Network/System Co-simulation Platform for Design Space Exploration of IoT Applications

Zhuoran Zhao[1], Vasileios Tsoutsouras[2], Dimitrios Soudris[2] and Andreas Gerstlauer[1]

[1]The University of Texas at Austin

[2]National Technical University of Athens

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

National Technical University of Athens

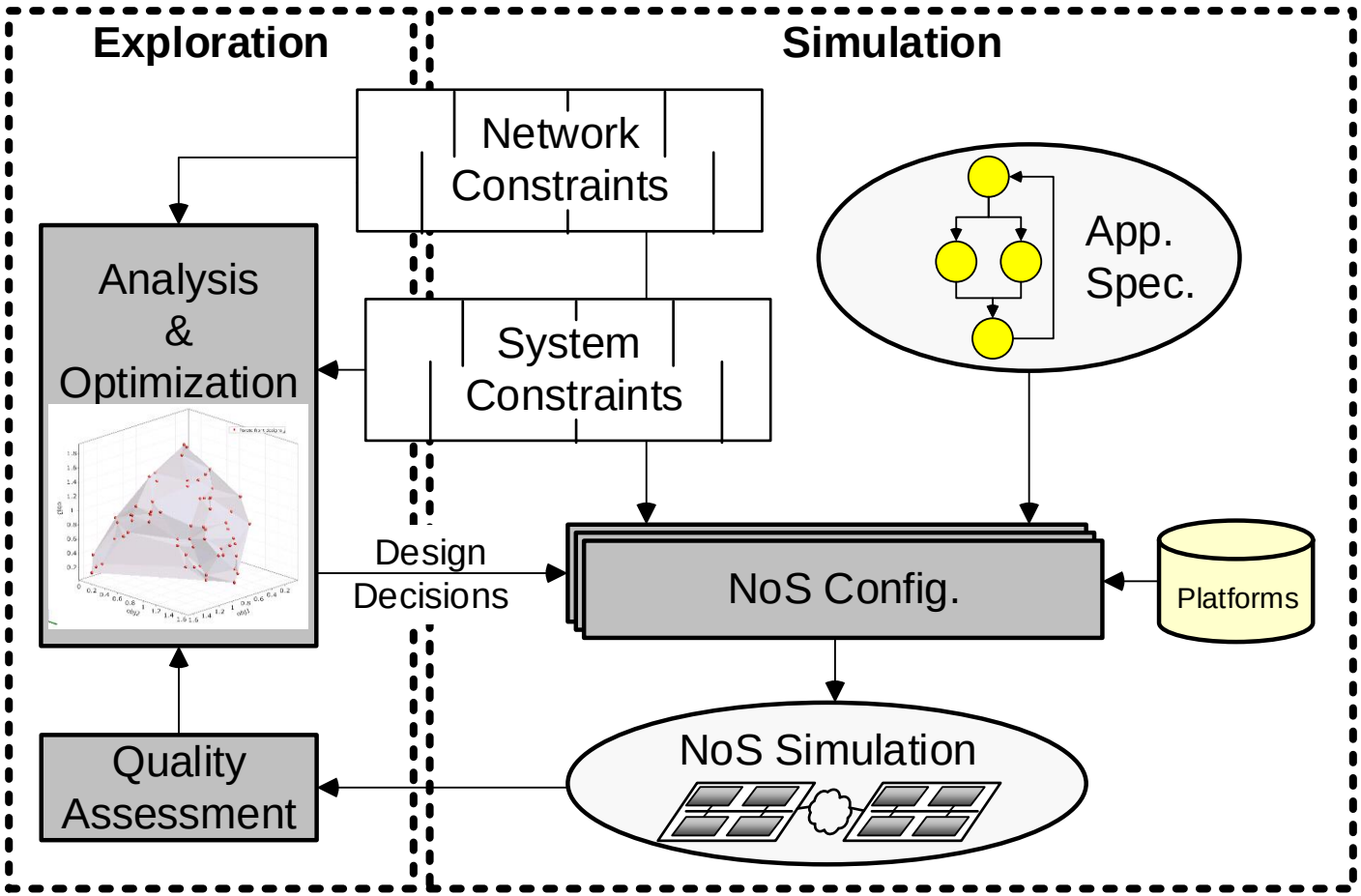
Background

- **Growing complexity and scale of future IoT applications**
  - Distributed data collection, aggregation and processing
  - Tightly coupled communication and computation workload
  - ➢ Non-obvious interactions and tradeoffs
- **Networks-of-Systems (NoS) environment**
  - System & network architecture configurations
  - Application mapping and offloading
  - ➢ Complex application/network/system interactions

