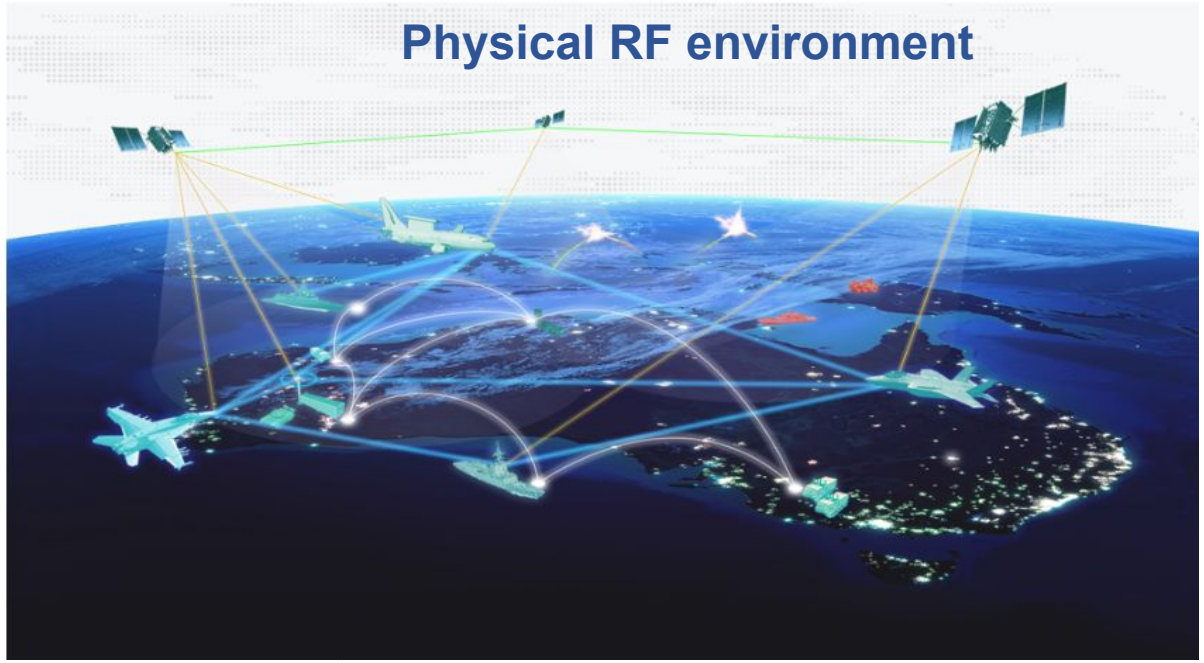
ARION: A Computing Platform for Real-time Emulation of Radio Frequency Interactions

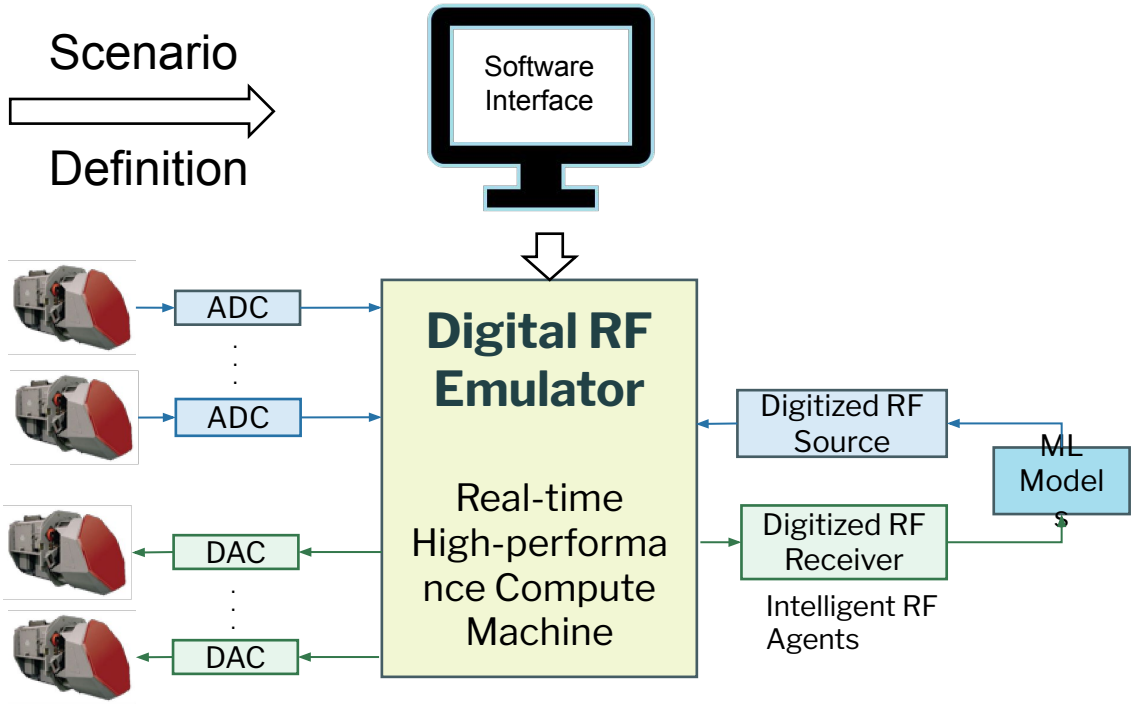
Speaker: Prof. Saibal Mukhopadhyay

School of Electrical & Computer Engineering

Georgia Institute of Technology

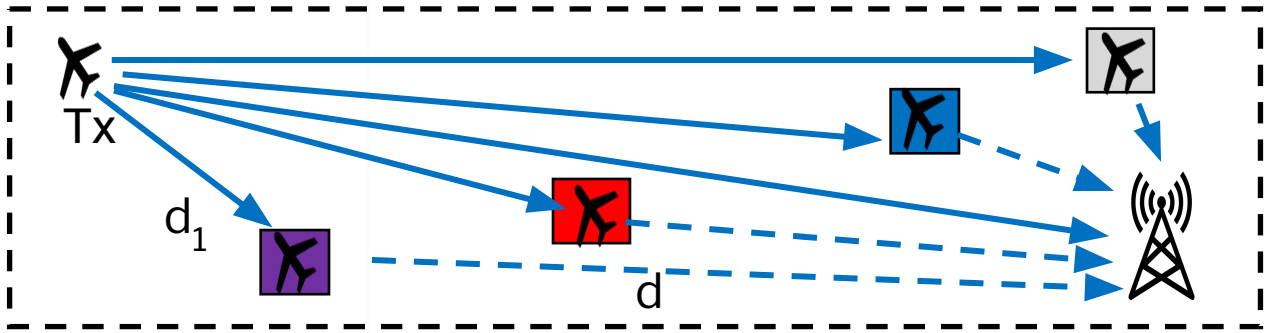
Real-time RF Emulation

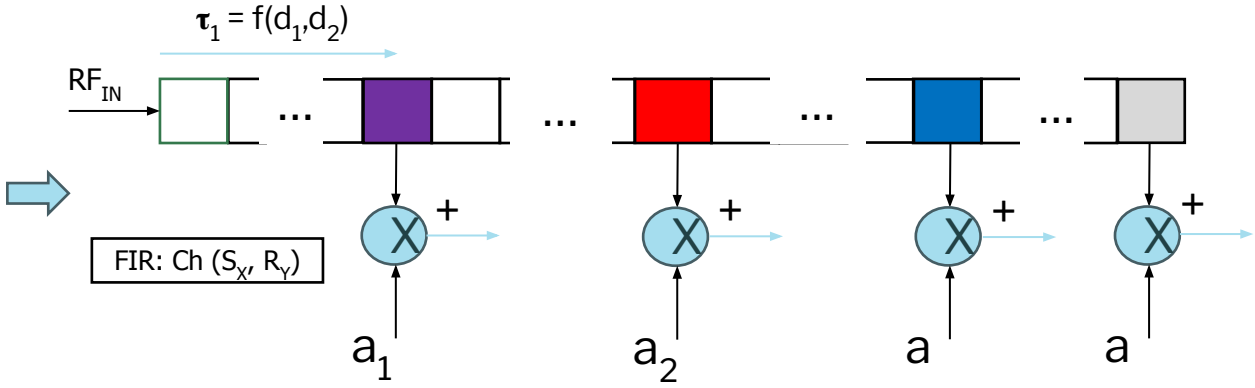




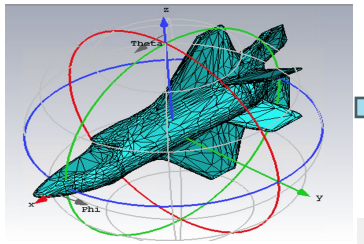
- Emulate complex RF scenarios for training and testing of RF equipment and/or machine learning enabled RF agents to reduce the need for physical testing.
- Software based simulation of RF interactions is extremely slow and do not support real-time emulation.
- Hardware acceleration can significantly advance the state-of-the-art of RF emulation.

Tapped-Delay Model for RF Emulation

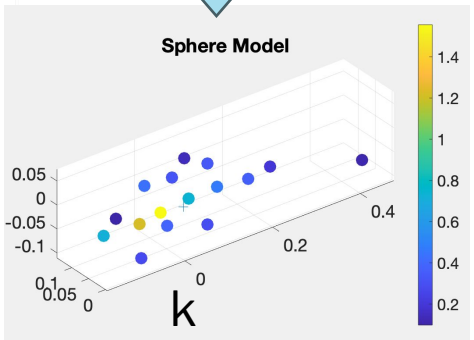


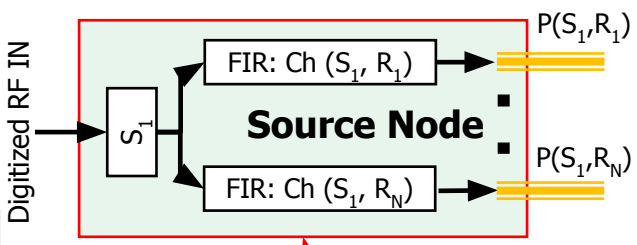


- Channel between Obj1 and Rx is represented using a tapped-delay FIR filter where each objects leads to a set-of non-zero (complex) FIR coefficients => **Sparse FIR filter**

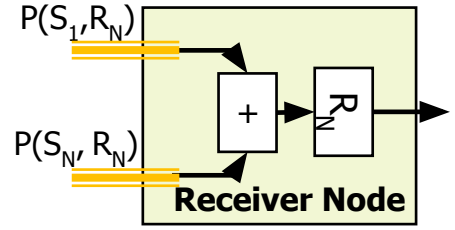


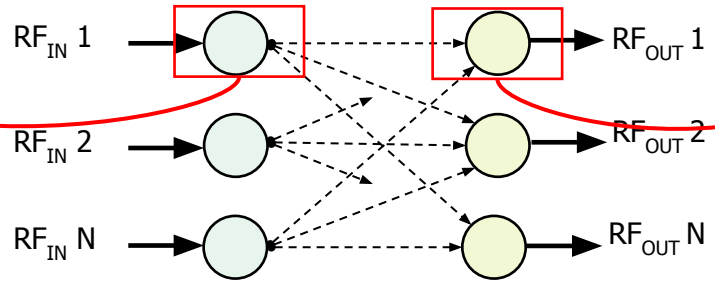
Point-scatterer model of Radar Cross Section





Computation Complexity ~ O(kN³)

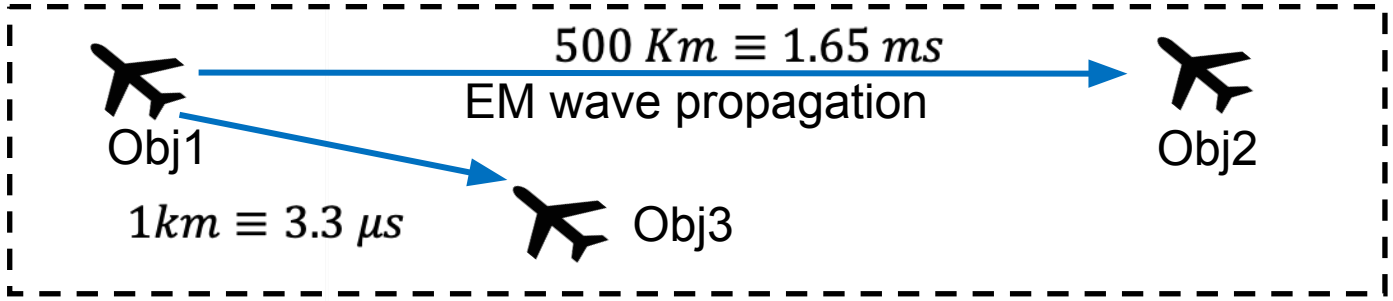




J. Seo, et. al. IMS 2022

Why is Real-time RF emulation *at Scale* Challenging?

