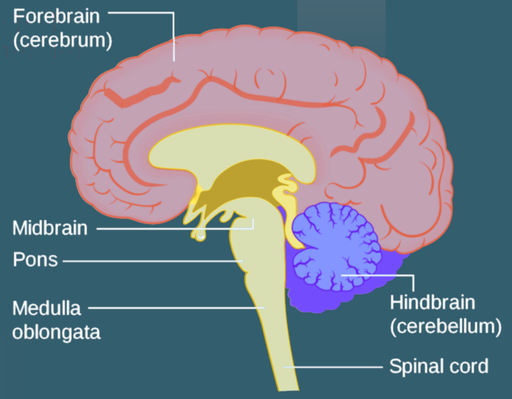


Cognitive Analytics

- Identify attacks
- Adapt to operate-through
- Demonstrate on threat-defended hardware

Threat-Defended Hardware

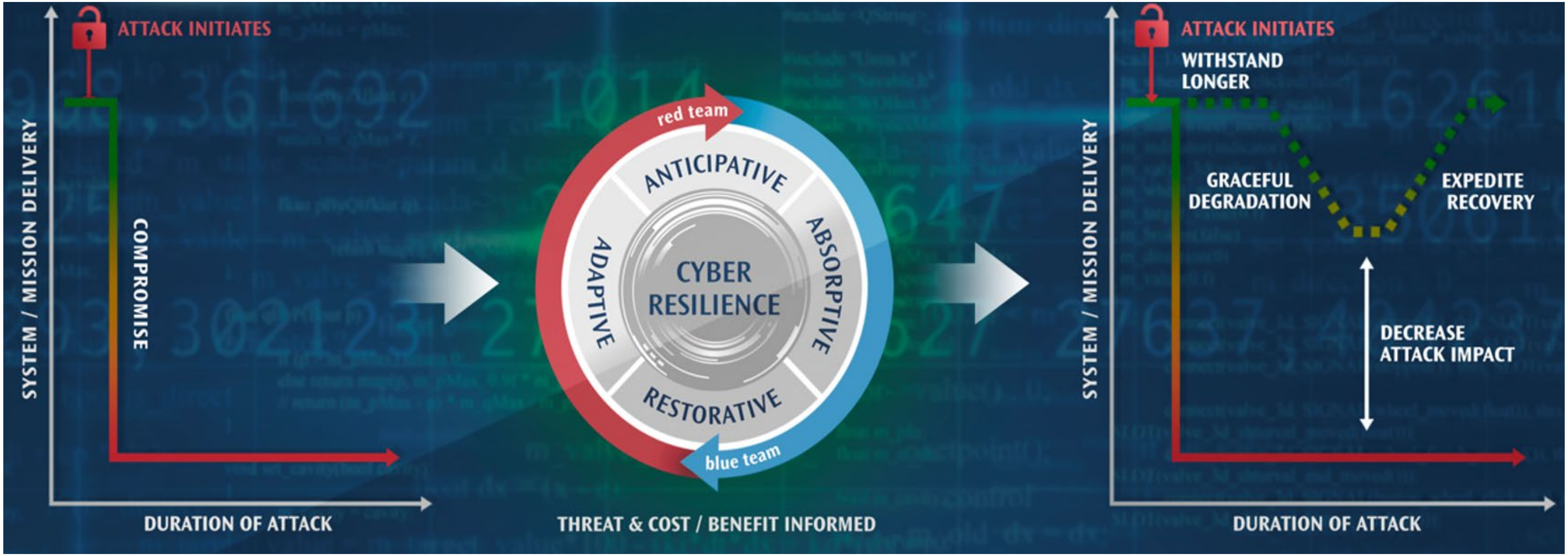
- Modsim Tools
- Hardened COTS
- Special Shielding
- Testable Hardware



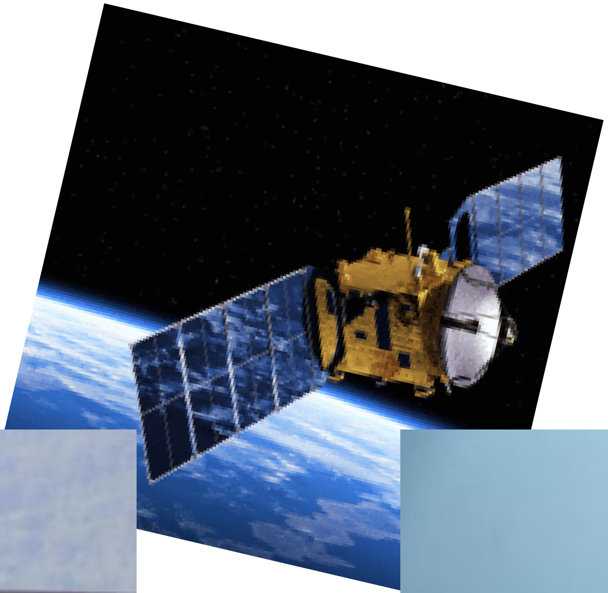
Neuromorphically-inspired

Background

Cyber Security vs. Cyber Resilience



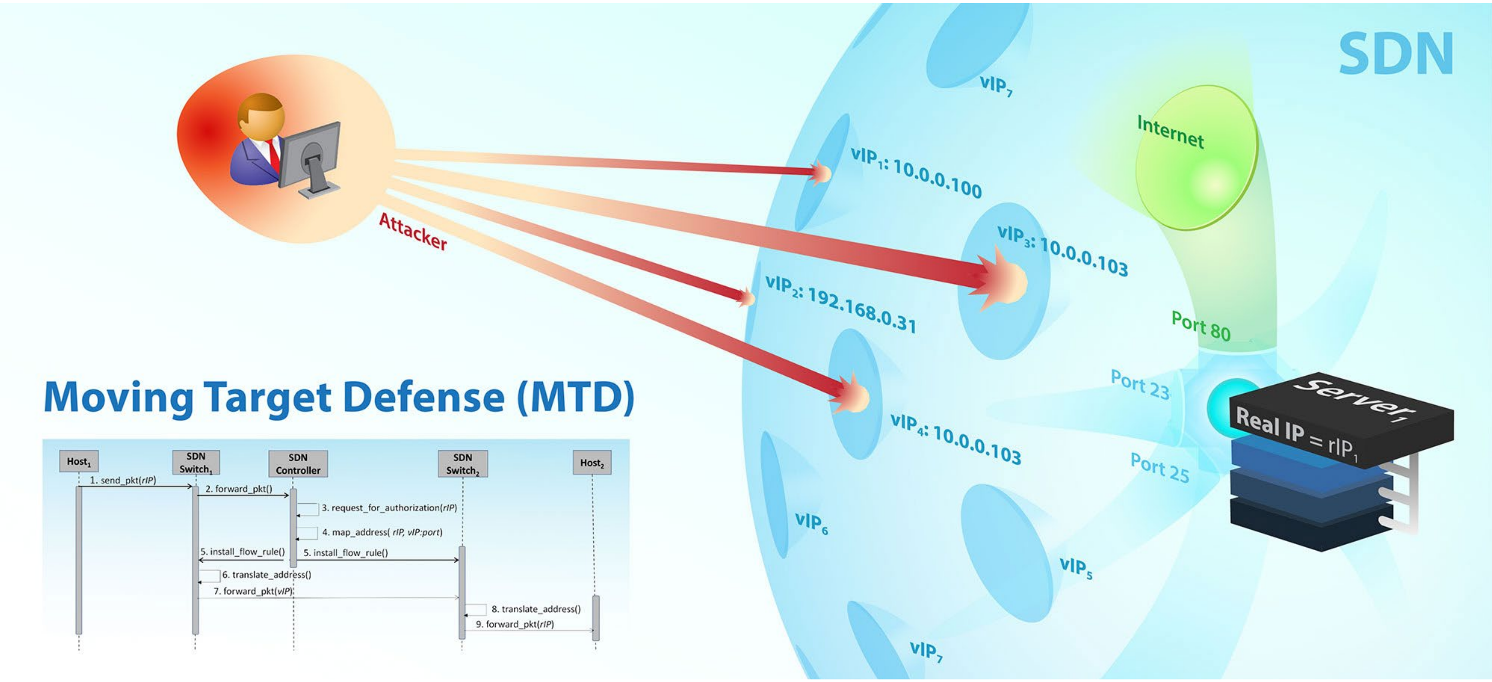
Protect Mission Systems



Concern: High consequence systems are becoming an attractive target for nation-state adversaries

Moving Target Defense

- Dynamic reconfiguration of environment
- Randomly change node address after n messages
- Mitigates risk of an attacker guessing the correct addresses and injecting data



Hypothesis: MTD increases cyber resilience

Research Goals

Hypothesis: integration of MTD with a **real-time protocol can increase cyber resilience** of platforms using the protocol

Key Research Questions:

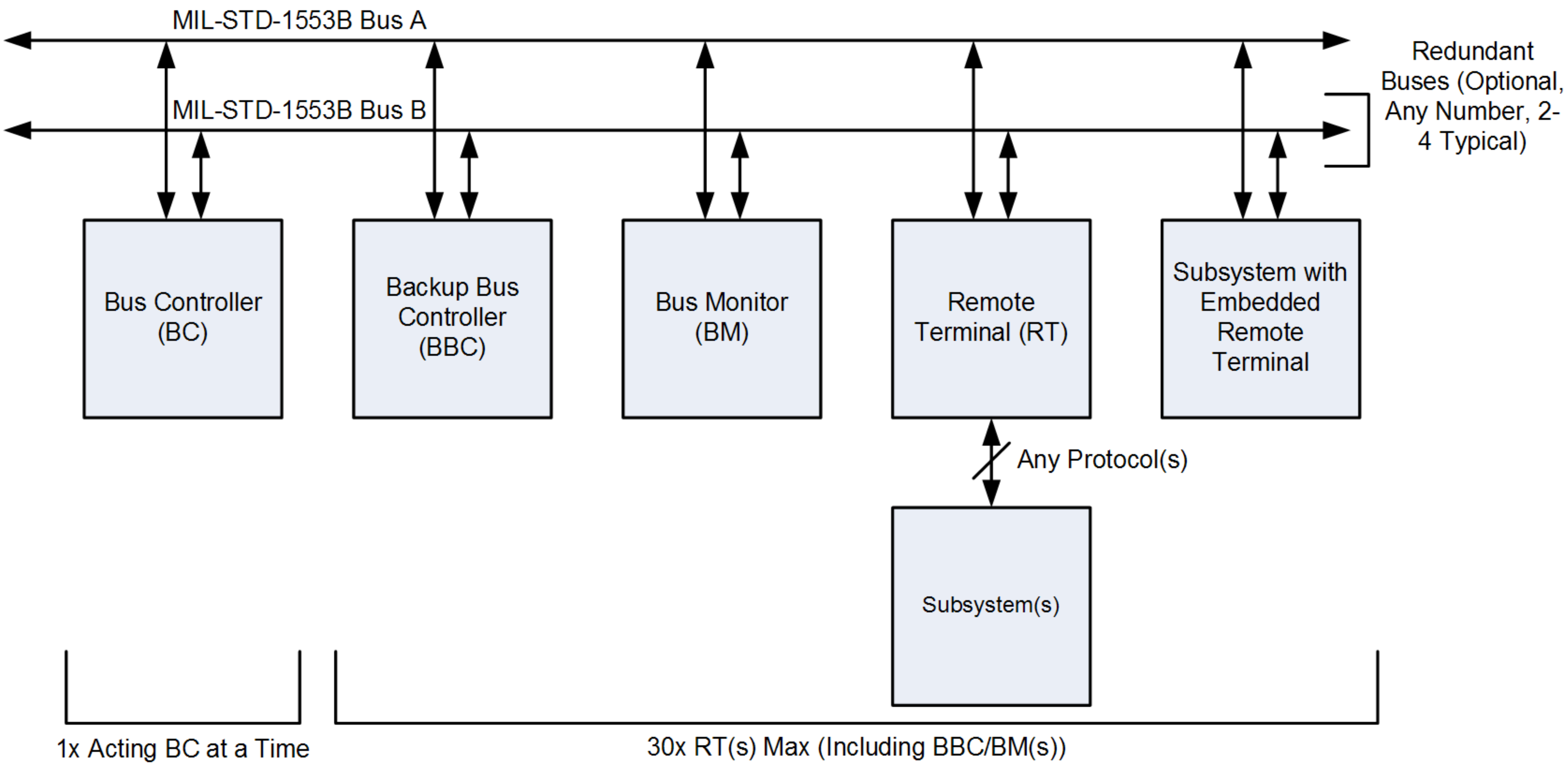
1. Can MTD be implemented in a manner that maintains operational constraints (e.g., accuracy, latency)?
2. Can we provide quantitative evidence that MTD does indeed improve cyber resilience?