➢ **Network and system co-design**

- Traditionally designed in isolation
- ➢ Joint consideration of design parameters from applications to network configurations and system platform definitions

Network/System Co-Design



- **Flexible NoS simulation platform to instantiate various network/system configurations for exploration**

Related Work

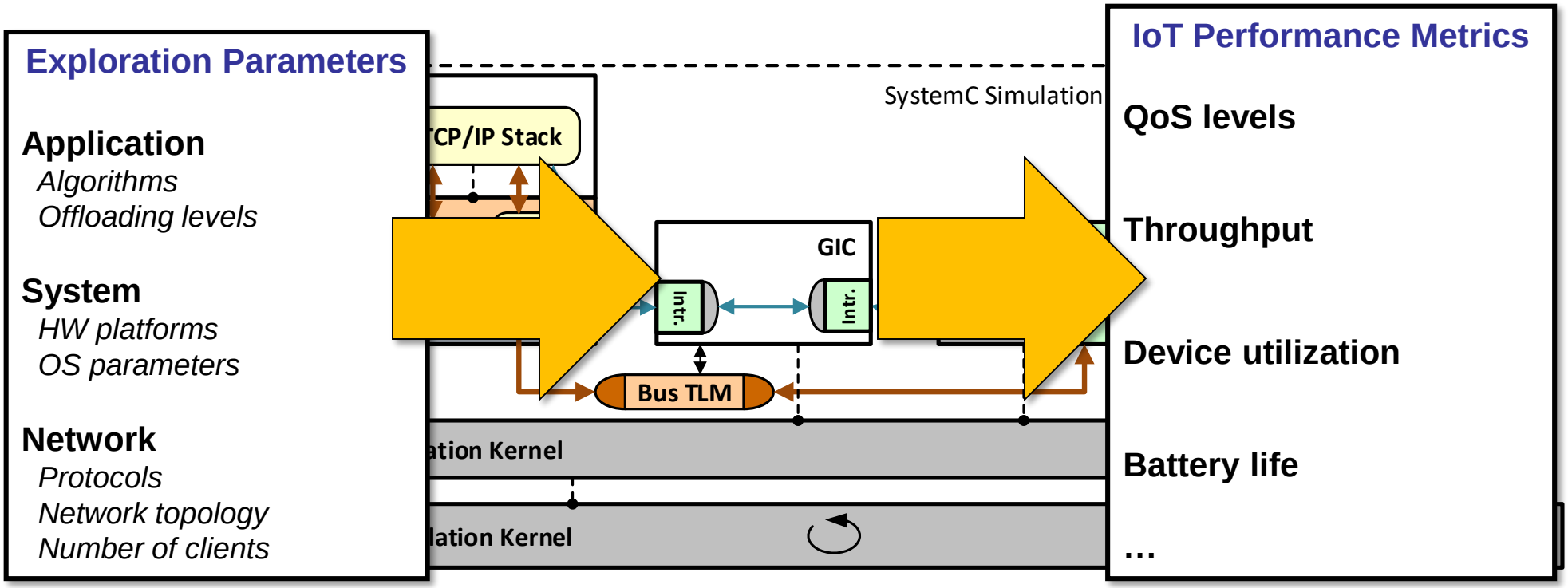
- **IoT design space exploration**
  - Application-specific with over-simplified models
    - Smart camera networks [Devarajan'06, Quaritsch'07]
    - Healthcare systems [Doukas'12, Catarinucci'15]
  - ➢ Limited design space exploration
- **Network and system simulation**
  - Existing network simulators
    - Traditional network simulators (ns-3, OMNeT++) model system devices without detailed system architectures
    - State-based system models and over-simplified network models in WSN-oriented simulators [Sommer'09, Bai'11, Du'11, Damm'10]
  - Existing system simulators
    - Host-compiled TLM simulation [Bringmann'15]
    - Simple network extension [Fummi'08, Banerjee'09]
  - ➢ Comprehensive NoS simulation platform are lacking

Outline

- ✓ **Introduction**
  - ✓ Motivation, background
  - ✓ Related work
- **NoS simulation platform overview**
  - System simulation model
  - Network simulation backplane
- **Experimental setup**
  - IoT application case studies
  - Exploration results
  - Simulation speed and accuracy