Physical World



Compute: Emulating scenes with many objects at wideband RF rate requires high computational throughout

- *200 RF platforms (16 scatters) 2GHz of RF instantaneous bandwidth (IBW) will require 2048 PFLOPS of computations.*

Memory: Emulating far interactions require large memory.

- $OBJ1 \rightarrow OBJ2$: $\frac{500Km*IBW}{3\times10^8}$ = 3.34 Millon Samples (~12 MB)

Latency: Emulating short interactions require low compute Latency

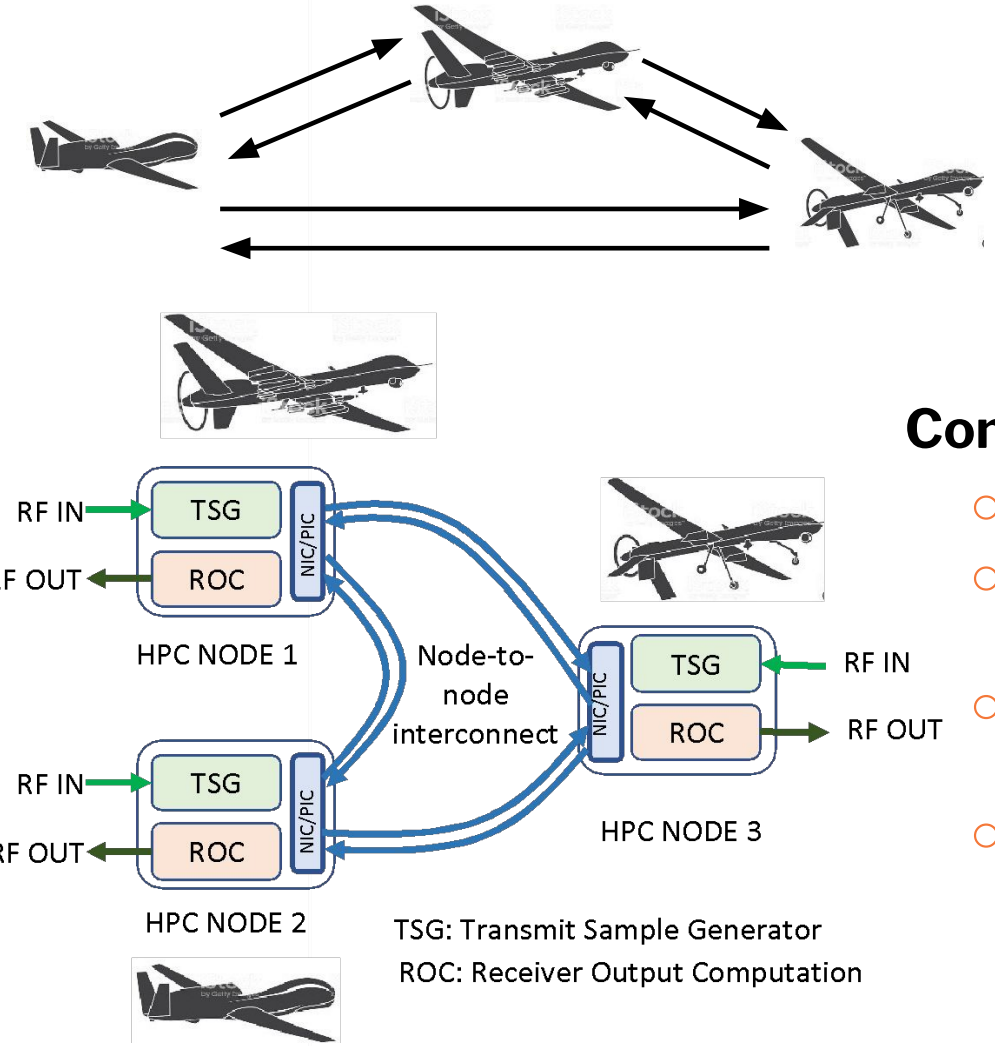
- *Latency for emulating propagation through a 1km channel < 3.3 μs*

We need innovations across the design stack to address the challenges of RF emulation

- RF emulation models that can reduce computational demand of emulation.
- Computational architecture and accelerator design to efficiently accelerate the emulation models with low-latency and high-throughput.
- A system architecture that can seamlessly scale to large emulation problem with many (100s) of objects while maintaining low latency.

Computational Model & Architecture

Direct Path Model for RF Emulation



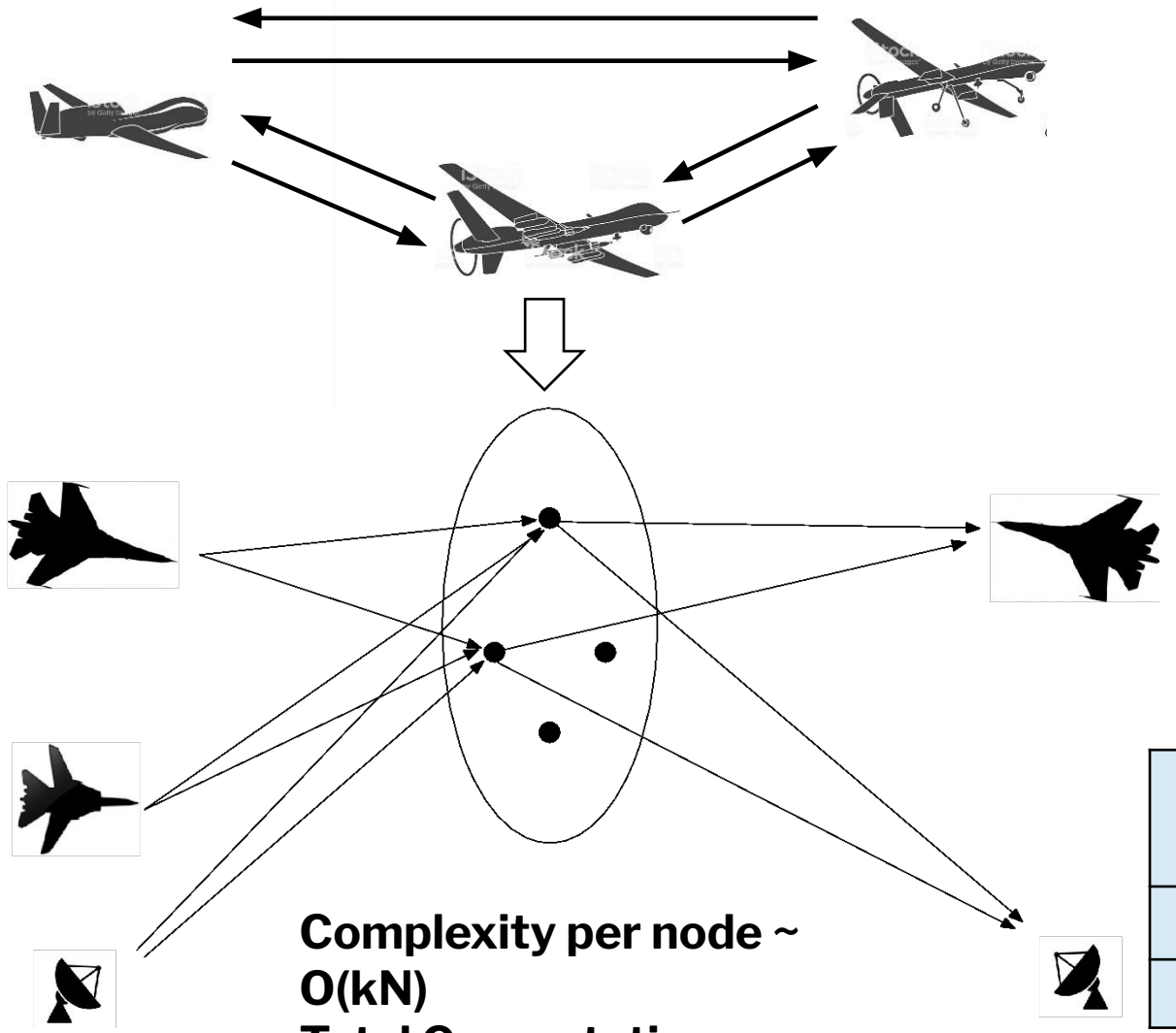
Direct path model

- Each computation node represents one object
- Two-way path between each pair of objects
- Channel creation decentralized

Computation in each node

- Receive signals from all other nodes
- Combine the received signals and generate output samples for all other nodes.
- Apply RCS gains and distance dependent direct path loss to each output signal.
- Delay each output signal depending on the direct path delay (distance dependent) and transmit to all other nodes.

Scatterer-Separable Direct Path Only Compute Model



Complexity per node ~ O(kN)
Total Computation ~ $O(kN^2)$

Radar Cross Section (RCS) emulated using multiple (k) scatterers with individual response of each scatterer being separable.

$$\sigma_k(\theta_n^i, \theta_l^o) = \sigma_k^{in}(\theta_n^{in})\sigma_k^{out}(\theta_l^{out})$$

Incoming signals weighted to form/store the scatterer response

$$v_k(t) = \sum_{n=1}^{N} \sigma_k^i(\theta_n^{in}) s_{i\to n}^{in}(t - \tau_{n,k}^{in})$$

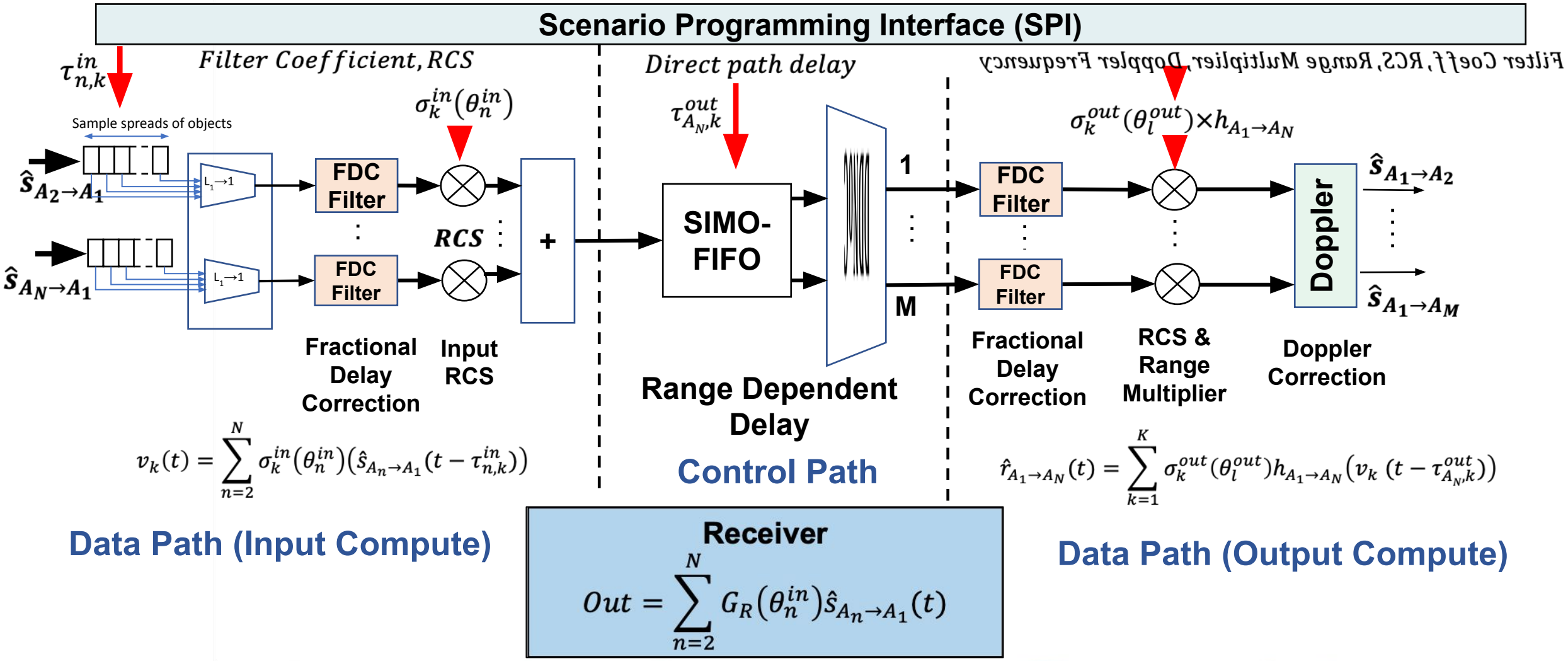
Outputs are generated by delaying scatterer samples by direct path delay (& output angle) and multiplied by direct path gain (loss)

$$\hat{r}_{A_1\to A_N}(t) = \sum_{k=1}^{K} \sigma_k^{out}(\theta_l^{out}) h_{A_1\to A_N}(v_k\,(t - \tau_{A_N,k}^{out}))$$