Uniqueness: Real-time, SWaP constrained systems

Uniqueness: Doesn’t require anomaly detection

MTD Algorithm

MIL-STD-1553 Bus Architecture



Typical BC-RT Message


```
MESSAGE #2 --------------------------------------
(89)17:22:30.074.576.560   IM Gap: 15154.6us
BUS A - CMD:1820 (3-R-1-32) BCRT
9EF3 1D47 FB99 A306 5A18 CAC8 BFEE 607D
FB0C ED27 27C5 04A9 22CE 6B31 A22A C0AF
9C44 15FA 086B D872 0E5F 187E 08BC CFB1
92EB 038A 7152 1D6D 388F 6EF6 4325 C051
Rsp Time 7.7us STS:1800
Message Time = 685.7us
```

Design Challenges

Keep underlying protocol – determinism, predictability, reliability, and real-time operation

Dynamic address generation – each node must index or use a disjoint set of addresses as compared to other nodes on the network. Also, have the ability to increase or decrease speed of address hopping

Synchronization – provide fast recovery if a device loses sync

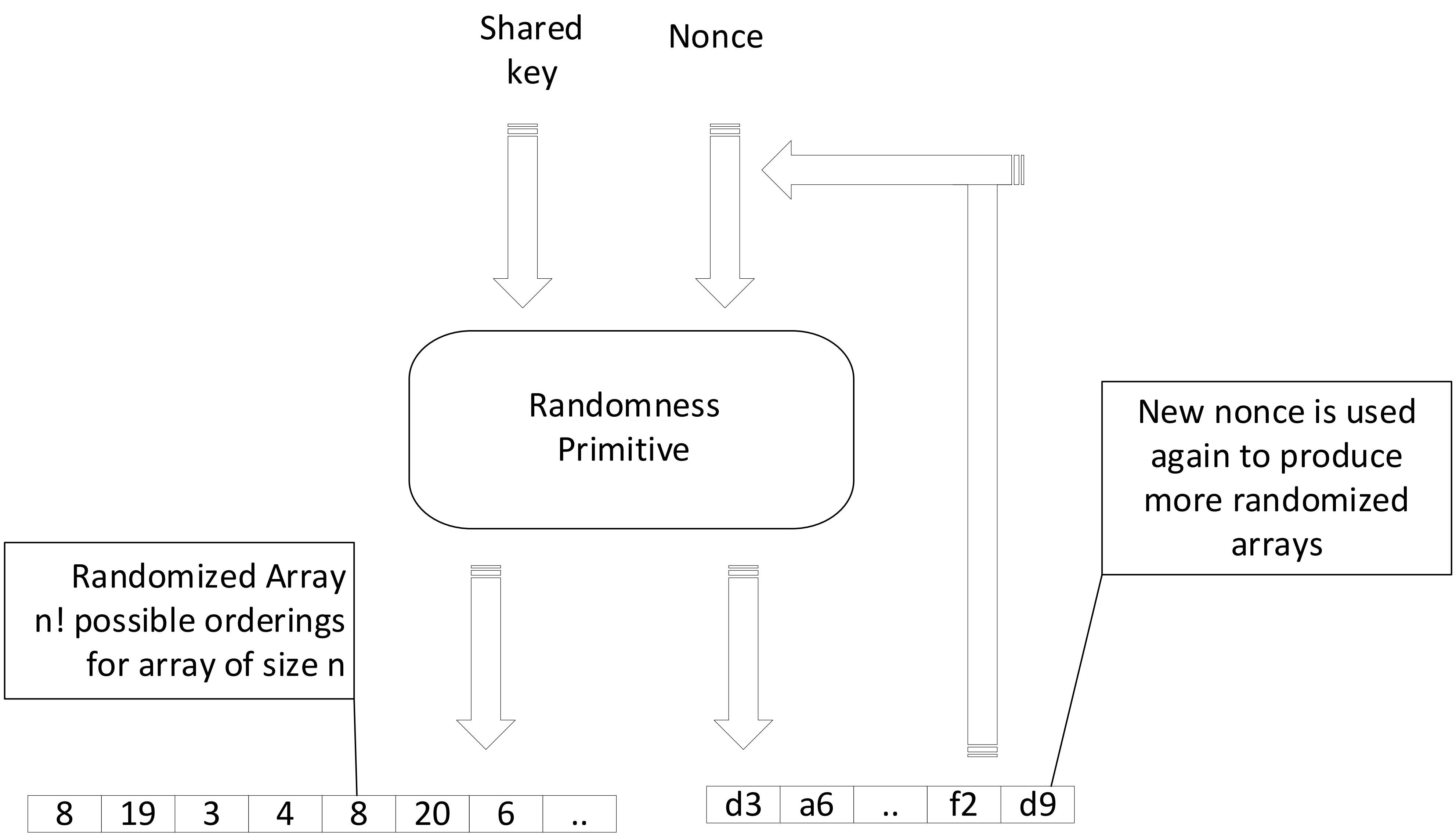
Entropy – provide enough randomness

Periodicity – provide sufficiently long hopping patterns

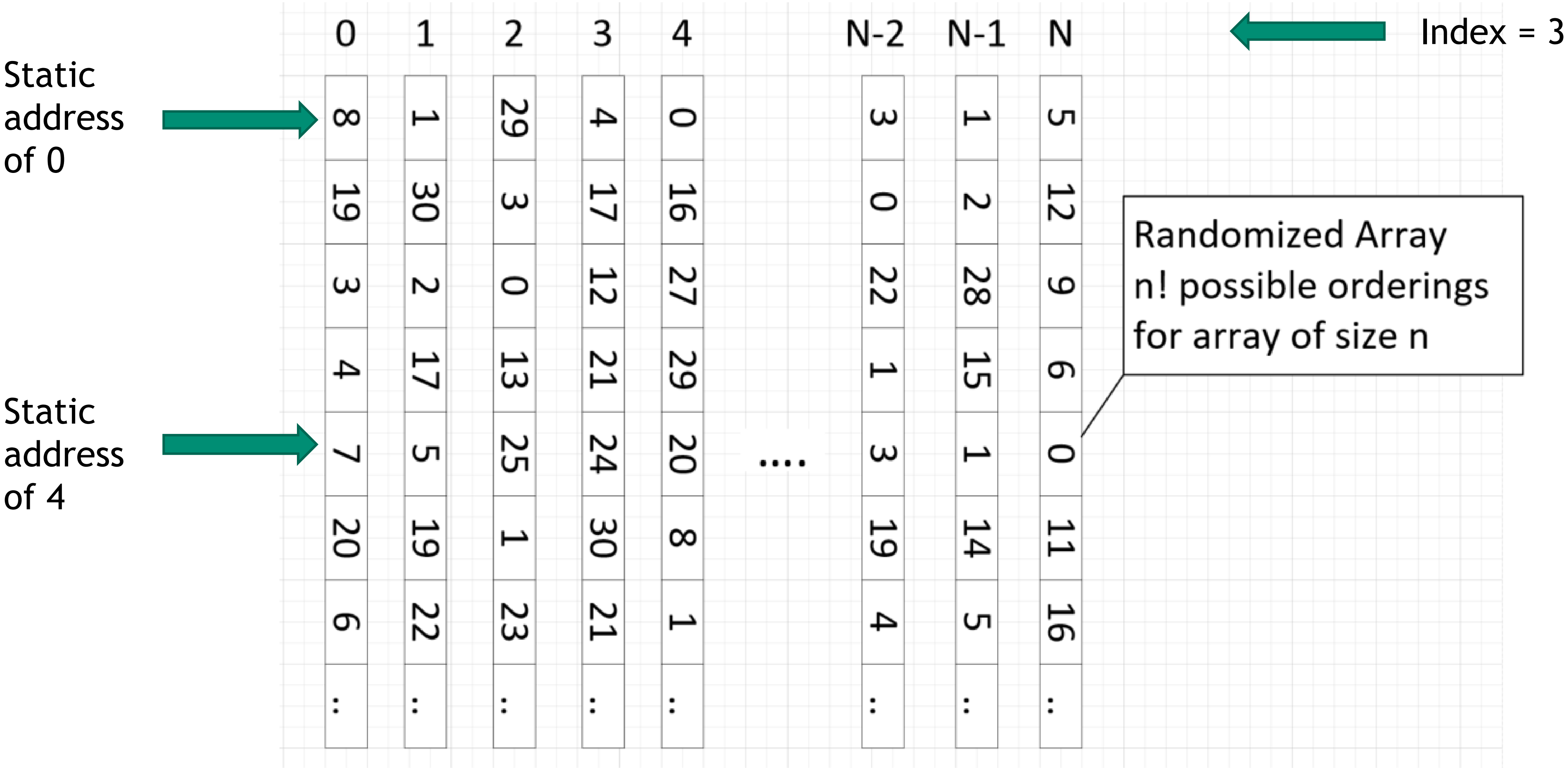
Authenticity – determine if MTD commands are authentic using analog signatures, MACs, MICs, etc.