NoS Simulation Platform Overview

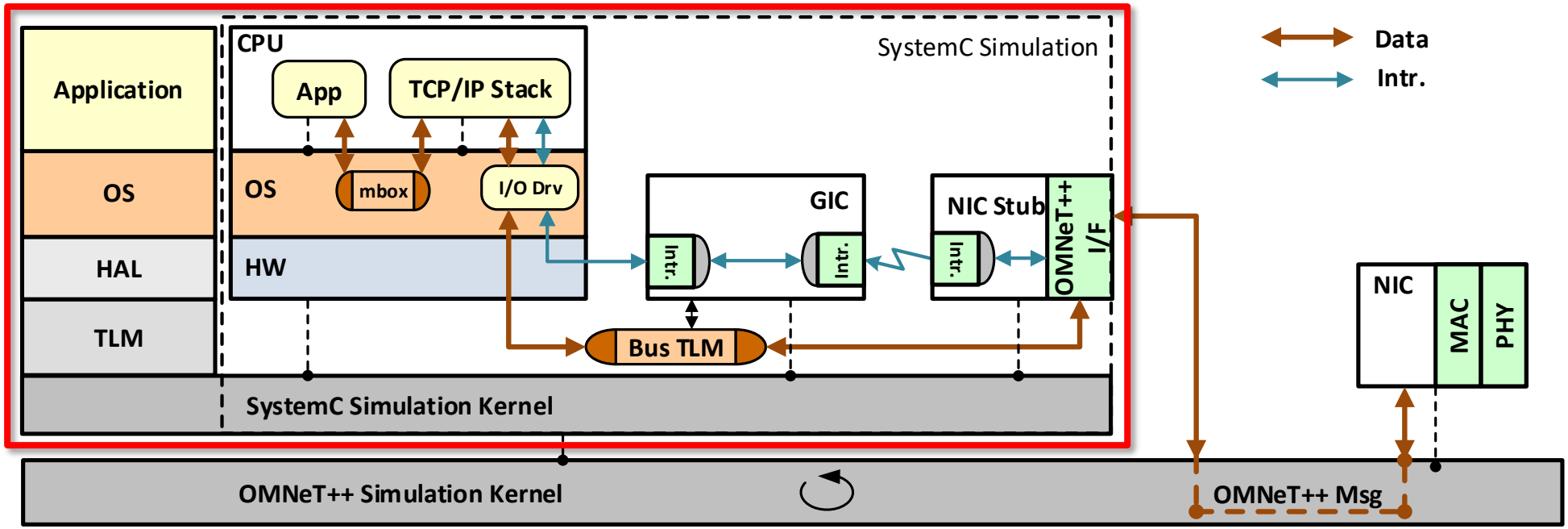
- **System simulation model**
  - SystemC-based host-compiled device model [Razaghi'14]
  - Capture system-wide interactions between application, OS and underlying hardware components
- **Network simulation backplane**
  - OMNeT++/INET network simulation framework