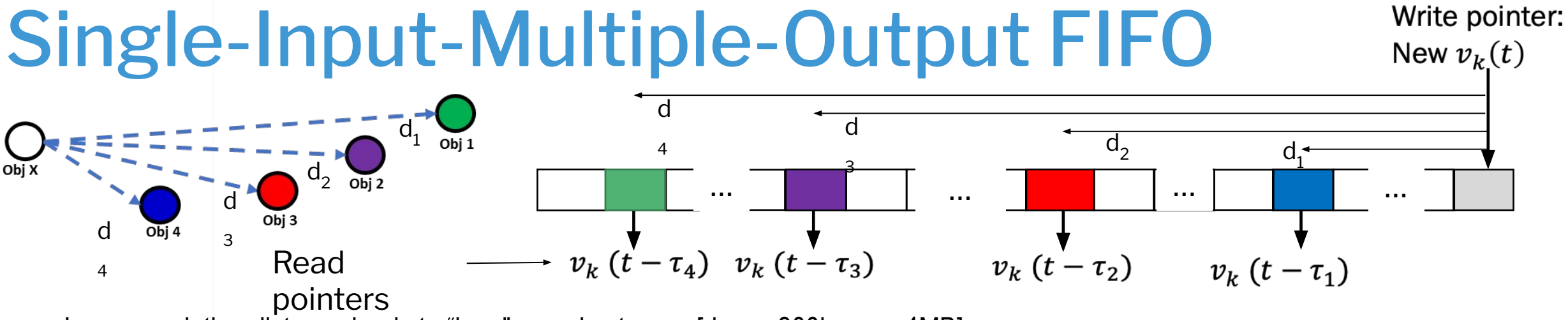
Computational Models assuming 16 scatters per object	System Size (N X N)	
	200 x 200	80 x 80
Tapped-delay	2048 PFLOPS	130 PFLOPS
Scatterer Separable	7.2 PFLOPS	1.2 PFLOPS

M. Mukherjee, et. al. Radarcon 2023

Accelerator Architecture: Single Scatterer

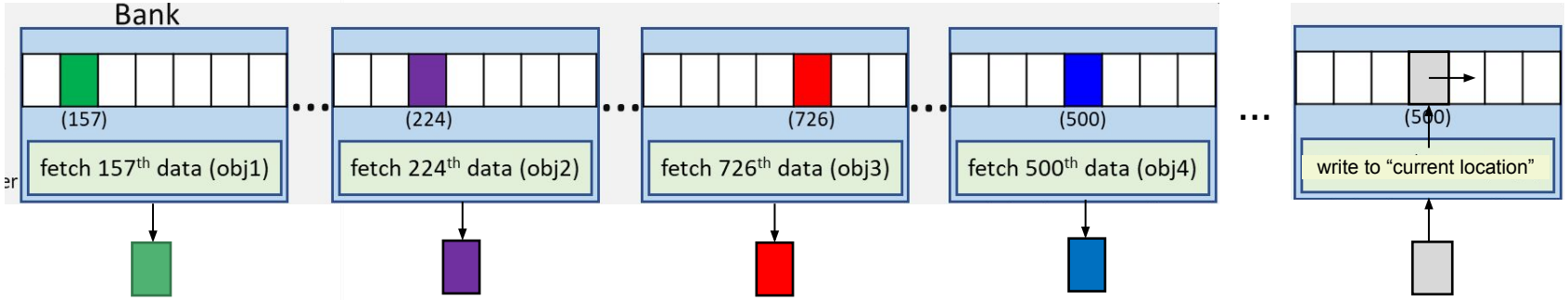


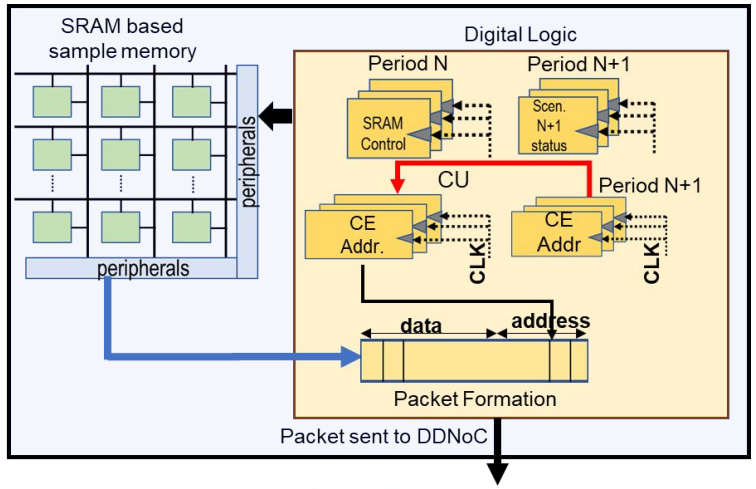
Sample Distribution: Single-Input-Multiple-Output FIFO



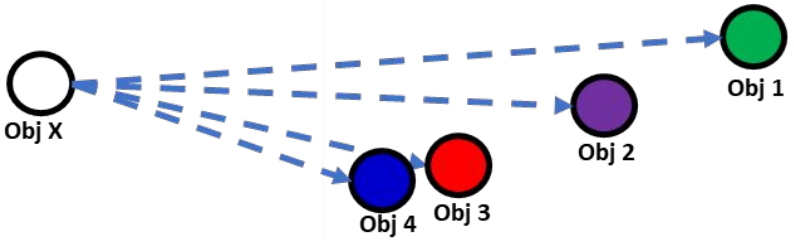
- Large emulation distance leads to "long" sample storage [d_{max} = 300km -> ~4MB]
- Shift-register based implementation is not feasible [area & power hungry]

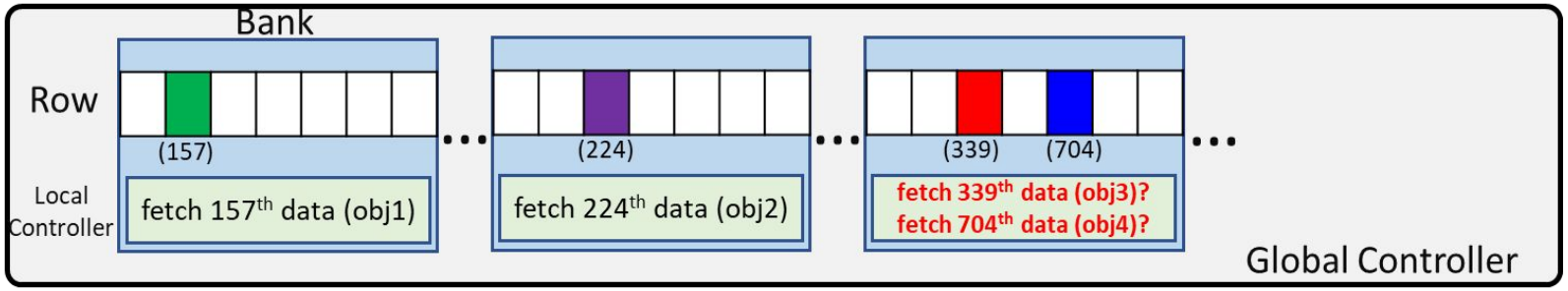
Memory centric approach to sample distribution