Today: Describe a patent-pending, novel MTD algorithm for use on MIL-STD-1553

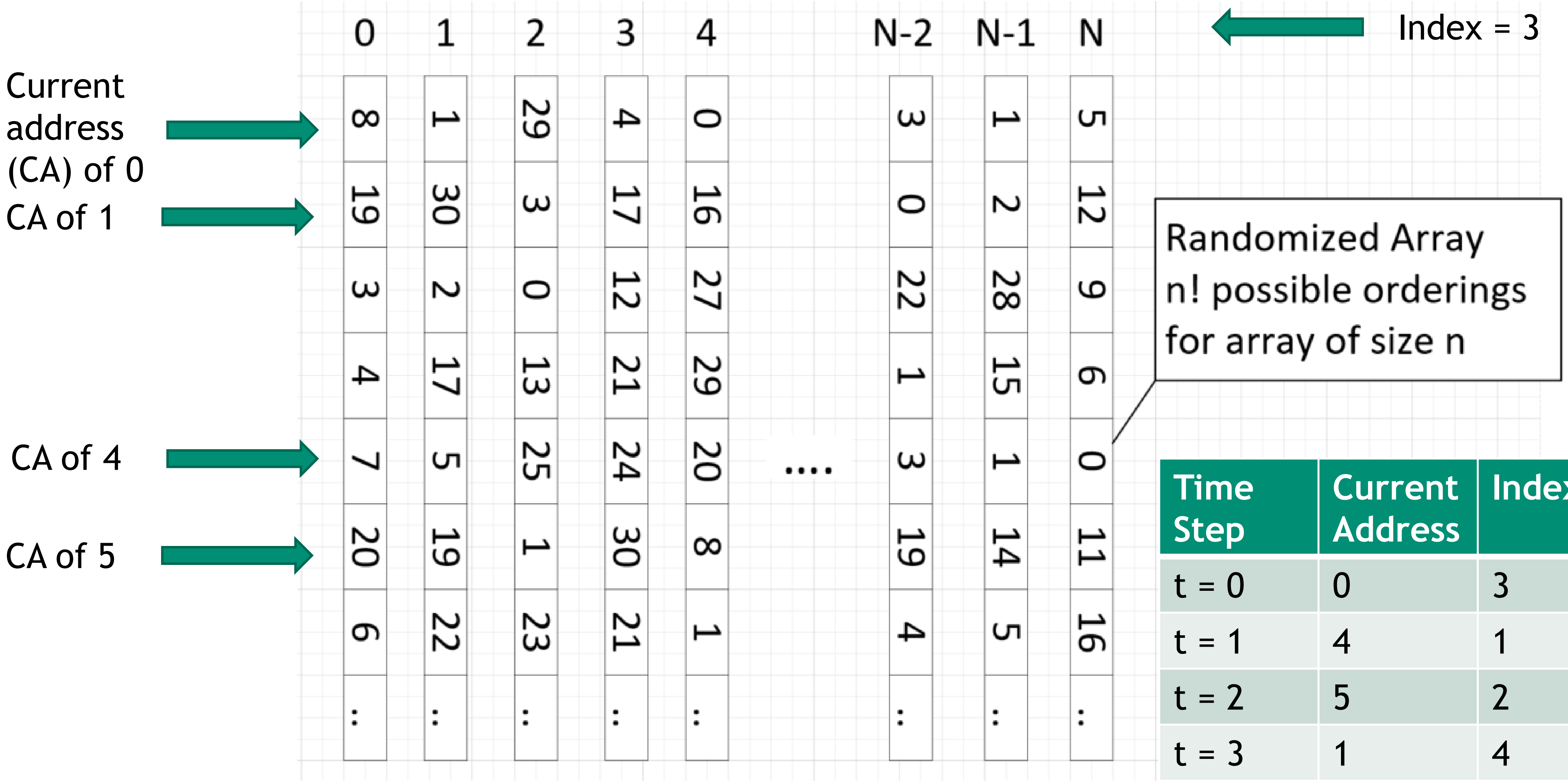
MTD Algorithm



State Matrix (Arrays) – Static Offset

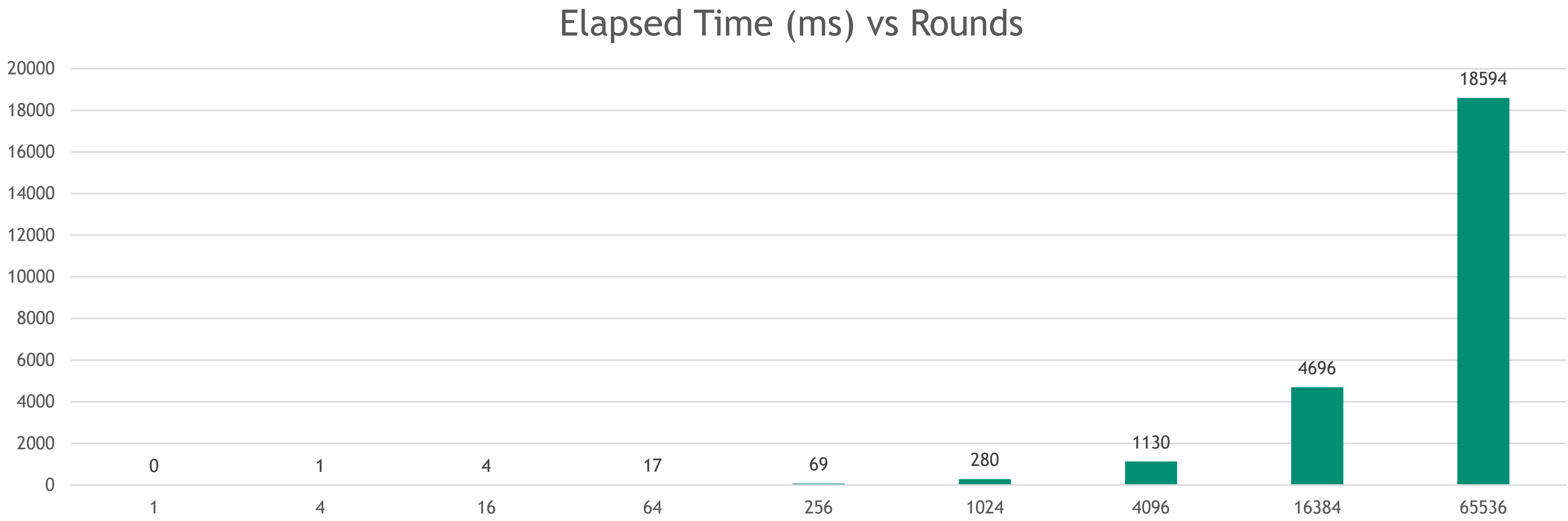


State Matrix (Arrays) – Current Offset

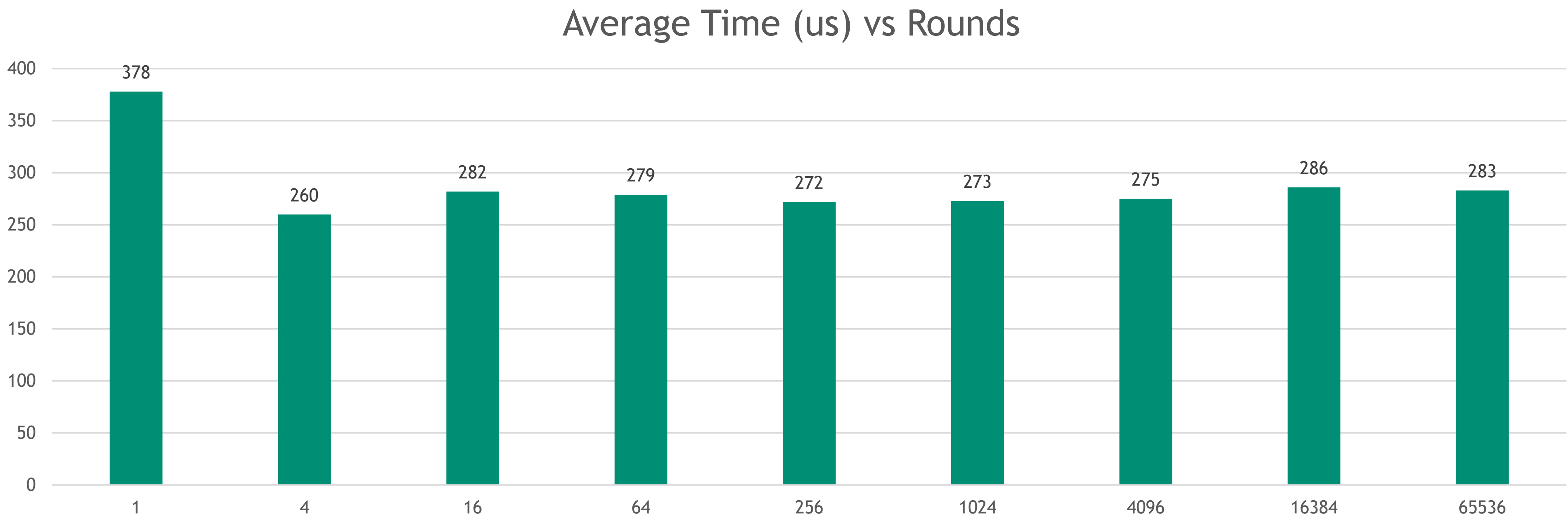


Time Step	Current Address	Index	Next Address
t = 0	0	3	**4**
t = 1	4	1	**5**
t = 2	5	2	**1**
t = 3	1	4	**16**

State Generation Performance (Elapsed Time)



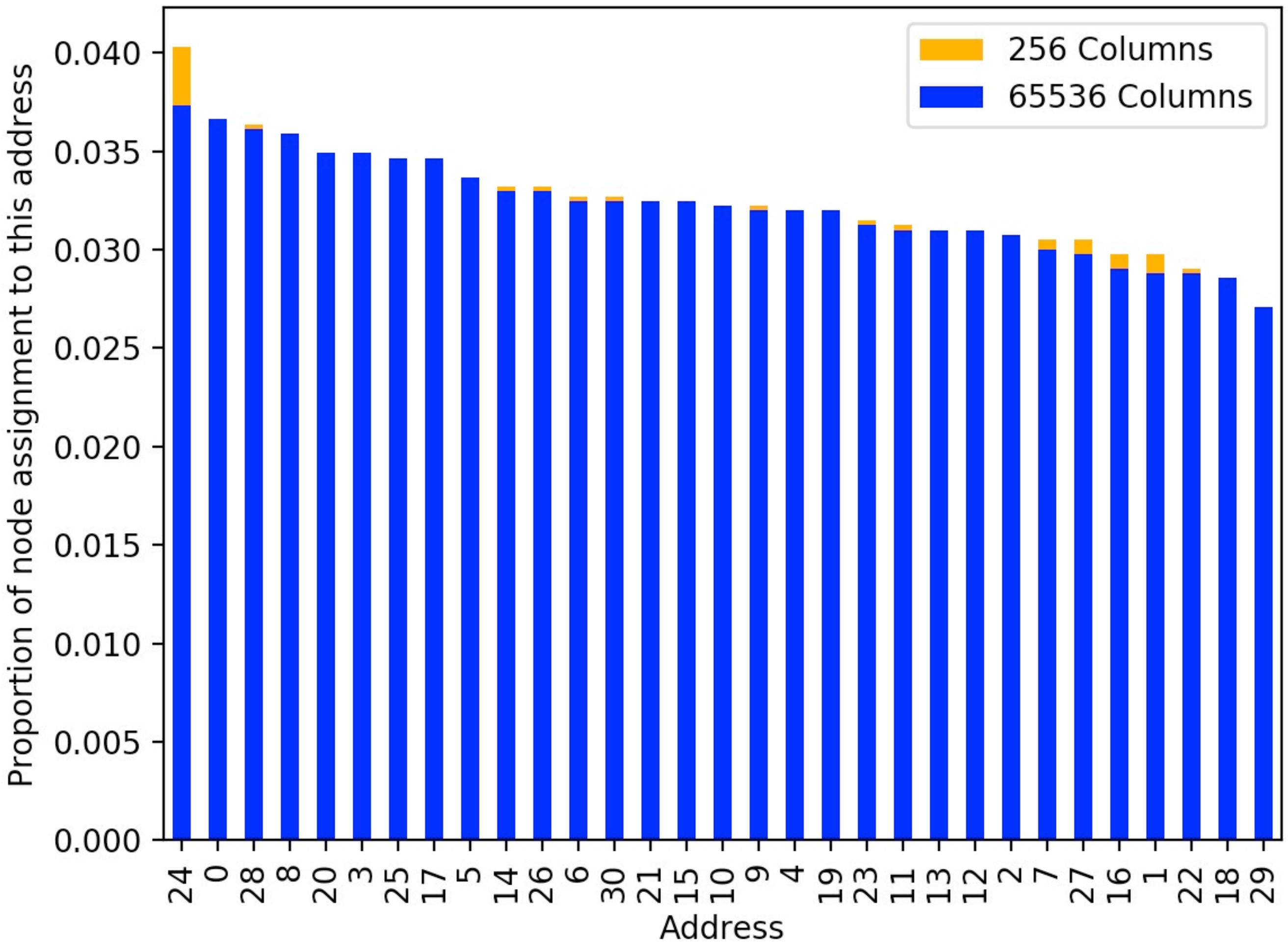
State Generation Performance (Average Time)



Space Requirements for State Matrix

Number of Arrays	Approximate Size (KB = 1024)
1	0.03125
4	0.125
16	0.5
64	2
256	8
1024	32
4096	128
16384	512
65536	2048