System Simulation Model

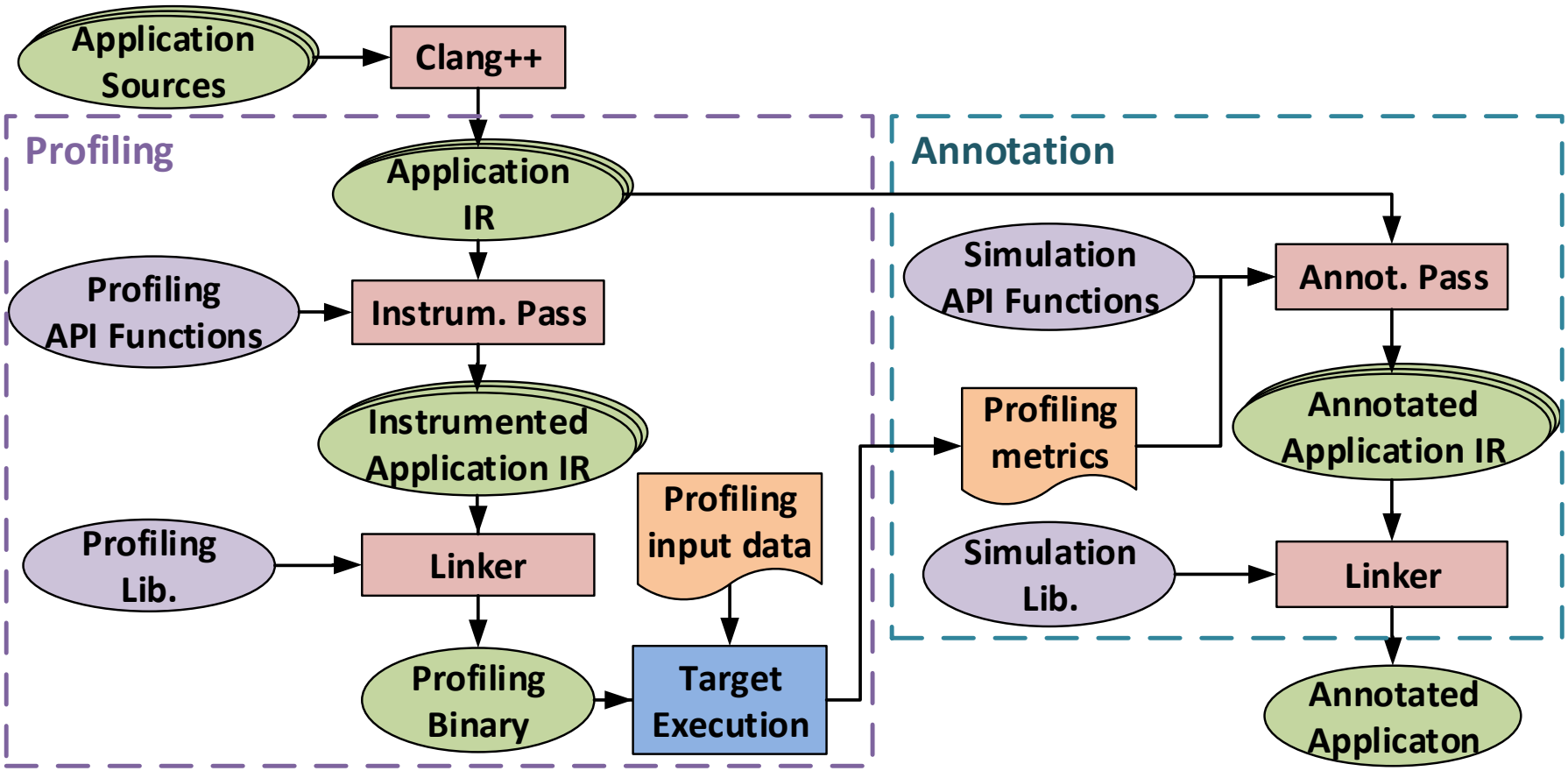
- **Host-compiled (HC) system simulator**
  - Source-level back-annotated application model
  - Network stack model [lwIP]

  Our work
  - Abstract multi-core OS and processor model [Razaghi'14]
  - Network interface & hardware peripheral models
  - Transaction-level modeling (TLM) base [SystemC]