How to Emulate Multiple Objects at a Similar Range? Memory Collision

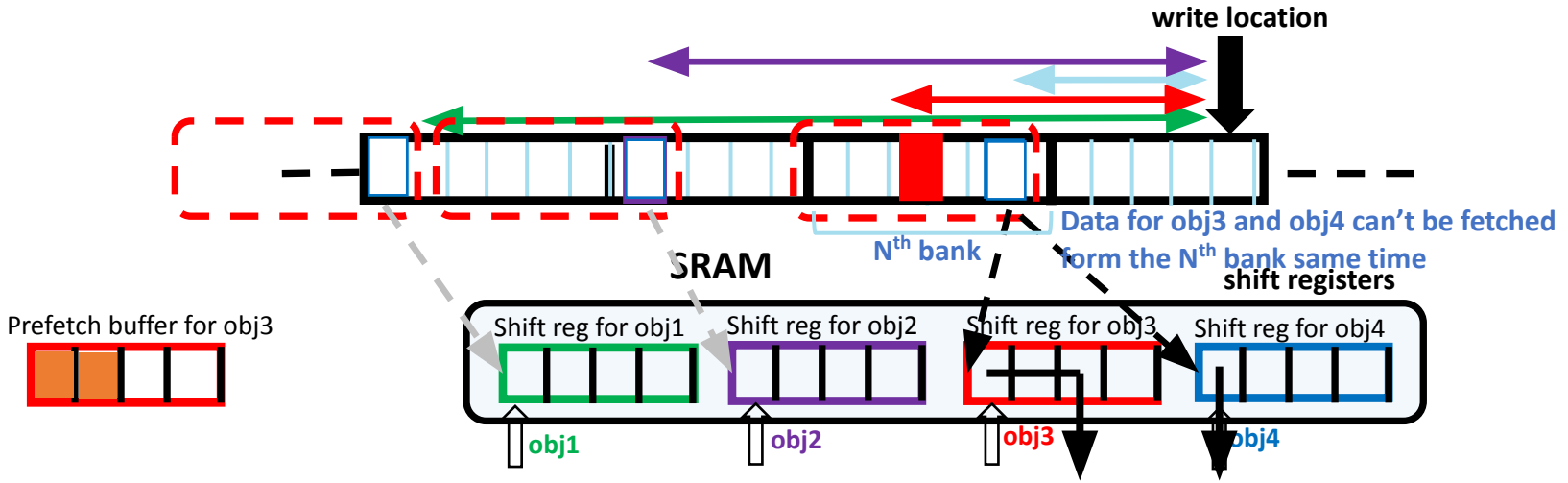




(N-1)th Scenario ①②③

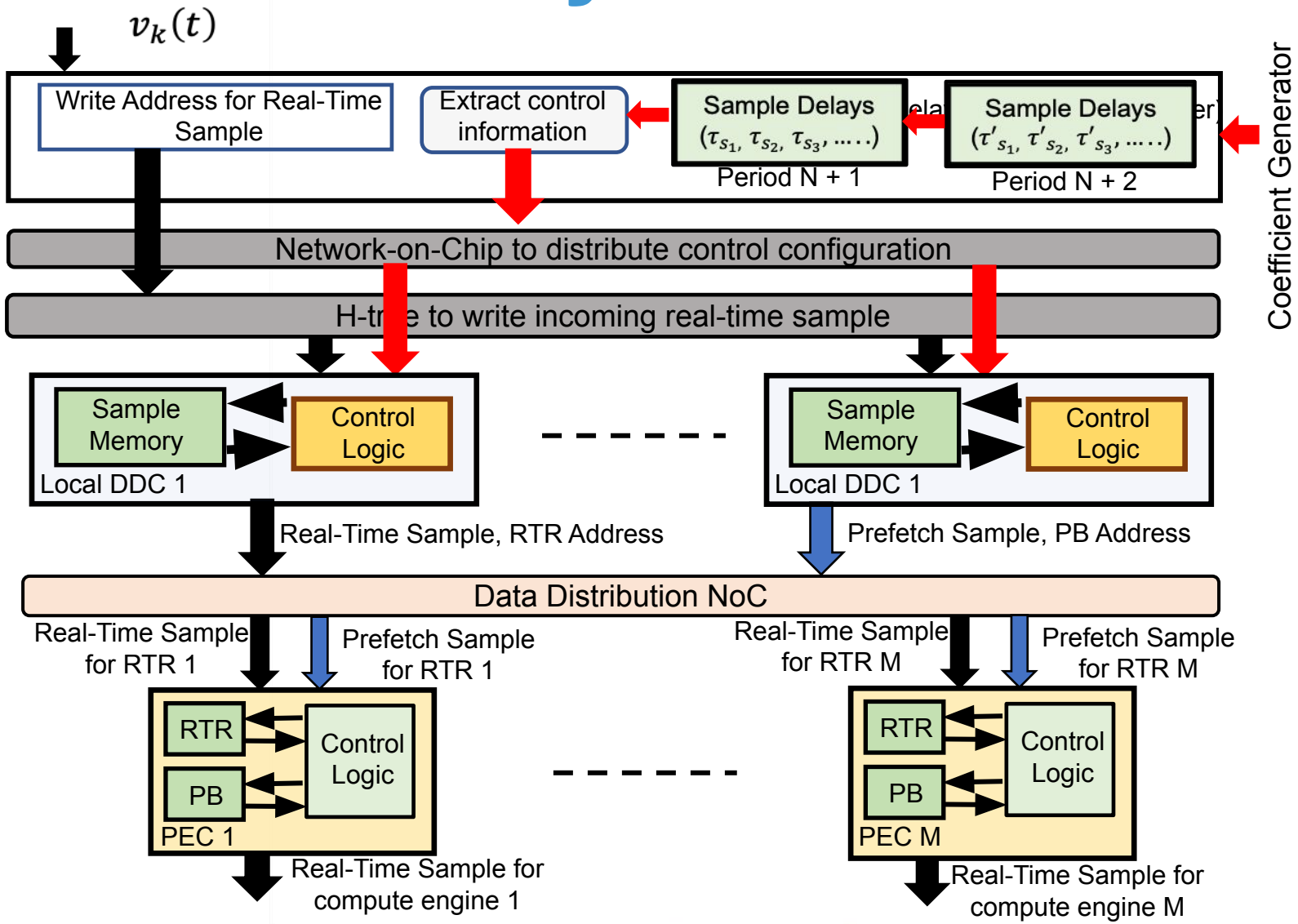
Scenario update ④

Nth Scenario ⑤⑥



ARION accelerator eliminates memory collision and enables seamless emulation of scenarios where multiple objects are at the same range from a source.

Near Memory Distributed Controller



Global Delay Distribution Controller

- Parse incoming sample delays from SPI
- Generate address for dynamic write pointer
- Generate read address & configure LDDC

Local Delay Distribution Controller

- Real-time data fetch and transmission.
- Read pointers initialized once for new scenarion. Sequential access thereafter
- Max. #active LDDC = #objects

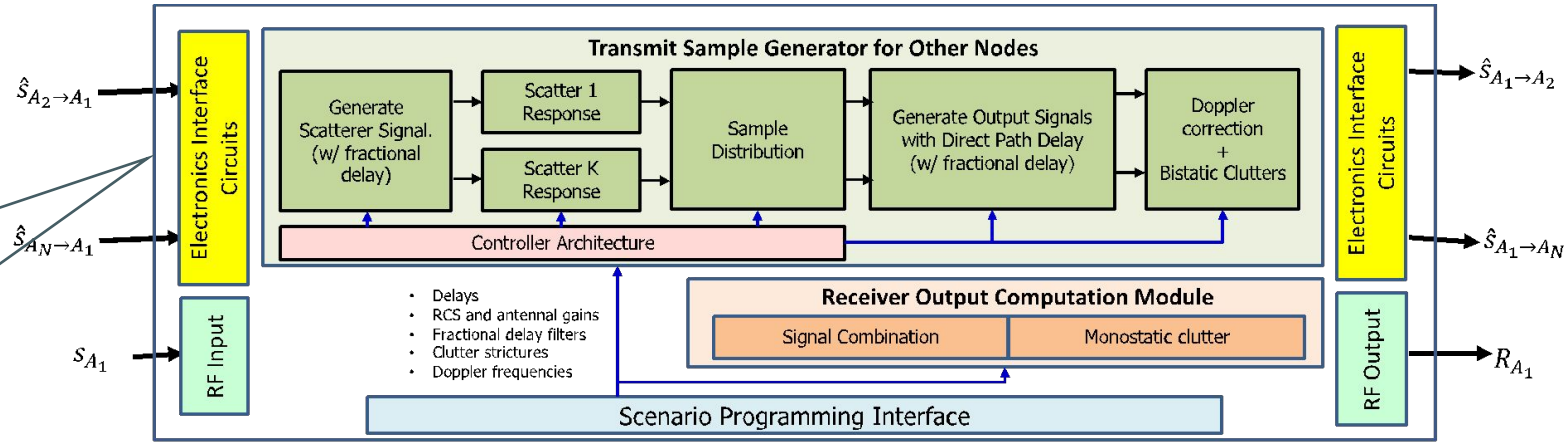
Data Distribution NoC

- High-throughput NoC
- Unicasting & multi-casting supports

Memory Collision support

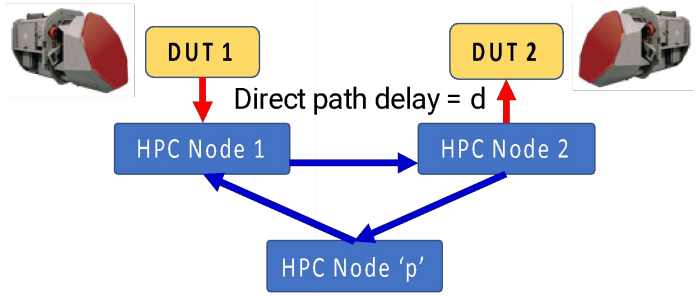
- Real-time register to delay samples
- Buffers for prefetching samples

Accelerator Architecture with Multiple Scatterers

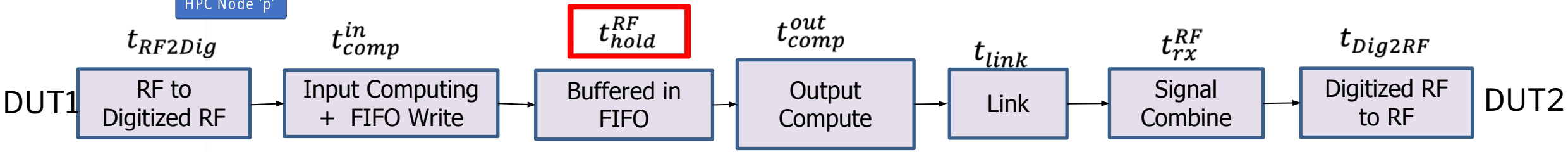


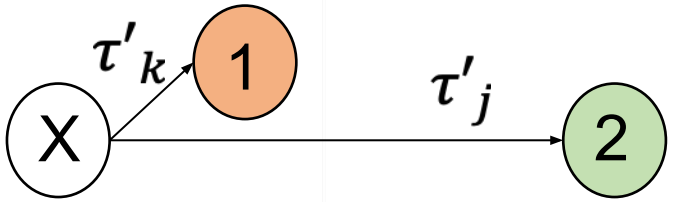
- Accelerator architecture contains: Data Path & Control Path
- Data Path applies all mathematical models for RF system except direct path delay
- Control path delays incoming signal according to expected EM wave propagation delay
- High throughput near-memory architecture

Why Computation Latency Matters?

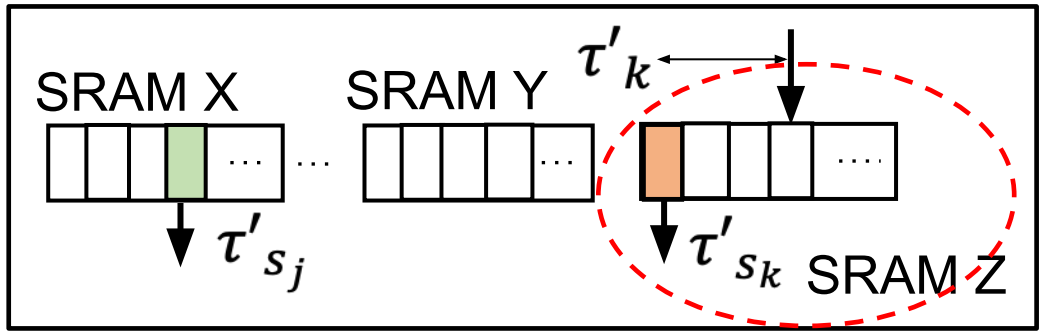


$$\tau_{s_1} \Rightarrow t_{hold}^{RF} = d - \left(t_{RF2Dig} + t_{Write}^{RF} + t_{comp}^{all} + t_{link}^{all} + t_{rx}^{RF} + t_{dig2RF}\right)$$





Cannot simultaneously read and write samples from the same SRAM



$\tau_{s_{min}} \geq$ *distance equivalent to # of samples in one SRAM bank*

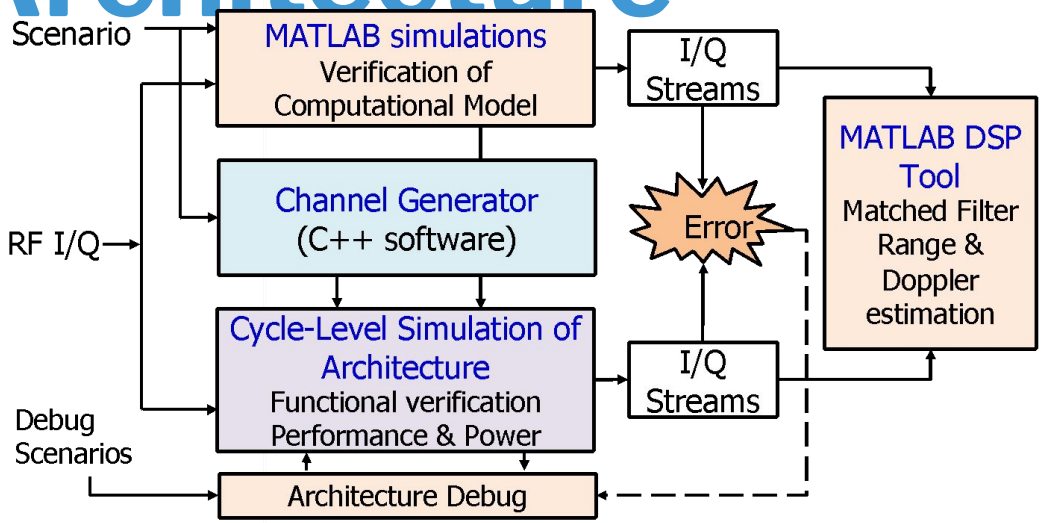
$$d_{min} = \tau_{s_{min}} + \left(t_{RF2Dig} + t_{comp}^{in} + t_{comp}^{out} + t_{link}^{all} + t_{rx}^{RF} + t_{dig2RF}\right)$$