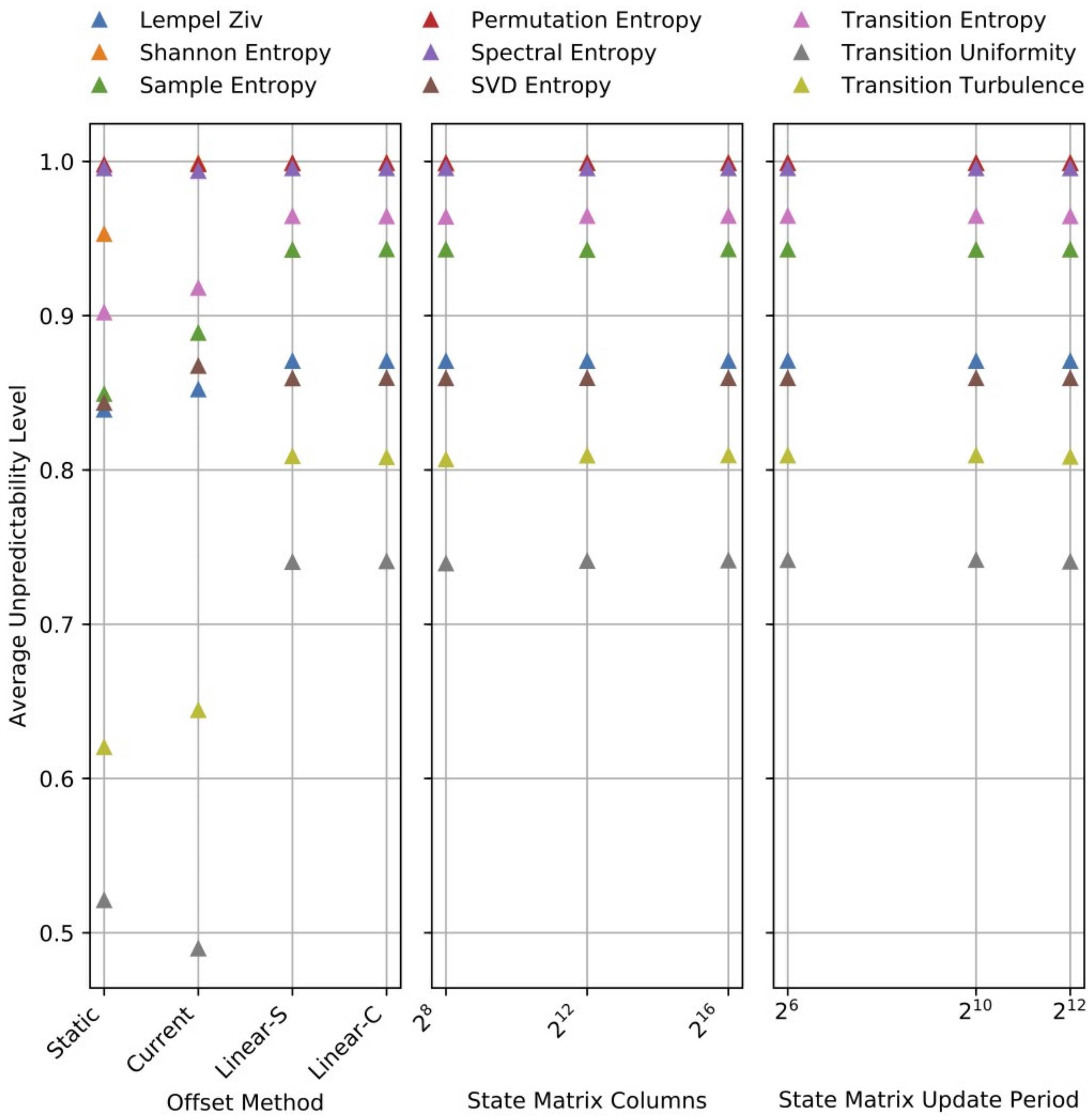
Entropy Results



Preliminary findings

- Frequency of addresses is not perfectly uniform, leaving some area for improvement
- Entropy for 256 columns is 0.9984
- Entropy for 65536 columns is 0.9989

Unpredictability Results



Analysis Process

1. Create 10 state matrices with 10 PRNG seeds
2. For each matrix, 31 address sequences (one for each node) for each of the 12 unique combinations of offset and matrix size (3,720 sequences)
3. Each sequence has a length of 4,096
4. Calculate set of 9 unpredictability metrics and average over 31 address sequences per state matrix and unique combination

- For update period, concatenated multiple matrix sequences to simulate state matrix updates

Preliminary findings Period

- Offset method has most effect on unpredictability metrics
- Number of state matrix columns and update period do not appear to significantly affect unpredictability

Experimentation

MIL-STD-1553 Research Plan

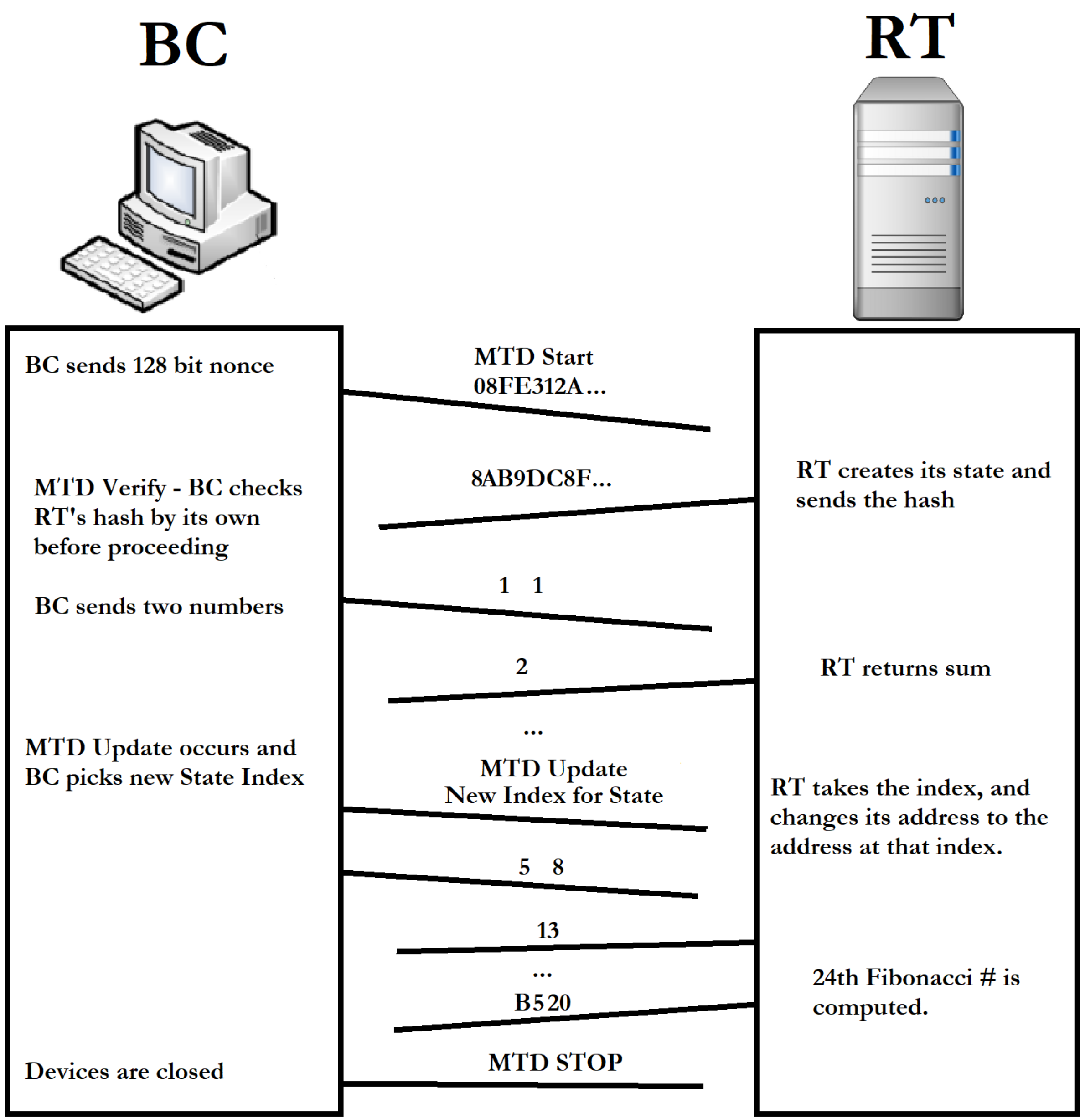
Phase 1: Calculate Fibonacci sequence w/ and w/o MTD

- BC sends 2 DWs which represent the previous 2 numbers in the sequence
- RT consumes the 2 DWs, computes the next number and responds (when requested) by the BC with a single DW containing the new number
- Run experiment to obtain the 24^{th} Fibonacci number
- Run experiment with MTD and update the address after every X frames (2 messages per frame)

Phase 2: Exfiltration

- Exfiltration data from target node on MIL-STD-1553 network
- Goal: Quantify reduction in adversarial knowledge

Data Flow


BC
RT
BC sends 128 bit nonce
MTD Start
08FE312A…
RT creates its state and sends the hash
8AB9DC8F…
MTD Verify - BC checks RT's hash by its own before proceeding
BC sends two numbers
1 1
2
RT returns sum
…
MTD Update occurs and BC picks new State Index
MTD Update
New Index for State
RT takes the index, and changes its address to the address at that index.
5 8
13
…
B520
24th Fibonacci # is computed.
MTD STOP
Devices are closed

MTD Commands

MTD Start

```
MESSAGE #1 ------------------------------------------
Time: [2019](218)14:19:50.152.086.960   IM Gap: 49202281us
BUS A - CMD:F828 (31-R-1-8) BRDCST BCRT
0000 0026 0000 1E27 0000 52F6 0000 0985
Message Time = 180us
```

BC's 128-bit Nonce

MTD Verify

```
MESSAGE #2 ------------------------------------------
Time: [2019](218)14:19:50.307.529.360   IM Gap: 155264.5us
BUS A - CMD:0CC0 (1-T-6-32) RTBC
Rsp Time 6.5us STS:0800
D541 934B 0B33 FF99 FFA8 3C5B FF94 FF9F
0106 6A45 FF81 E84A FF95 FF8A 385E 4F40
214A 055E FFAA 3364 FFCA FFD0 FFF0 2D78
FFCB E257 FFCC D378 DA40 FF91 930E FF8C
Message Time = 684.5us
```

Hash of State

MTD Update

```
MESSAGE #15 ------------------------------------------
Time: [2019](218)14:19:50.385.049.880   IM Gap: 13739.9us
BUS A - CMD:F841 (31-R-2-1) BRDCST BCRT
0017
Message Time = 40us
```

New Index for RT's State

Finished

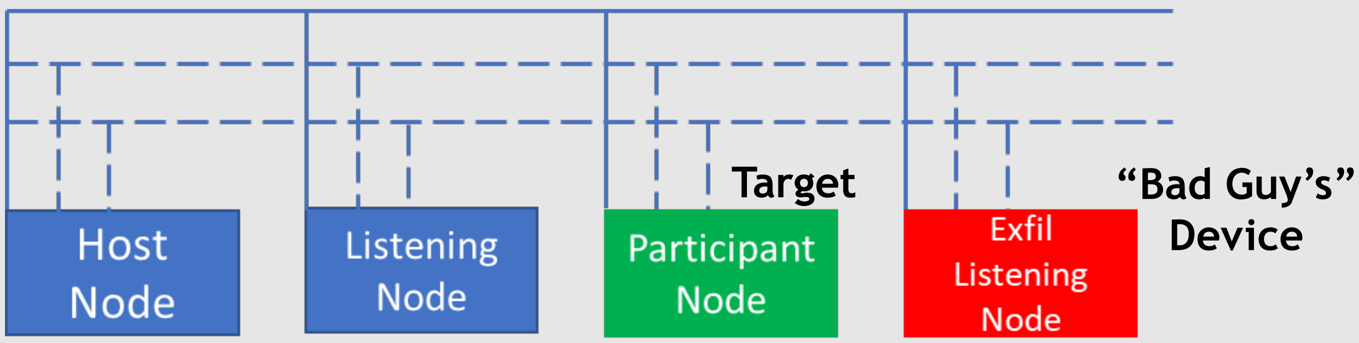
```
MESSAGE #117 ------------------------------------------
Time: [2019](218)14:19:51.019.122.240   IM Gap: 9041.6us
BUS A - CMD:E882 (29-R-4-2) BCRT
452F 6FF1
Rsp Time 6.5us STS:E800
Message Time = 84.5us

MESSAGE #118 ------------------------------------------
Time: [2019](218)14:19:51.020.207.240   IM Gap: 1002.6us
BUS A - CMD:ECA1 (29-T-5-1) RTBC
Rsp Time 6.5us STS:E800
B520
Message Time = 64.5us
```

