**Overview**

**Motivation & Objectives**
- Increasingly networked cyber-physical systems
  - Internet of Things (IoT), Wireless Sensor Networks (WSNs)
  - Distributed data collection, aggregation and processing
- Tight computation and communication coupling
  - Non-obvious interactions and tradeoffs
  - Traditionally networks and systems are designed in isolation
  - Ignores joint optimization challenges and opportunities
- Systematic computation/communication co-design
  - Comprehensive design space exploration
  - Joint consideration of design parameters from applications to network configurations and system platform definitions
- Unified models of computation and communication (MoCC) for NoS specification
- Extension of existing (synchronous) dataflow models
- Empty tokens and actor variants: expose network losses to the application level
- Reactive island: firing idle variant of source actor triggers all subsequent connected actors to fire their idle variants

**Design Flow**
- From network-level specification
  - Formal models of computation and communication for high-level network-of-systems (NoS) specification
  - Exposing network uncertainties
  - Dynamic aspects of adaptivity and reactivity
- To networked system implementation
  - Network and system co-design
  - Architecture definition and application mapping
  - Fast yet accurate network-of-systems (NoS) simulation for validation, prototyping and exploration
- Specification & implementation models for NoS design automation

**Network-of-Systems (NoS) Specification**

**Motivation**
- Traditional models
  - Models of computation require lossless communication and can not simultaneously capture streaming and reactive behavior
  - Models of communication support richer network semantics but do not account for expressing system computation & concurrency
- Unified models of computation and communication (MoCC) for NoS specification
- Extension of existing (synchronous) dataflow models
- Empty tokens (e)
  - Lost data and absence of sporadic events in input patterns
  - Maintain guaranteed determinism
- Actor variants
  - Different variants per token patterns
  - Idle version executed when input patterns are all empty-tokens

**Adaptivity**
- Empty tokens and actor variants: expose network losses to the application level
- Reactive island: firing idle variant of source actor triggers all subsequent connected actors to fire their idle variants

**Performance Analysis**
- Worst-case throughput and latency
- Conversion to scenario-based/modal model leads to exponential complexity
  - Account for all possible actor variant combinations
- Calculate throughput & latency of the graph formed by taking the WCET of each actor
  - Lossy channels isolate actor variants
  - Might under-estimate the worst-case

**Future Work**
- Implementation of RADF semantics
  - Multiple distributed implementation choices
- Analysis techniques for probabilistic performance metrics
  - Tradeoff between latency, throughput and QoS versus token loss probability

**Network-of-Systems (NoS) Simulation**

**Motivation**
- Traditional system models
  - Over-simplified system models
  - Flexible to instantiate a wide range of configurations
- Traditional network models
  - Over-simplified network models
  - Combined with instruction-set models
  - Abstract operating system (OS) model
  - SystemC transaction-level modeling (TLM) base
- Network system co-simulation
  - Capture and emulate complicated system/network interactions
  - Fast and accurate to support large scale and complexity of NoS
  - Flexible to instantiate a wide range of configurations

**Host-compiled NoS Simulator**
- Host-compiled (HC) system simulator
  - Source-level back-annotated application model
  - Abstract operating system (OS) model
  - Network stack model (nW)
  - SystemC transaction-level modeling (TLM) base
- Network simulation backplane
  - OMNeT++ network simulator
  - OMNeT++ network topology
  - INET package for media access (MAC) and physical (PHY) layer simulation
  - Host-compiled SystemC device instances in an overall OMNeT++/INET network topology

**NoS Design Space Exploration**
- IoT application case study
  - ECG diagnosis application
  - 4 offloading stages (O-n)
  - Wireless client-server topology
- System/network parameters
  - Client/server core types and counts ($S_xY_Y$)
  - Communication protocols

**References**

**Simulation Speed**
- Simulation speed
  - 0.18 simulated sec./real sec on average

**Supported by National Science Foundation (NSF) Grant CSR-1421642**