Memory Bank Sizing

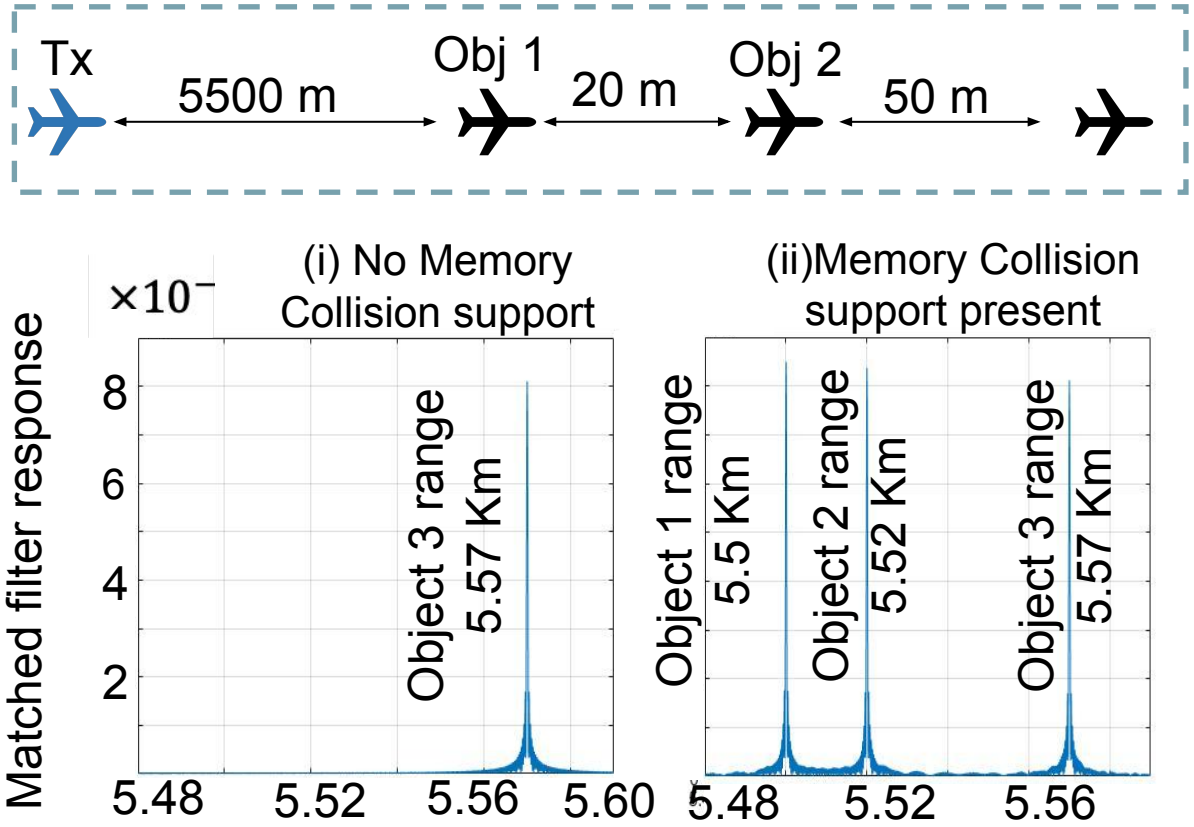
Accelerator Hardware Architecture & Physical Design

Prototyping and Validation of the Compute Model

Cycle-level Simulation of Compute Architecture

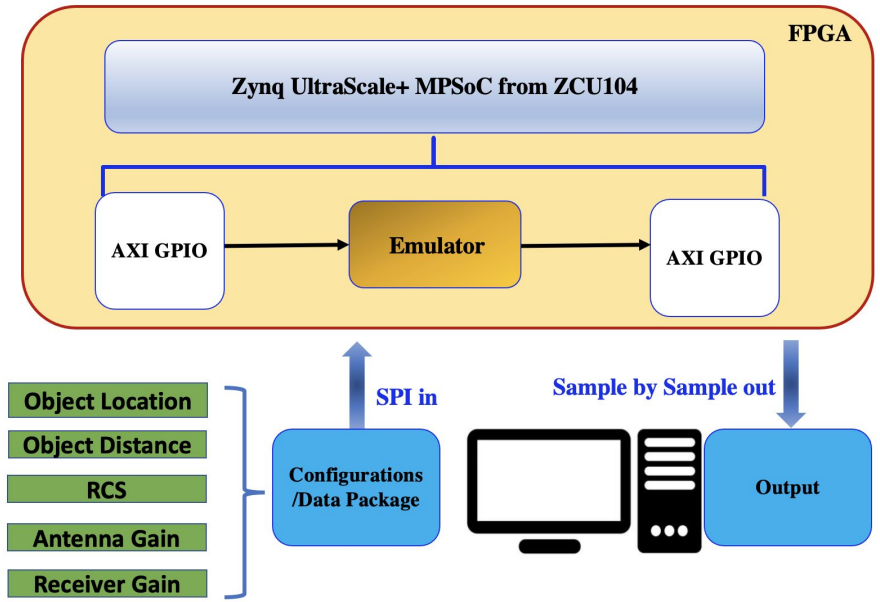


- **MATLAB simulation**
  - Test accuracy of fractional delay, doppler approximation, quantization errors
- **Cycle-level datapath simulator**
  - Test accuracy various static and dynamic scenarios, static/moving objects, memory collision prevention, large object etc.



4-object scenario with memory collision

FPGA Prototyping



Emulated system: 1Tx, 1Rx, and 4 relfector-nodes [4 Scenarios each with 8176 Sample Testing Case]

	C++ Simulation	MATLAB Simulation	Our Design
Simulation Time	10.2512s	29.8247s	0.3612ms

~28380x faster than C++

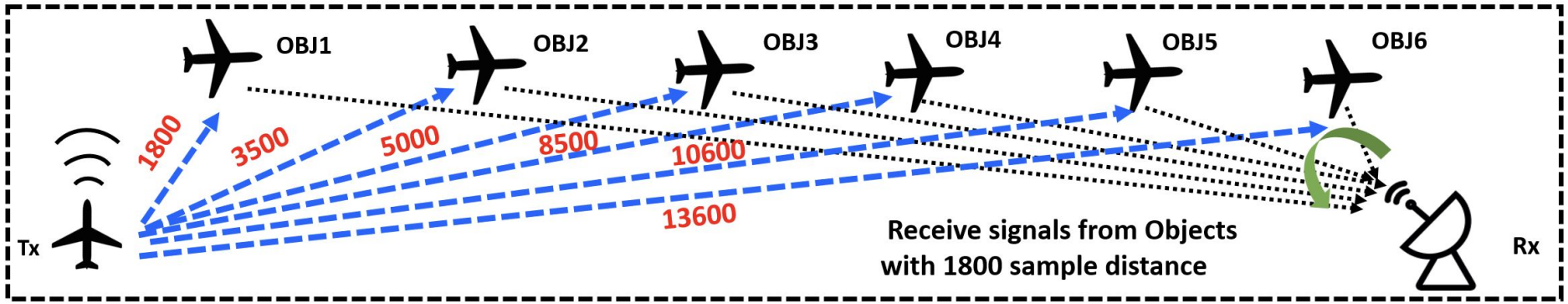
~82570x faster than MATLAB

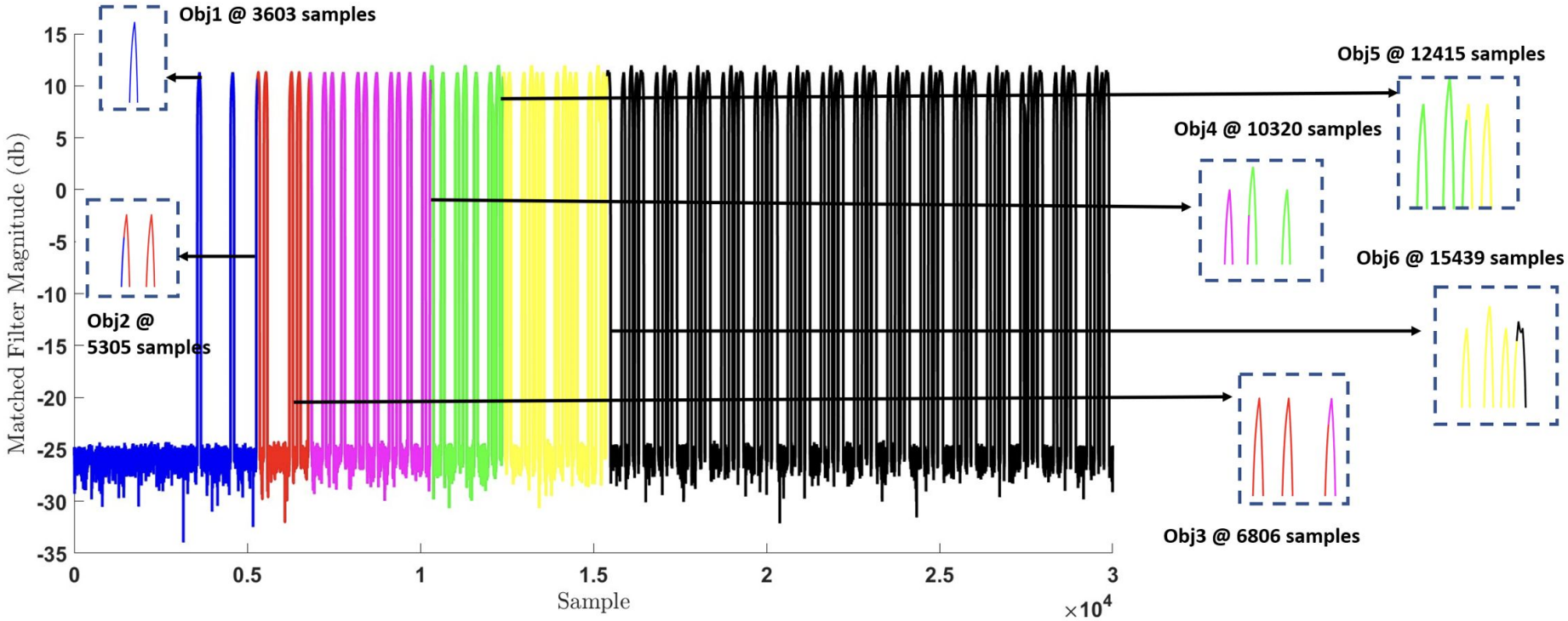
Emulation Cases	6-object	8-object
MAX IBW (MHz)	180	180
Range (km)	2.78-27.3	1.13-109.1
Power (W)	5.576	5.757

MAX Emulation Error <2%

	Davide Villa	Ashish Chaudhari	Inaki Val	This Work
Bandwidth	80MHz	100MHz	100MHz	180MHz
Year	2023	2018	2014	-