24th Fibonacci #

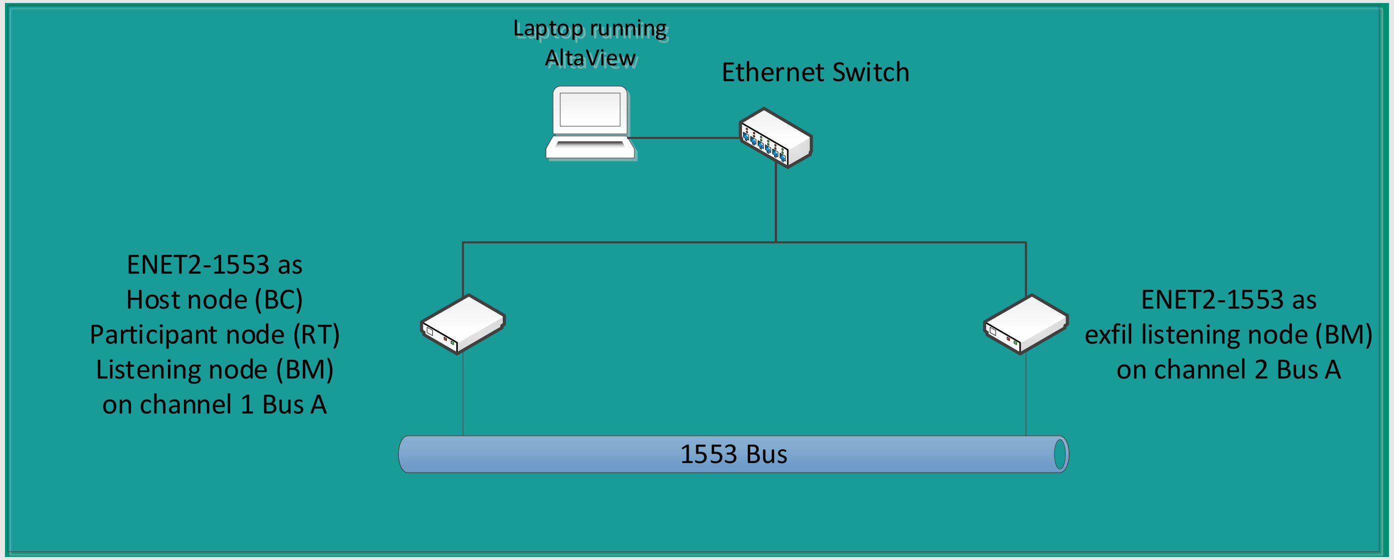
MTD Command Trace

```
MESSAGE #13 ---------------------------------------------
Time: [2019](218)14:19:50.370.162.640   IM Gap: 7407.6us
BUS A - CMD:0882 (1-R-4-2) BCRT
0002 0003
Rsp Time 6.5us STS:0800
Message Time = 84.5us

MESSAGE #14 ---------------------------------------------
Time: [2019](218)14:19:50.371.247.640   IM Gap: 1002.6us
BUS A - CMD:0CA1 (1-T-5-1) RTBC
Rsp Time 6.5us STS:0800
0005
Message Time = 64.5us

MESSAGE #15 ---------------------------------------------
Time: [2019](218)14:19:50.385.049.880   IM Gap: 13739.9us
BUS A - CMD:F841 (31-R-2-1) BRDCST BCRT
0017
Message Time = 40us

MESSAGE #16 ---------------------------------------------
Time: [2019](218)14:19:50.397.934.480   IM Gap: 12846.7us
BUS A - CMD:D882 (27-R-4-2) BCRT
0003 0005
Rsp Time 6.5us STS:D800
Message Time = 84.5us

MESSAGE #17 ---------------------------------------------
Time: [2019](218)14:19:50.399.019.440   IM Gap: 1002.6us
BUS A - CMD:DCA1 (27-T-5-1) RTBC
Rsp Time 6.5us STS:D800
0000
Message Time = 64.5us
```

Resilience Expt.: Exfiltration Attack Scenario



System of Interest



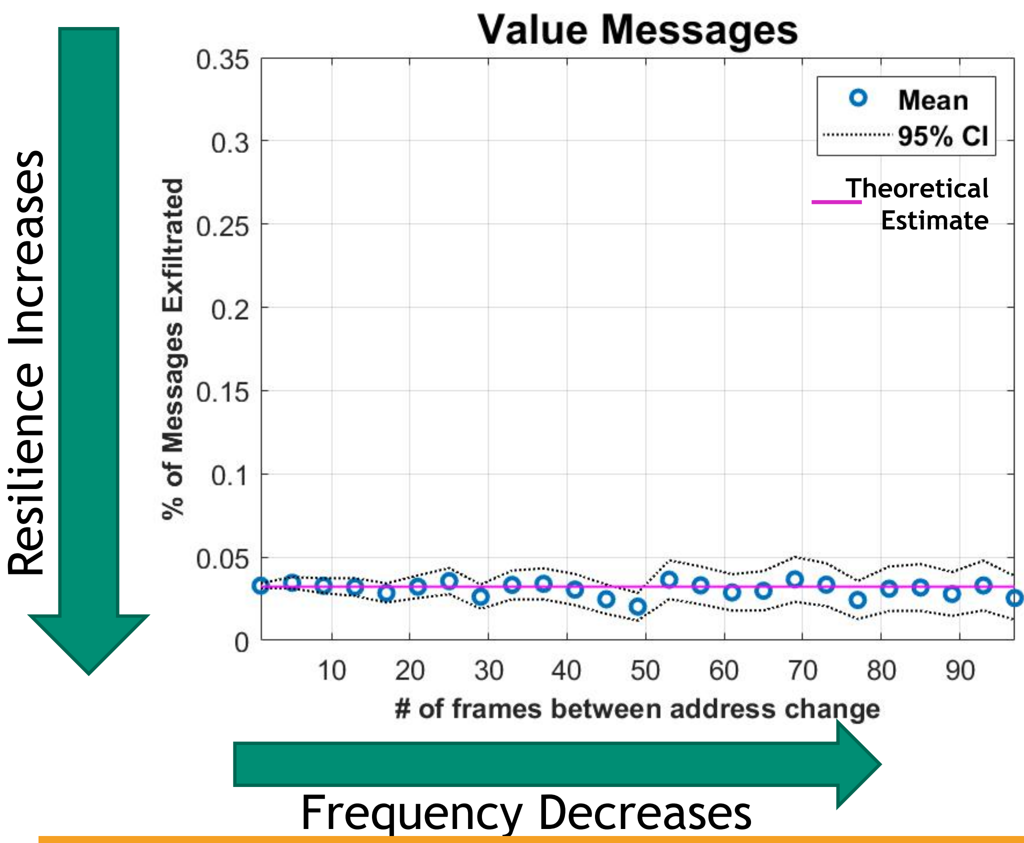
Experimental Set Up

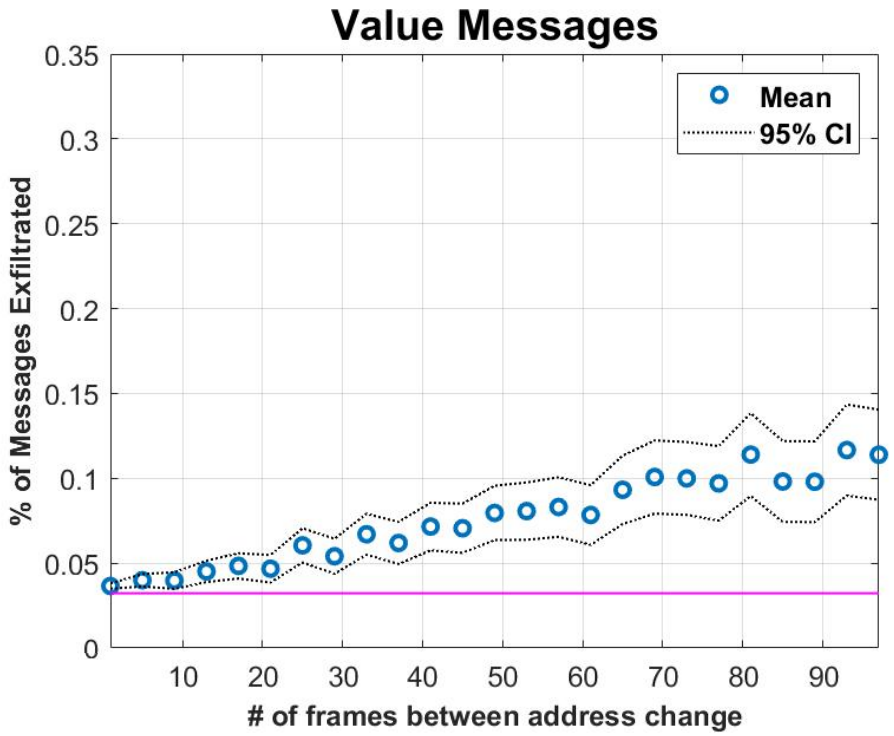
Set Up

- Attacker has corrupted an node to be an exfil listening node (red)
- Messages to/from target participant node (green) = messages of value to the attacker
- Exfil listening node monitors & exfils all messages to/from target
- With no MTD, exfil listening node will see and exfil 100% of messages to/from target

Question: does the implementation of MTD reduce the fraction of “messages of value” that are exfiltrated?

Exfil Expt.: Results





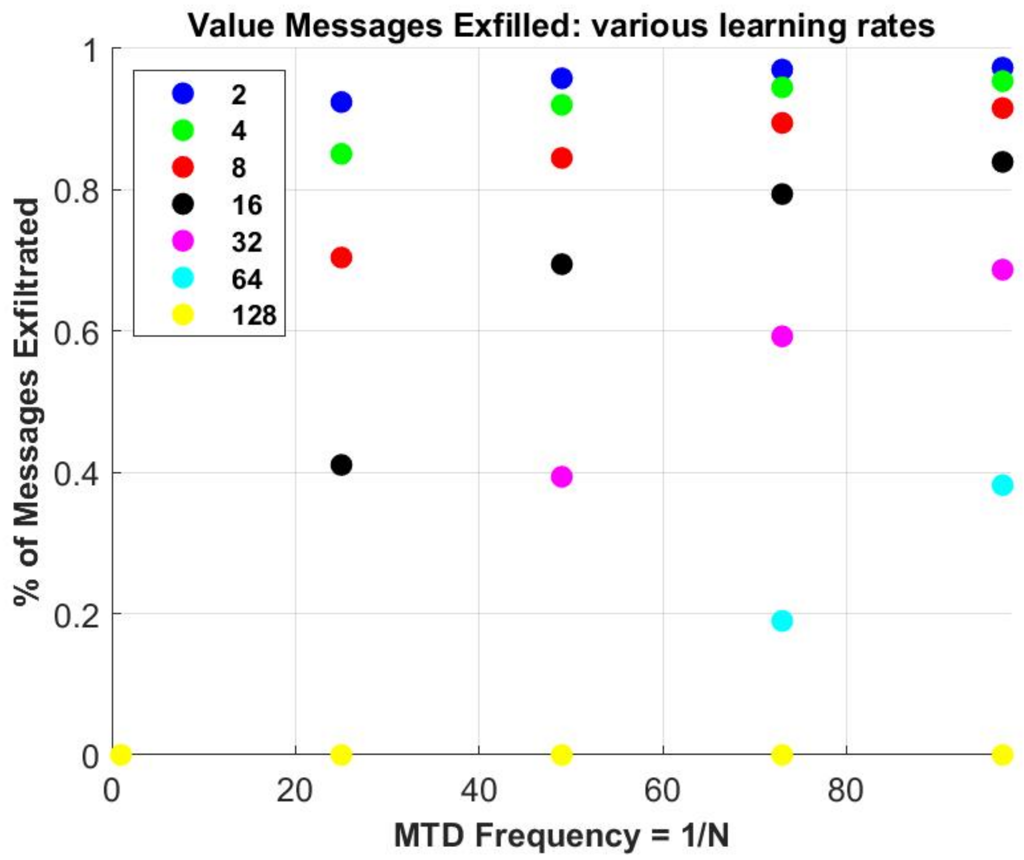
In this scenario

- MTD reduces % of value messages exfiltrated by ~97%
- Experimental results match theoretical estimates

When the adversary knows the starting address for the target

- Low frequencies give poorer results in the expts
- This observation is due to relatively short length of experiments (50 generations)
- When the length is increased, the expected # of messages exfiltrated decrease closer to 3%

Exfil Expt. Results: Learning Adversary



1000 Fibonacci Generations, 25 trials

Assume adversary learns new address after X frames

Example:
- Freq = 25, learned = 8 frames, exfil = 70%
- Freq = 25, learned = 16 frames, exfil = 40%
- Freq = 25, learned = 32, exfil = 0%

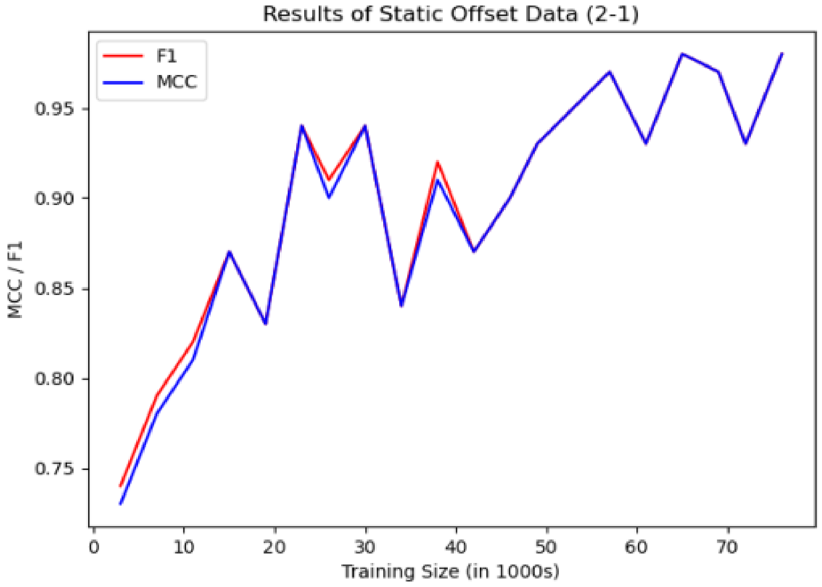
Takeaways:
- Against a learning adversary, MTD frequency needs to be faster than adversary learning rate to significantly mitigate exfil attacks
- These data can start informing design requirements

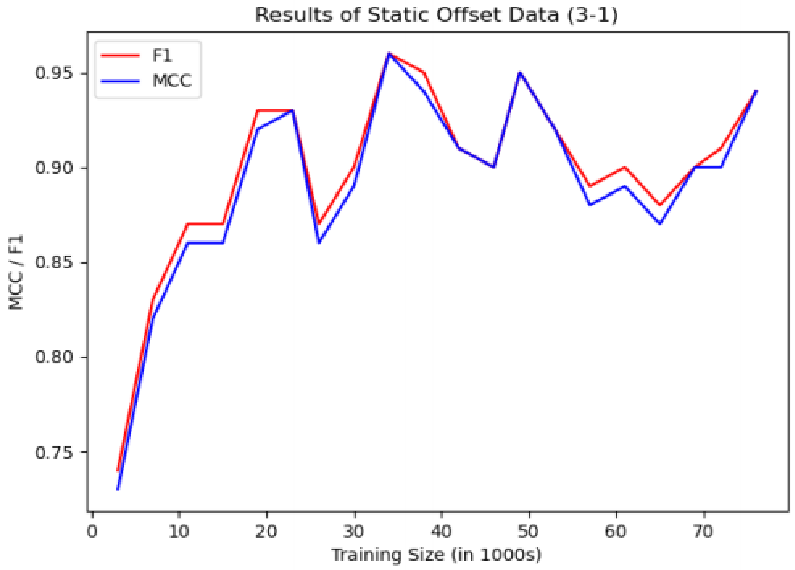
Machine Learning

ML Experimentation

- Given a log of all messages on the bus
  - ~~Can you figure out the state matrix?~~
  - Can you identify MTD messages?
  - **Can you determine the next address?**
  - ~~Are any other side channels present?~~
- Models Used
  - LSTM model for predicting the next address
  - Varied the number of previous addresses the model remembers
  - Training size varied
  - Test size always 20% of total data

ML Results (from Purdue University)





Future Work

Future Work & Ideas

Realized generic structure of algorithm: 4 components

IPv4 random port generator

- Existing SNL technology
- Randomized TCP port using MTD (ADDSec)
- This algorithm allowed different keys per packet as opposed to the same key for all packets
- NDA with commercial startup to integrated technology into their product

2FA Passphrase Generator

- Implement two-factor authentication system
- Doesn't not require typical infrastructure as other systems (e.g., RSA tokens, authenticators)
- Update state matrix to represent passphrases

Transparent filesystem

- Protection against ransomware
- Can't encrypt what you don't see
- Integrate MTD into filesystem with trigger to 'reveal' hidden files

Thank you!

Backup

Sandia's Impact

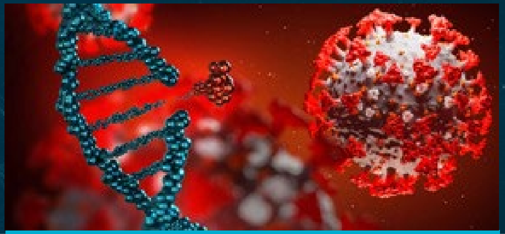
Sandia is often called upon to respond to high-profile events

Mars Perseverance rover

NASA's Perseverance rover landed safely on Mars after a seven-month journey through space. The event could only take place following a safe launch that had been vetted by Sandia scientists.
(Courtesy of NASA/JPL-Caltech)

Cleanroom invented 1963

As the birthplace of the modern cleanroom, Sandia helped revolutionize manufacturing in electronics and pharmaceuticals and advance space exploration.
$50 billion worth of cleanrooms built worldwide.



COVID-19 Pandemic

Sandia has more than 50 COVID-related science and engineering projects that are designed to help the nation during the pandemic.

(Image by Loren Stacks)

Sustainable Energy

Sandia seeks to support the creation of a secure energy future for the US by using its capabilities to enable an uninterrupted and enduring supply of energy from domestic sources, and to assure the reliability and resiliency of the associated energy infrastructure.

Fulfilling Our National Security Mission

Global Security

Nuclear Deterrence

National Security Programs

Energy & Homeland Security

Advanced Science & Technology

Some of the critical national security issues that we address lie in the cyber area.

MTD Update Command Indexing

Command Word Bit Usage

Remote Terminal address (0 - 31)					Receive or Transmit	Location (sub-address) of data (1 - 30)					Number of words to expect (1 - 32)				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Column Selection (10 bits)

Multiplier

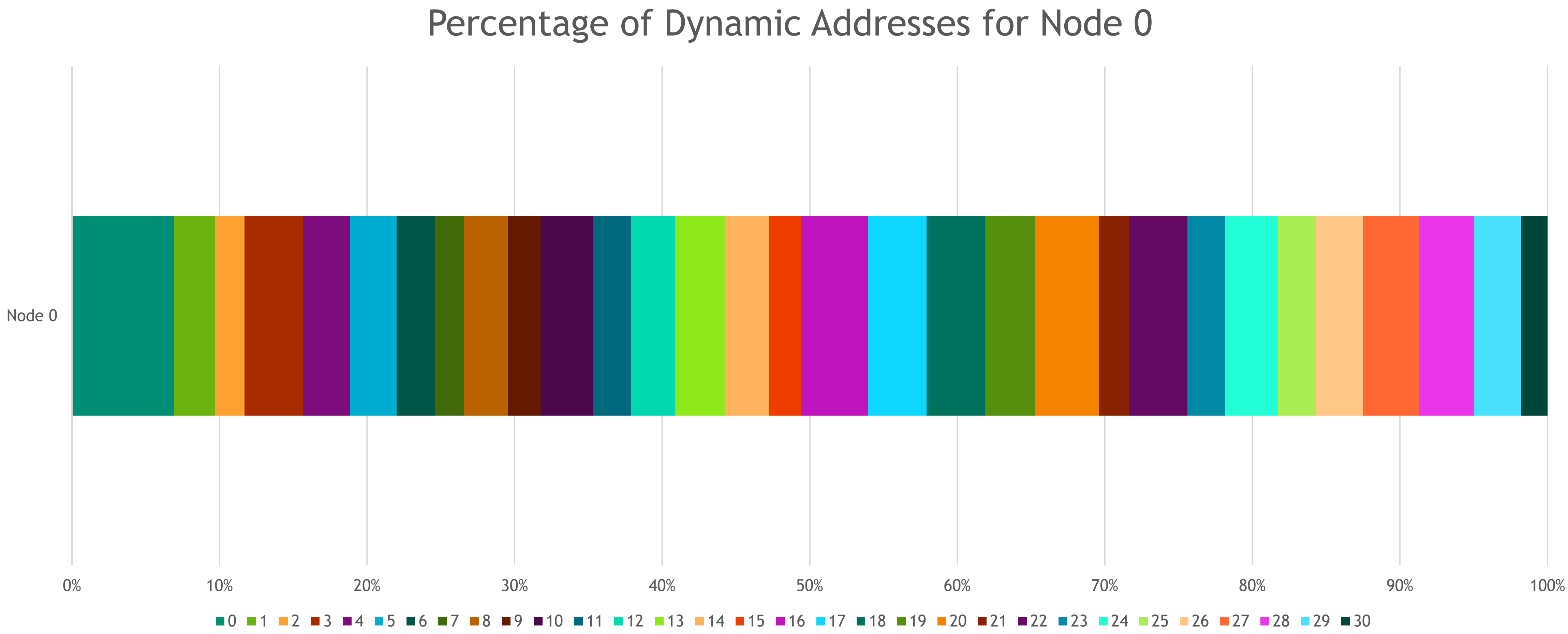
Adder

Command Word Bit Usage

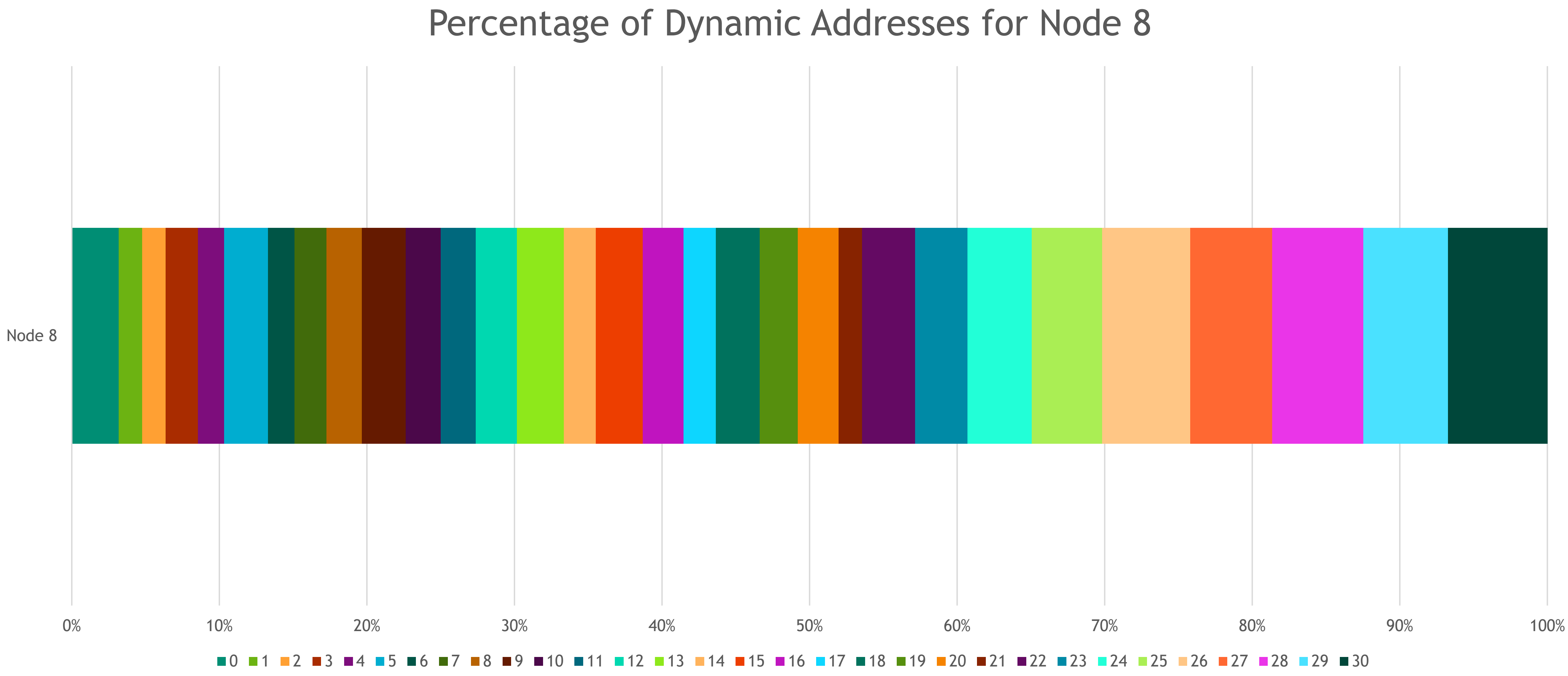
Remote Terminal address (0 - 31)					Receive or Transmit	Location (sub-address) of data (1 - 30)					Number of words to expect (1 - 32)				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Column Selection (16 bits)