Example Emulations: Range Estimation

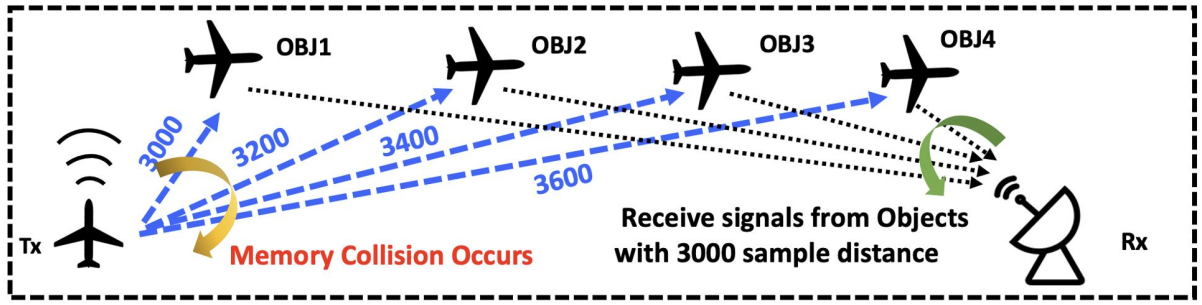


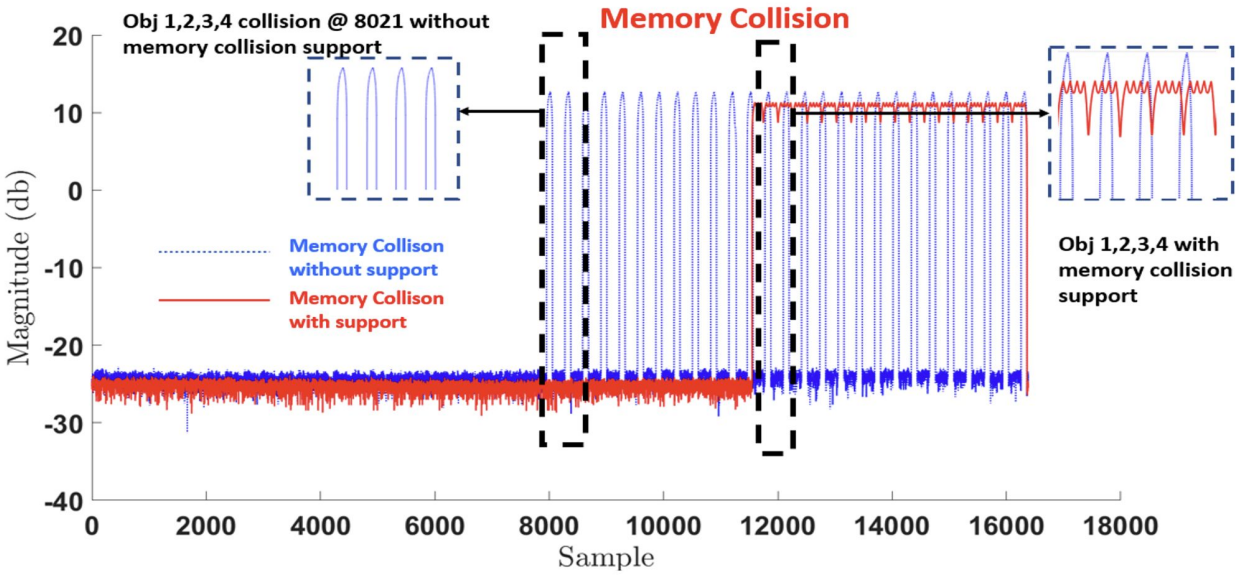


X. Mao, et. al. IMS 2023

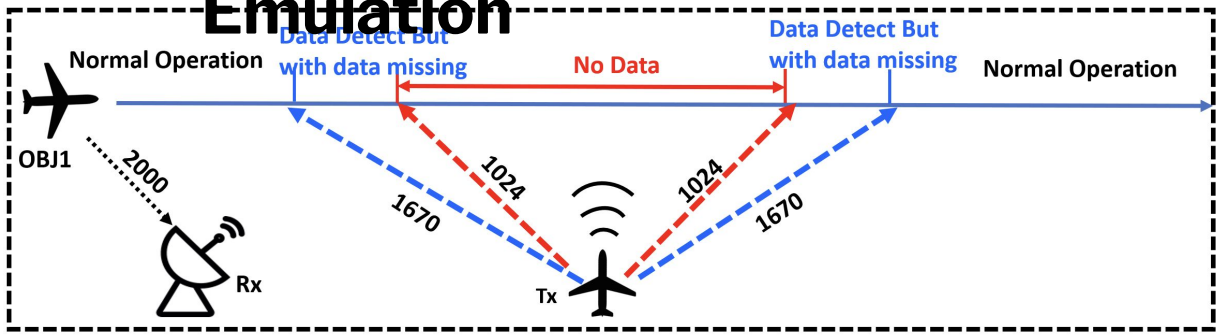
Example Emulations

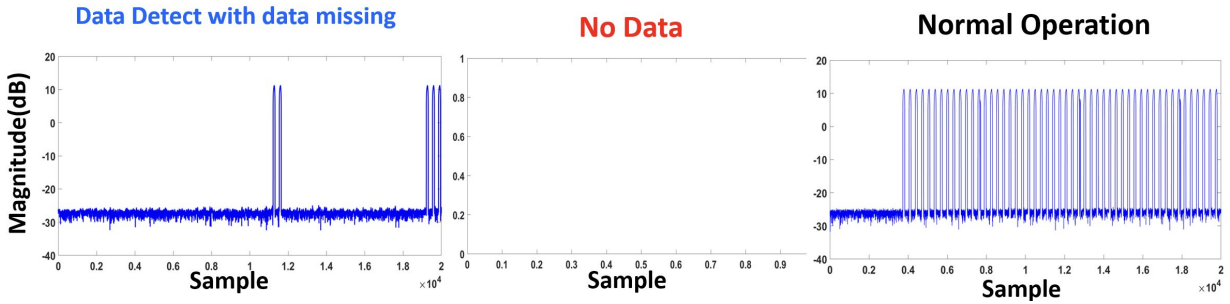
Memory





Minimum Distance Emulation

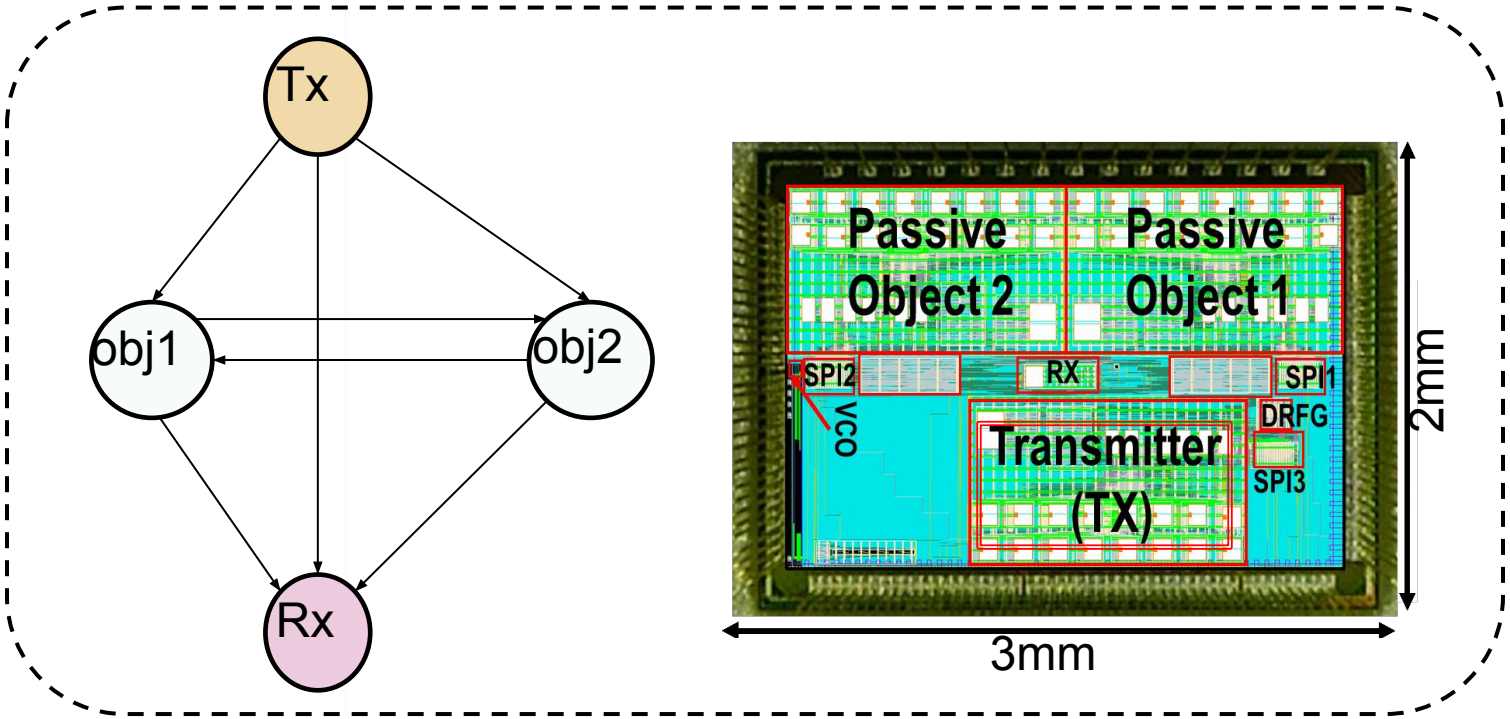




Object flies through the Tx horizontally from "far away" to "min distance", and then move Tx away again.

X. Mao, et. al. IMS 2023

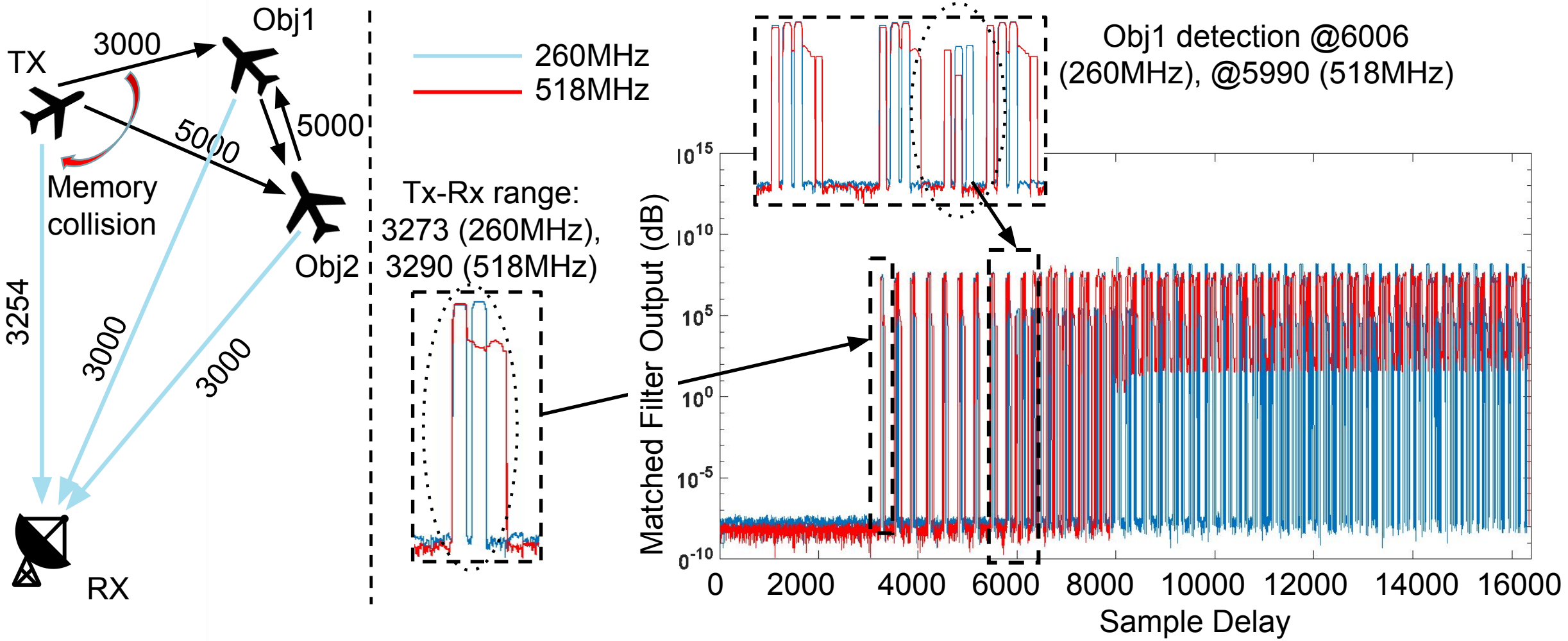
Physical Design & Test-chip



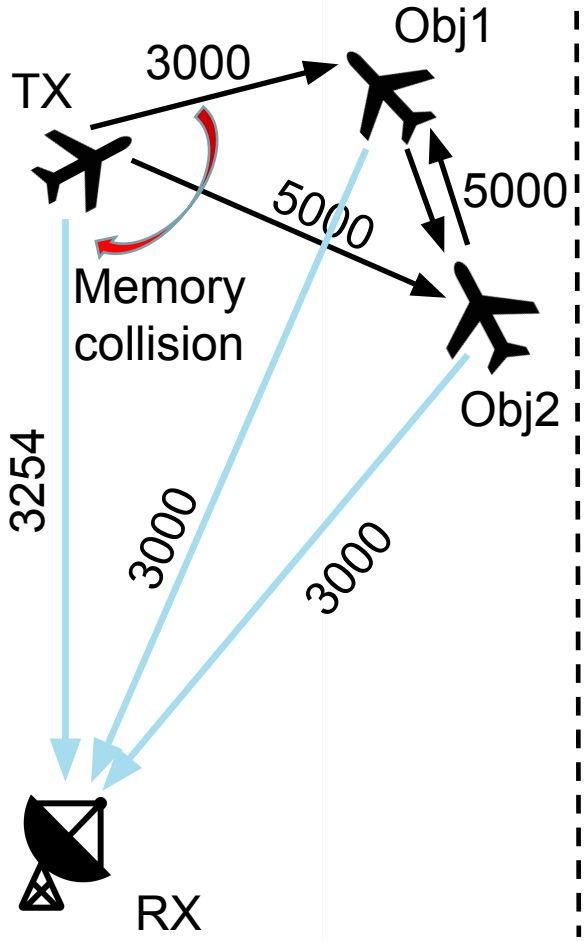
Test-chip Summary	
Technology	28nm CMOS
Chip Area	6mm^2
On-chip memory/node	64KB SRAM
Maximum BW	518MHz
Measured power	790mW
Memory collision support	6KB DPSRAM/node
Emulation Range	0.67-9.5Km
Hardware Latency	0.24µs