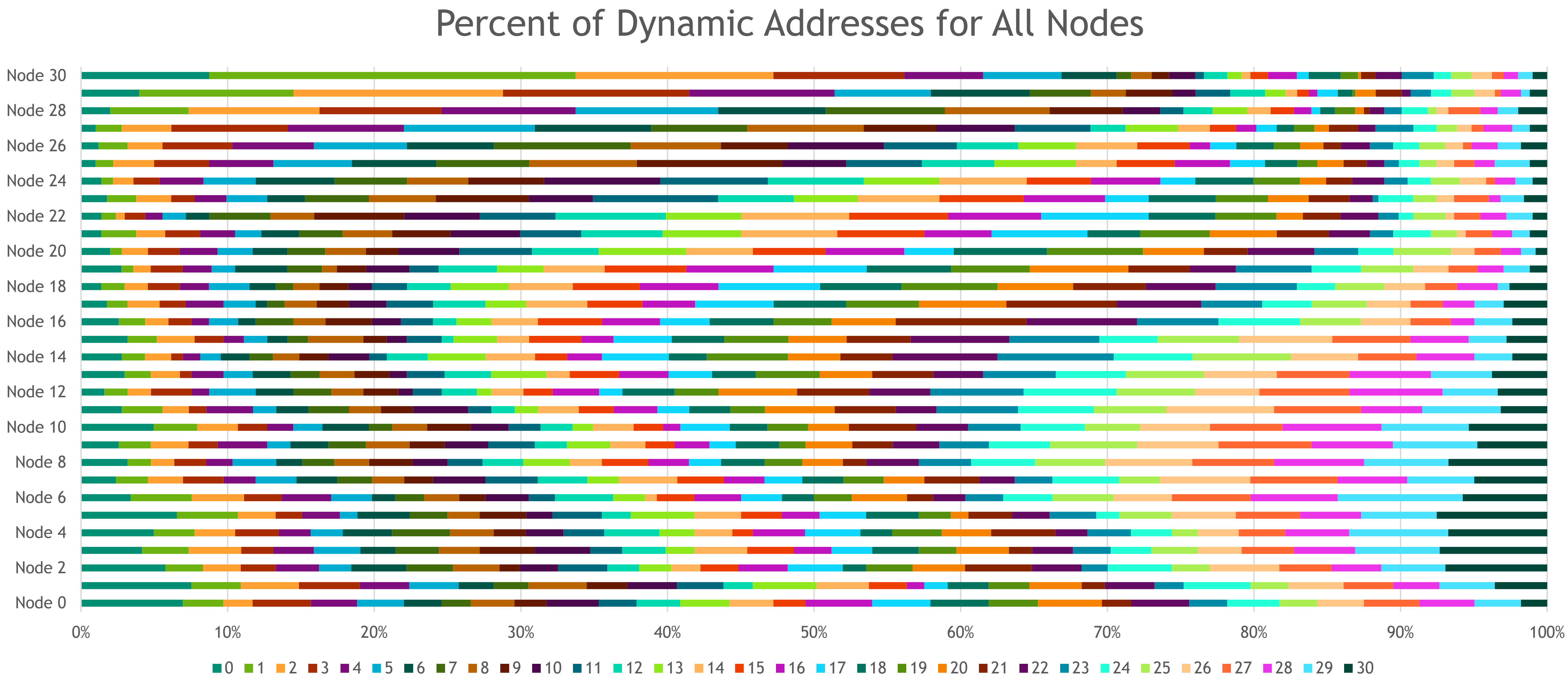
State Generation Randomness



State Generation Randomness



State Generation Randomness (All nodes)



State Array Performance (65536 rounds)

Random nonce generation ~ 10us

Hash generation ~ 500 ms

Hash verification ~ 500 ms

Access Performance

Execute state access ten million times

Generate random index

Access new address and store it

Use new address next time (current)

Average access time = 1 us

Time Required to see all possible addresses (Random Index)

Cell Offset Method	# of columns	Total	Average	Minimum	Maximum	Std. Dev
Static	512	7686	247.96	87	606	136.82
Current	512	4083	129.13	65	278	38.14
L-Static	512	5842	188.45	85	384	70.32
L-Current	512	3988	128.64	71	239	40.36
Static	8192	5934	191.42	86	369	77.55
Current	8192	3855	124.35	71	204	36.18
L-Static	8192	4272	137.81	87	270	37.20
L-Current	8192	4259	137.39	67	244	45.07
Static	65536	5501	177.45	75	351	74.52
Current	65536	3827	123.45	71	196	33.34
L-Static	65536	3756	121.16	61	246	36.77
L-Current	65536	3587	115.71	77	198	28.72

Offset Methods

Index – 16-bit index

Static – use static address as offset

Current – use current address as offset

Offset Selection Mechanism		Index Interpretation	
		Unsigned integer	**Linear combination**
Address Used	**Initial address**	Static	Linear-static
	Current address	Current	Linear-current

16-bit index: 10-bit (sub-)index, 3-bit multiplier, 3-bit adder

Linear static (Linear-S) – c is the static address

Linear current (Linear-C) – c is the current address

$4a+b+c \bmod n = d$, where a, b, c, d, and n are unsigned integers

Time Required to see all possible addresses (Sequential index)

Cell Offset Method	Total	Average	Minimum	Maximum	Std. Dev
Static	5842	188.452	85	384	70.322
Current	3903	125.90	73	237	34.08
L-Static	1798	58	58	58	0
L-Current	4262	137.48	85	250	36.28

Size of state matrix does not matter for sequential index

MTD Algorithm Summary

All addresses are not created equal (non-uniform distribution)

All addresses are used given enough time

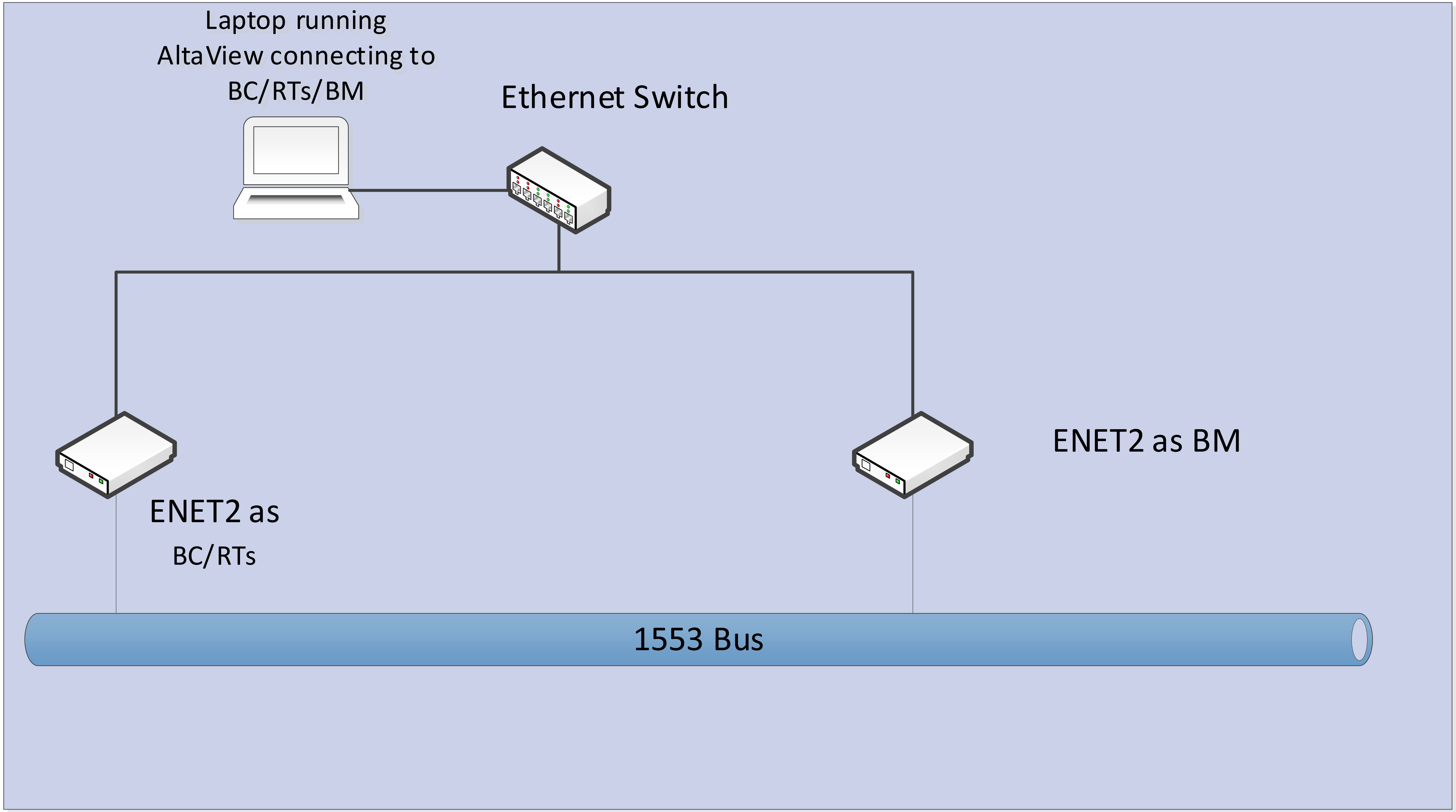
Try different primitive (AES, LFSR, RDRAND, etc.)

Don't need large matrix to have good entropy

Index into state (or states) (don't generate state array on-the-fly)

Non-address attributes or different size addresses may not have same profile

Testbed


Laptop running
AltaView connecting to
BC/RTs/BM
Ethernet Switch
ENET2 as BM
ENET2 as
BC/RTs
1553 Bus

ENET2


Alta dt
data technologies
eNet-1553
J1
1553 Ethernet/Power
Made in USA
J2
AUX I/O
PWR BIT
ETH LNK
ETH ACT
1553 ACT

MTD Commands

Start (send 128-bit nonce)
- 1 CW, 8 DWs
- 31-R-1-8

Verify (HMAC-512)
- 1CW, 32 DW
- RT-T-6-32

Update (frequency)
- 1 CW, 1DW
- 16-bit index
- 31-R-2-1

Stop (DW doesn't matter)
- 1CW, 1DW
- 31-R-3-1

- Add HMAC-256 to message for authenticity

Method #1
- 16-bit index
- 65536 orderings/arrays
- Use 16 bits to index
- Static address is the cell number

Method #2
- 10-bit column selection (i.e. sub-index)
- 3-bit multiplier (a)
- 3-bit adder (b)
- Static address or current address (c)
- I = 4a + b + c mod 31, I = cell number

MTD Update

```
MESSAGE #13 -------------------------------------------
Time: [2019](218)14:19:50.370.162.640   IM Gap: 7407.6us
BUS A - CMD:0882 (1-R-4-2) BCRT
0002 0003
Rsp Time 6.5us STS:0800
Message Time = 84.5us

MESSAGE #14 -------------------------------------------
Time: [2019](218)14:19:50.371.247.640   IM Gap: 1002.6us
BUS A - CMD:0CA1 (1-T-5-1) RTBC
Rsp Time 6.5us STS:0800
0005
Message Time = 64.5us

MESSAGE #15 -------------------------------------------
Time: [2019](218)14:19:50.385.049.880   IM Gap: 13739.9us
BUS A - CMD:F841 (31-R-2-1) BRDCST BCRT
0017
Message Time = 40us

MESSAGE #16 -------------------------------------------
Time: [2019](218)14:19:50.397.934.480   IM Gap: 12846.7us
BUS A - CMD:D882 (27-R-4-2) BCRT
0003 0005
Rsp Time 6.5us STS:D800
Message Time = 84.5us

MESSAGE #17 -------------------------------------------
Time: [2019](218)14:19:50.399.019.440   IM Gap: 1002.6us
BUS A - CMD:DCA1 (27-T-5-1) RTBC
Rsp Time 6.5us STS:D800
0000
Message Time = 64.5us
```

MTD Update (2)

```
MESSAGE #13 ---------------------------------------------
Time: [2019](218)14:19:50.370.162.640   IM Gap: 7407.6us
BUS A - CMD:0882 (1-R-4-2) BCRT
0002 0003
Rsp Time 6.5us STS:0800
Message Time = 84.5us

MESSAGE #14 ---------------------------------------------
Time: [2019](218)14:19:50.371.247.640   IM Gap: 1002.6us
BUS A - CMD:0CA1 (1-T-5-1) RTBC
Rsp Time 6.5us STS:0800
0005
Message Time = 64.5us

MESSAGE #15 ---------------------------------------------
Time: [2019](218)14:19:50.385.049.880   IM Gap: 13739.9us
BUS A - CMD:F841 (31-R-2-1) BRDCST BCRT
0017
Message Time = 40us

MESSAGE #16 ---------------------------------------------
Time: [2019](218)14:19:50.397.934.480   IM Gap: 12846.7us
BUS A - CMD:D882 (27-R-4-2) BCRT
0003 0005
Rsp Time 6.5us STS:D800
Message Time = 84.5us

MESSAGE #17 ---------------------------------------------
Time: [2019](218)14:19:50.399.019.440   IM Gap: 1002.6us
BUS A - CMD:DCA1 (27-T-5-1) RTBC
Rsp Time 6.5us STS:D800
0000
Message Time = 64.5us
```