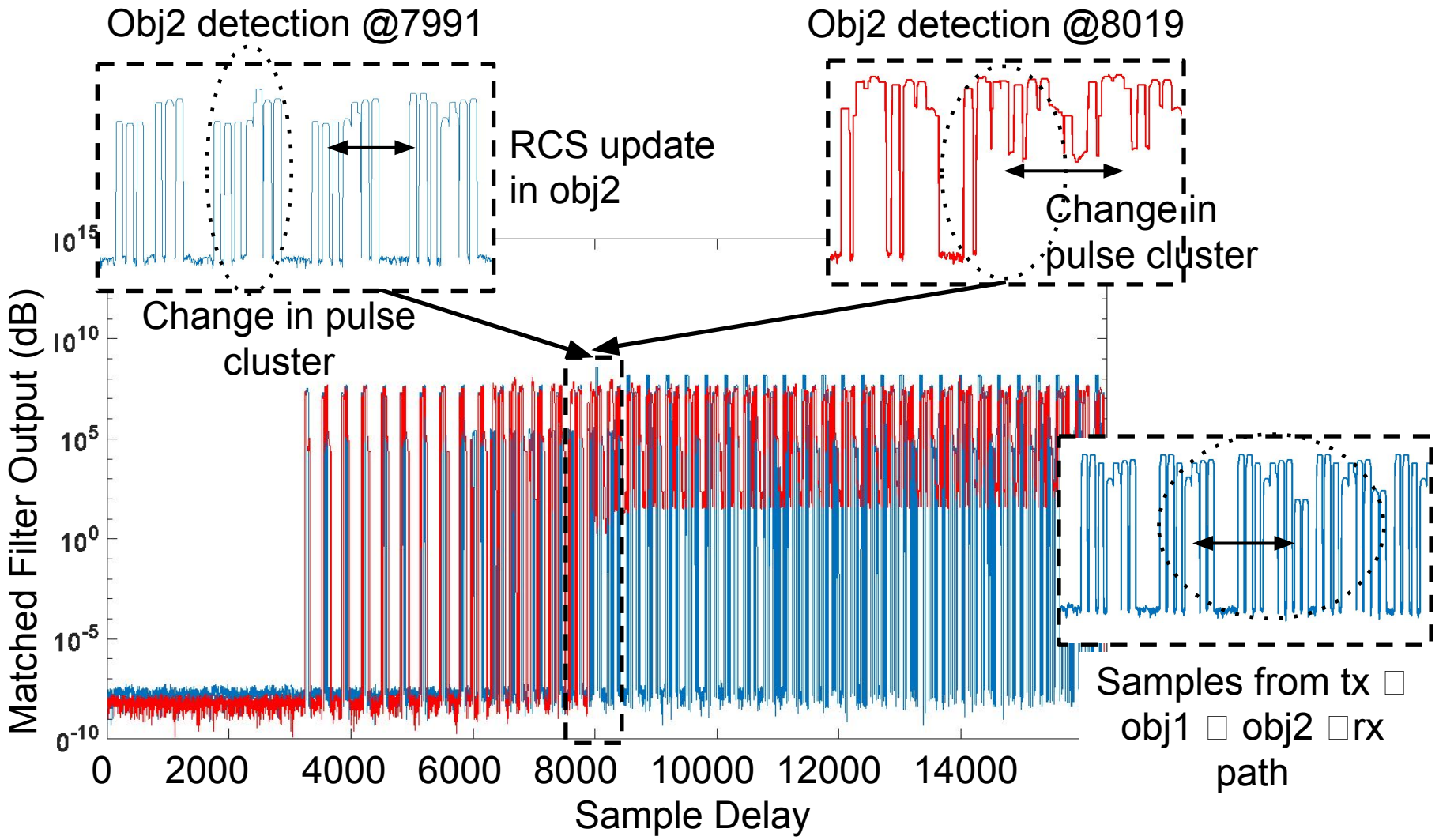
- 4-node prototype system in 28nm CMOS: Tx, Rx and 2 passive reflectors
- SIMO-FIFO in Tx and reflectors are same
- Designed with 16 LDDC + SRAM (16K samples). Memory collision support

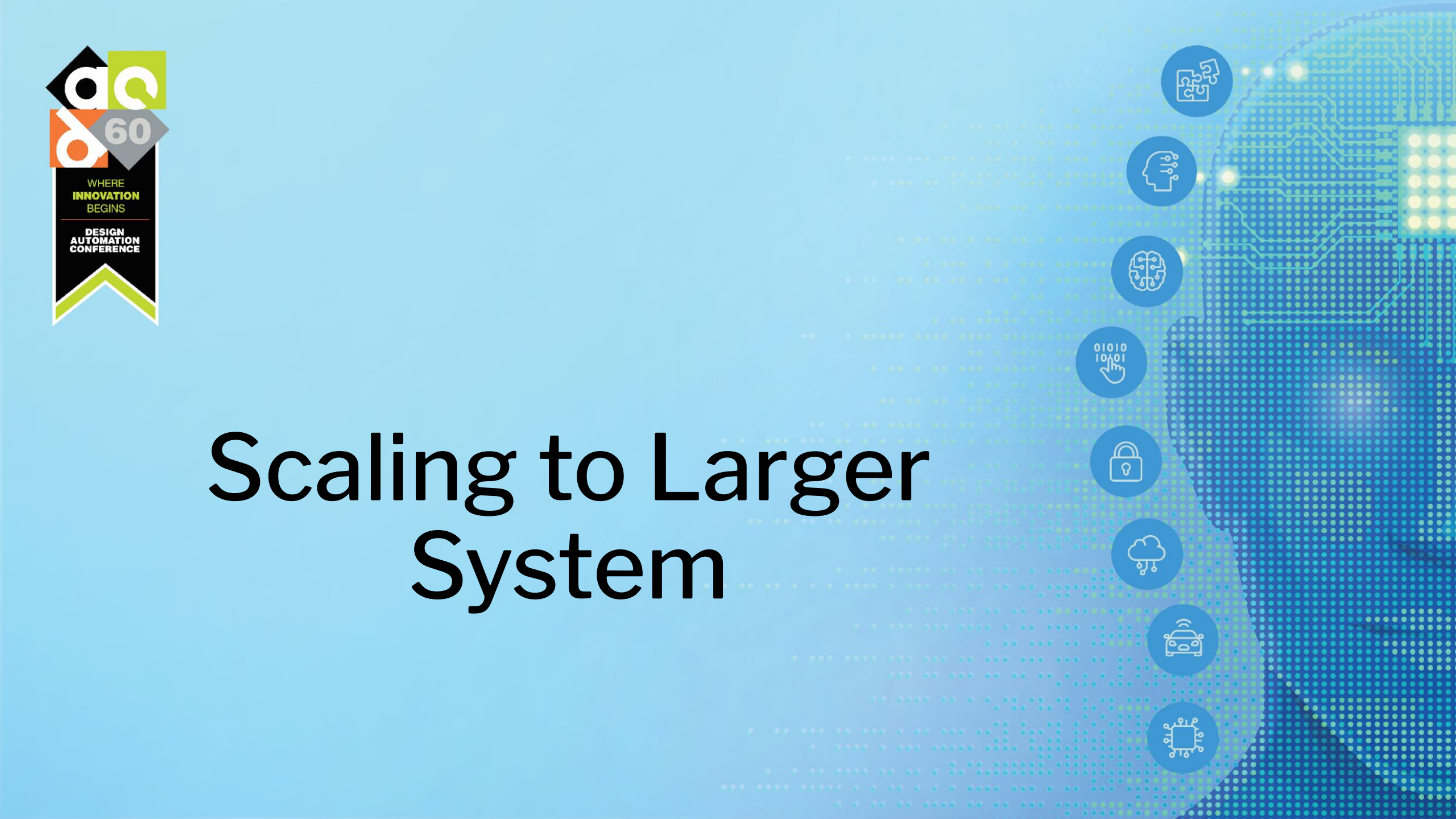
Measurement Results



Measurement Results







Scaling to Larger System

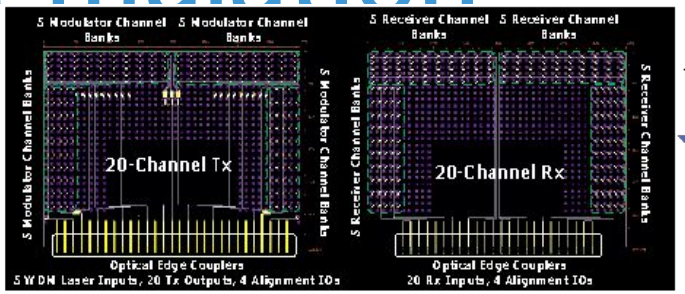
Scaling to Larger Systems with More Objects

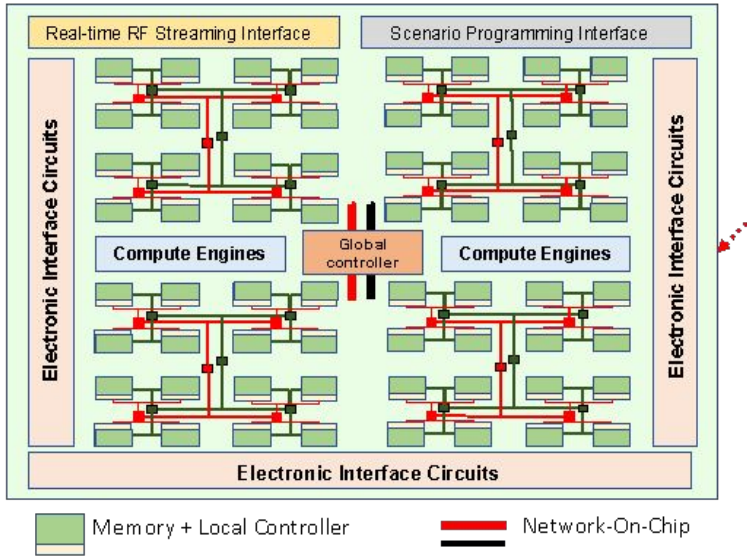
Computation [node ~ O(kN) & system ~ O(kN^2)]				
	80 Nodes		200 nodes	
	Node	System	Node	System
Baseline Interaction	15 TFLOPS	1.2 PFLOPS	36 TFLOPS	7.2 PFLOPS
Including Models of Other Physical Phenomenon	208 TFLOPS	16.7 PFLOPS	515 TFLOPS	103 PFLOPS

I/O data between nodes		
# Objects	Total Bandwidth	
	IO BW per node	Bi-Directional BW
8	1.4Tb/s	11.26 Tb/s
80	14.1 Tb/s	1.13 Pb/s
200	35.2 Tb/s	7.04 Pb/s

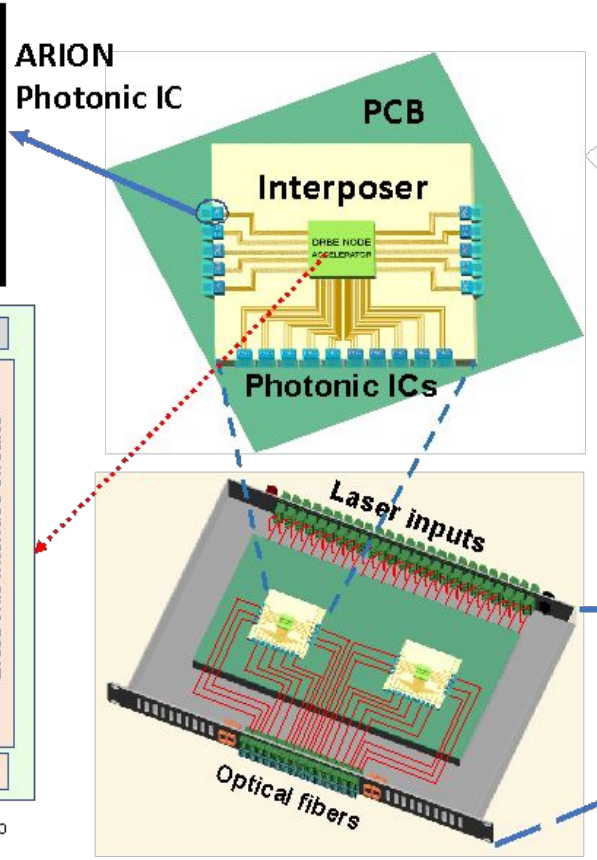
- Physical distributed computation with chip-to-chip high-bandwidth connectivity is necessary to emulate larger system.
- The system design challenge is to ensure the computation latency remains largely unchanged even if the system size grows to emulate more objects to be able to emulate close interactions.

ARION HPC Platform for Real-time RF Emulation

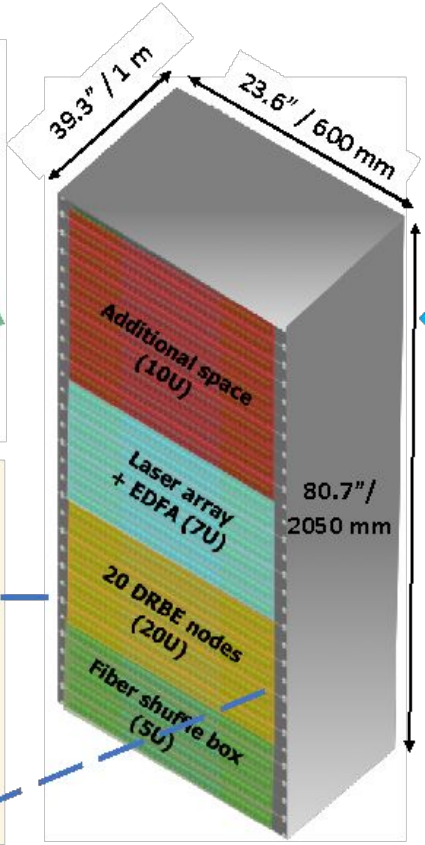




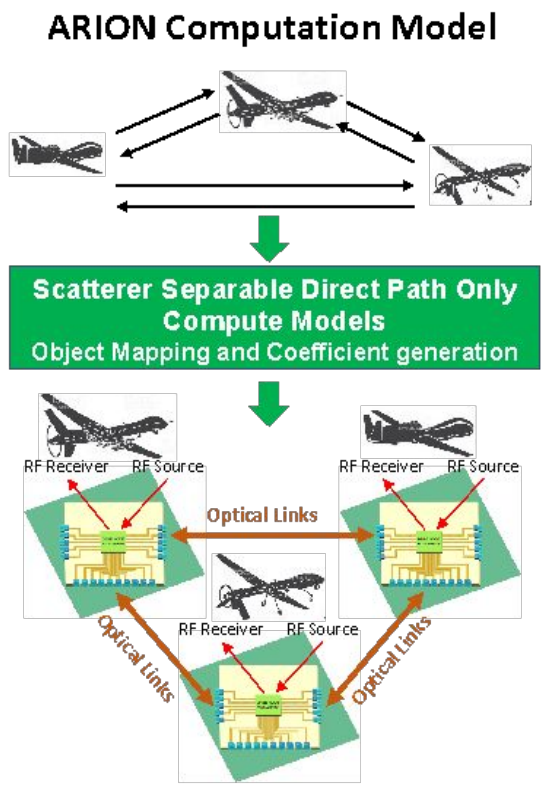
ARION Accelerator (Compute-In-Memory)



ARION Processing Node



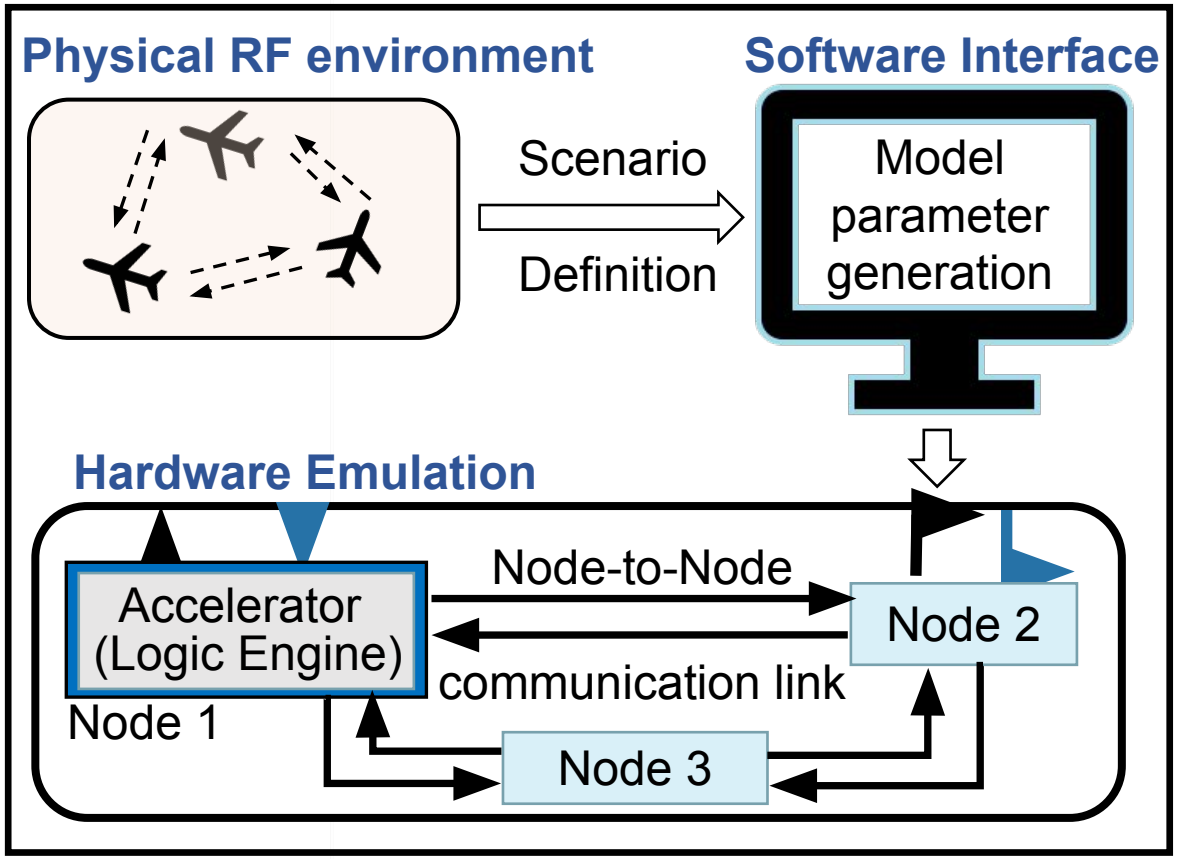
ARION HPC RACK

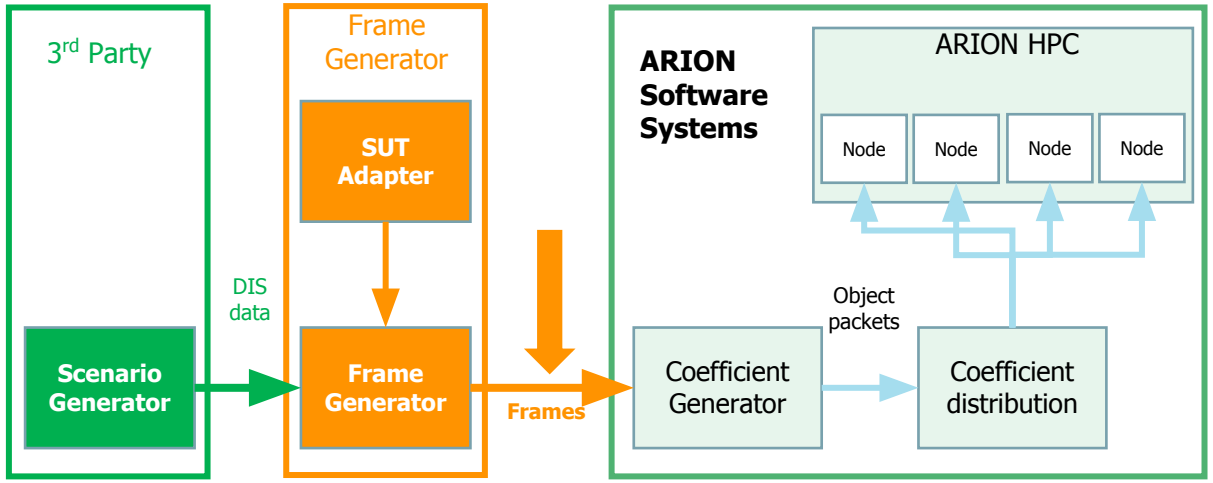


Each object is mapped to one ARION node to compute all physical phenomena. All-to-all optical network emulates multi-object interactions.

- Heterogeneous Integration effort is led by **Prof. Madhavan Swaminathan**
- Networking & HPC efforts are led by **Profs. Keren Bergman, Mingoo Seok, and Luca Carloni**

Software-based Coefficient Generator





Low (~ms) real-time frame-by-frame coefficient generation is necessary to be able to model "fast" moving scenarios

Multi-threaded Control Software

- Ability to design, debug and test a variety of scenarios
- Support heterogeneous platforms with multiple Tx / Rxs
- Supports for dynamic scenarios including addition, removal, enabling, disabling of platforms and dynamic changes in RCS and interactions between platforms
- Vector intrinsics, data locality, multi-threading to reduce latency
- Demonstrated < 1ms of coefficient generation latency.

Software effort on coefficient generator is led by **Prof. Santosh Pande**

Summary

- **Real-time emulation of radio frequency signal interaction can assist with design, training, and testing of various RF systems.**
  - Machine learning enabled Cognitive RF systems can benefit from fast generation of large volume of training data via RF emulation.
- **Design of real-time RF emulator is challenging as we simultaneously need**
  - large computation (complex physical models)
  - high sample-by-sample throughput (RF bandwidth)
  - low latency (emulation of close interactions)
  - large memory (emulation of far interactions)
- **ARION effort have studied design of special purpose architectures that couples innovative compute with novel data-flow models to address preceding challenges.**
- **Ultimately, the design effort is required at the system scale including optimal networking, heterogeneous integration, and large scale HPC system design.**

ARION Team


Mukhopadhyay (GT)


Krishna (GT)


Swaminathan (GT)


Romberg (GT)


Pande (GT)


Bergman (CU)


Seok (CU)


Carloni (CU)

Students on Compute Model and Compute Architecture Design

NOT APPROVED FOR PUBLIC RELEASE.