Average Overhead

Change frequency of MTD update command
Increase delays so increase in message count is due only to MTD overhead
1000 runs (generations)

MTD Frequency (# of frames)	Predicted Overhead (%)	Actual Overhead (%)
1	50.0	50.1
2	25.0	25.1
3	16.7	16.7
5	10.0	10.2
10	5.0	5.0
20	2.5	2.6
50	1.0	1.0
100	0.5	0.5

ENET2-1553 Challenges

Flow control
- RT may not compute answer before BC requests answer

Latency
- Polling-based interrupts
- 200+ microseconds round trip time (some functionality takes multiple calls)
- Typical messages take less than 100 microseconds

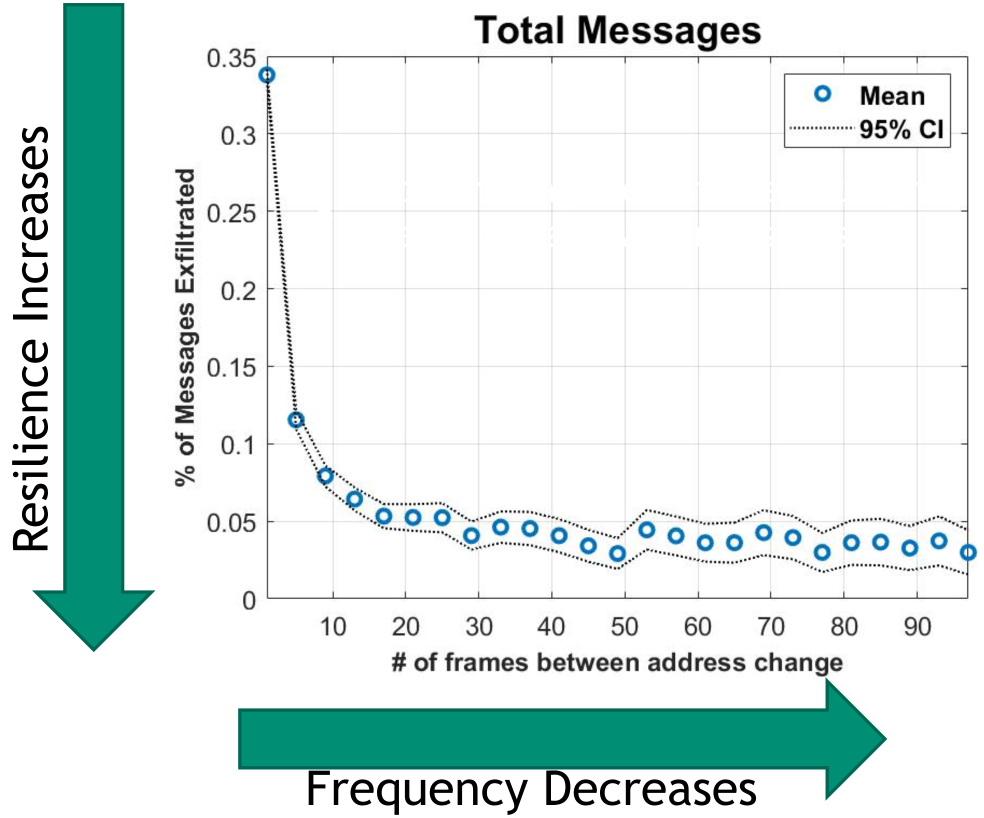
Non-empty buffers during 'wrap-around'
- Previous entries in buffers not cleared
- Can disrupt calculation by providing stale data

All above issues can be addressed with code

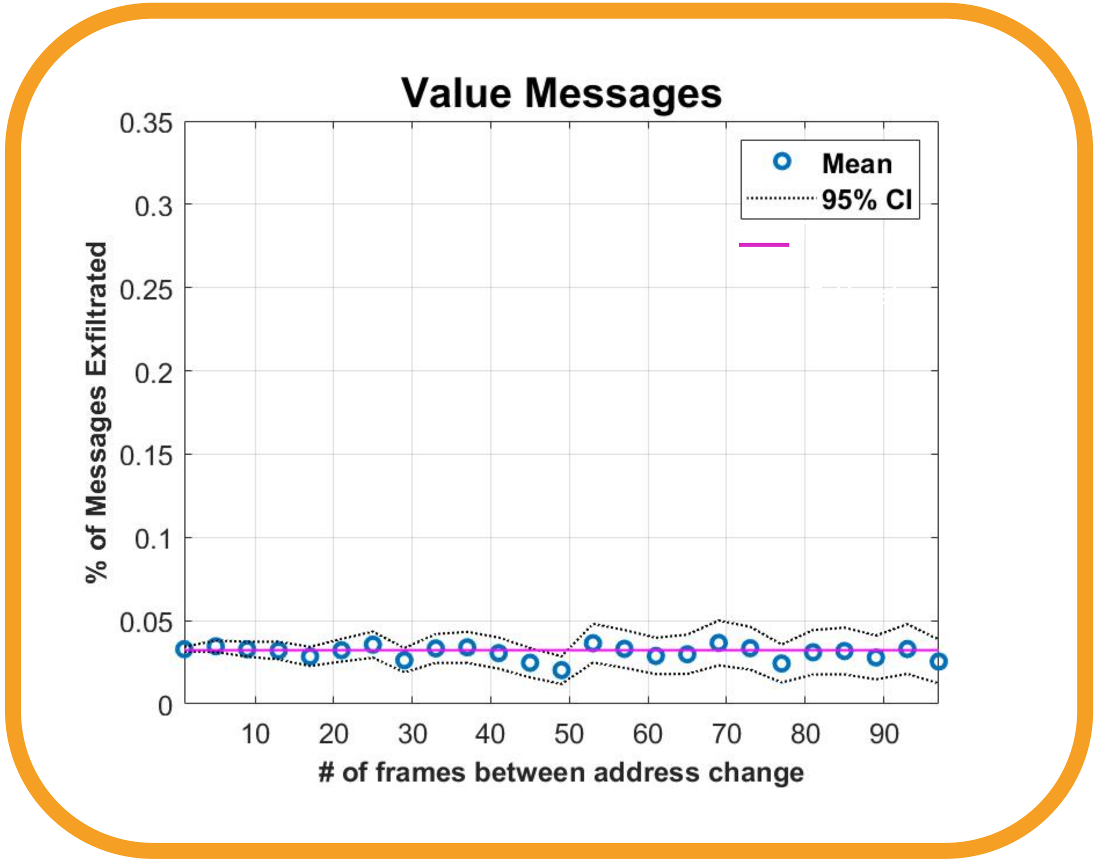
Potential bottleneck with BM, RT, and BC using same IP stack and queues (need to validate with vendor)

Solutions:
- Purchased PCIe card to reduce latency and push the MTD performance 'envelope'
- Use separate computers for functionality

Exfil Expt.: Results



KEY RESULT

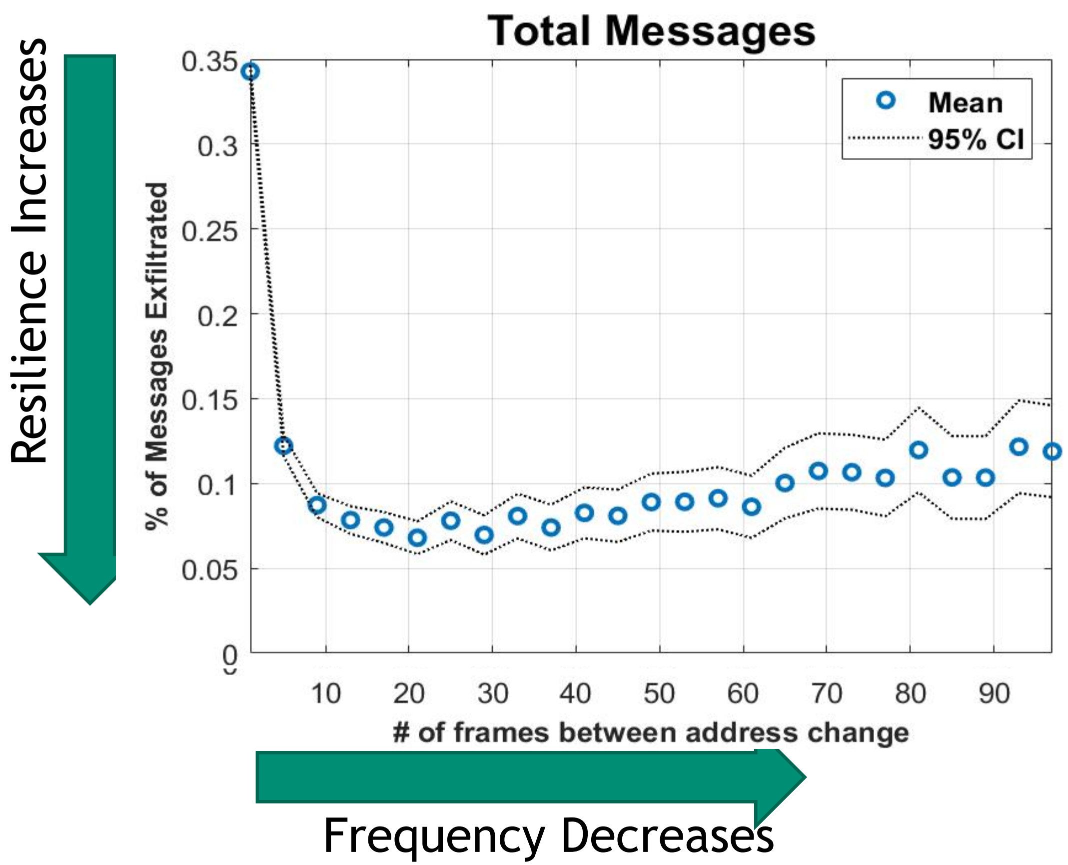


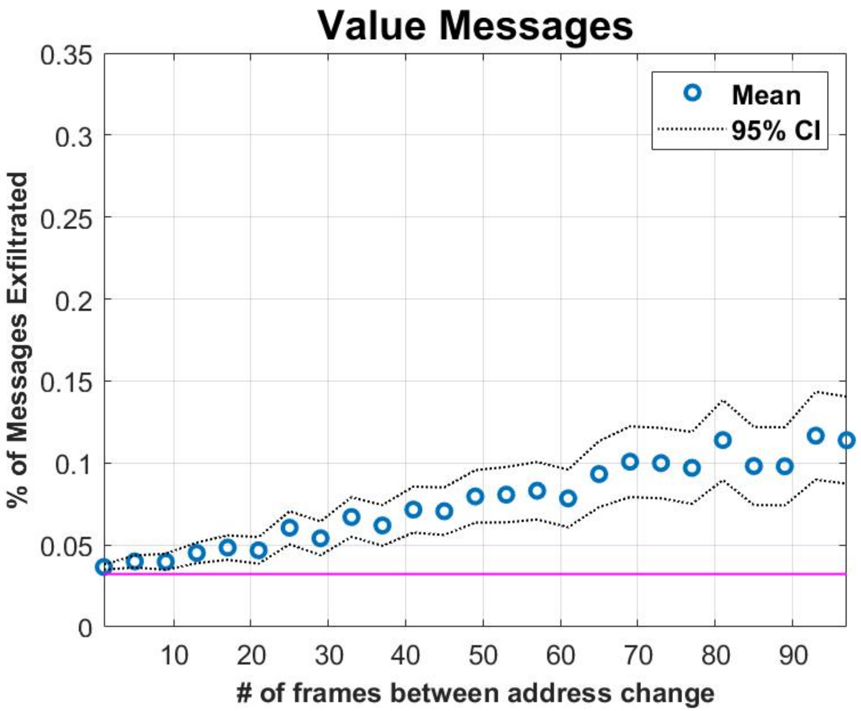
In this scenario

- MTD reduces % of value messages exfiltrated by ~97%
- Experimental results match theoretical estimates
- We can quantify how well MTD increases resilience

Exfil Expt. Results: Learning Adversary

Result when adversary knows the starting address





When the adversary knows the starting address for the target

- Low frequencies give poorer results in the expts
- This observation is due to relatively short length of experiments (50 generations)
- When the length is increased, the expected # of messages exfiltrated decrease closer to 3%