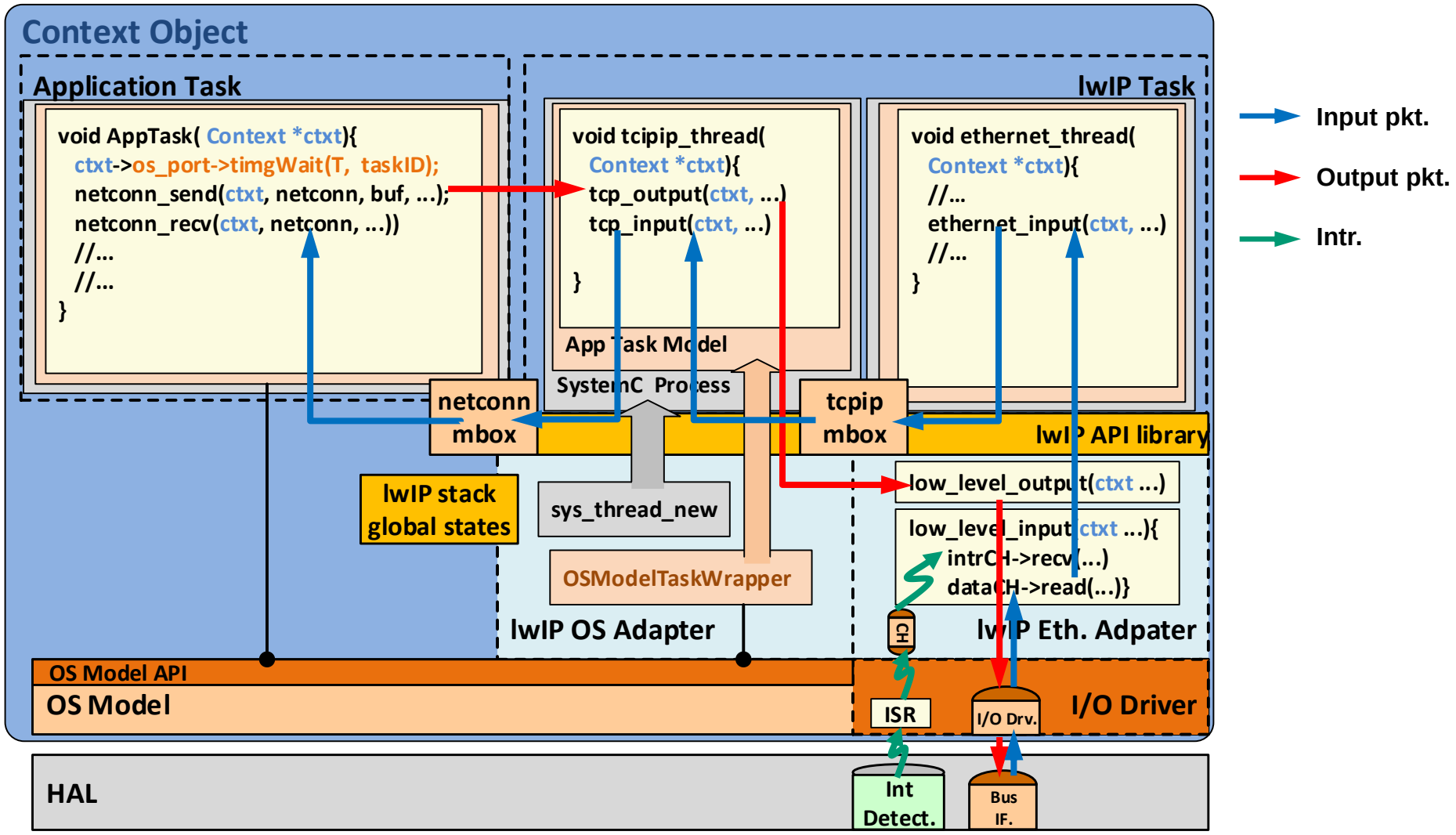
Application Task Model

- **Function-level performance back-annotation**
  - Function-level profiling/annotation pass [LLVM]
  - Back-annotation with average per-function execution time



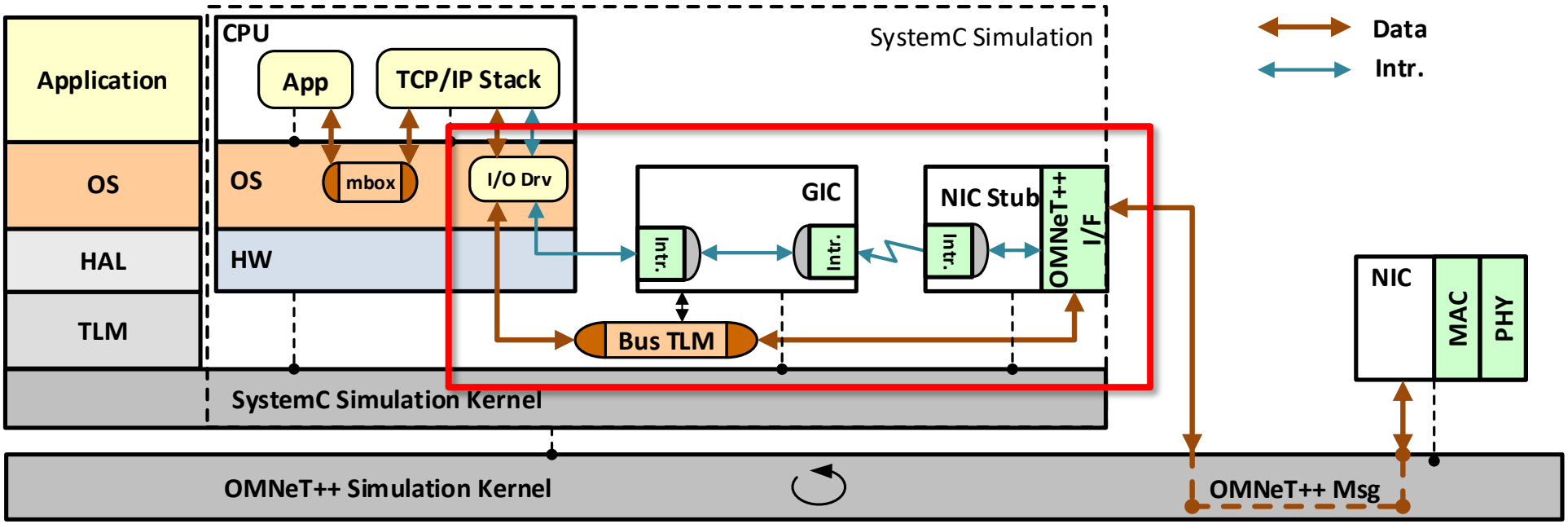
Network Stack Model

- Open source lightweight TCP/IP stack [lwIP]



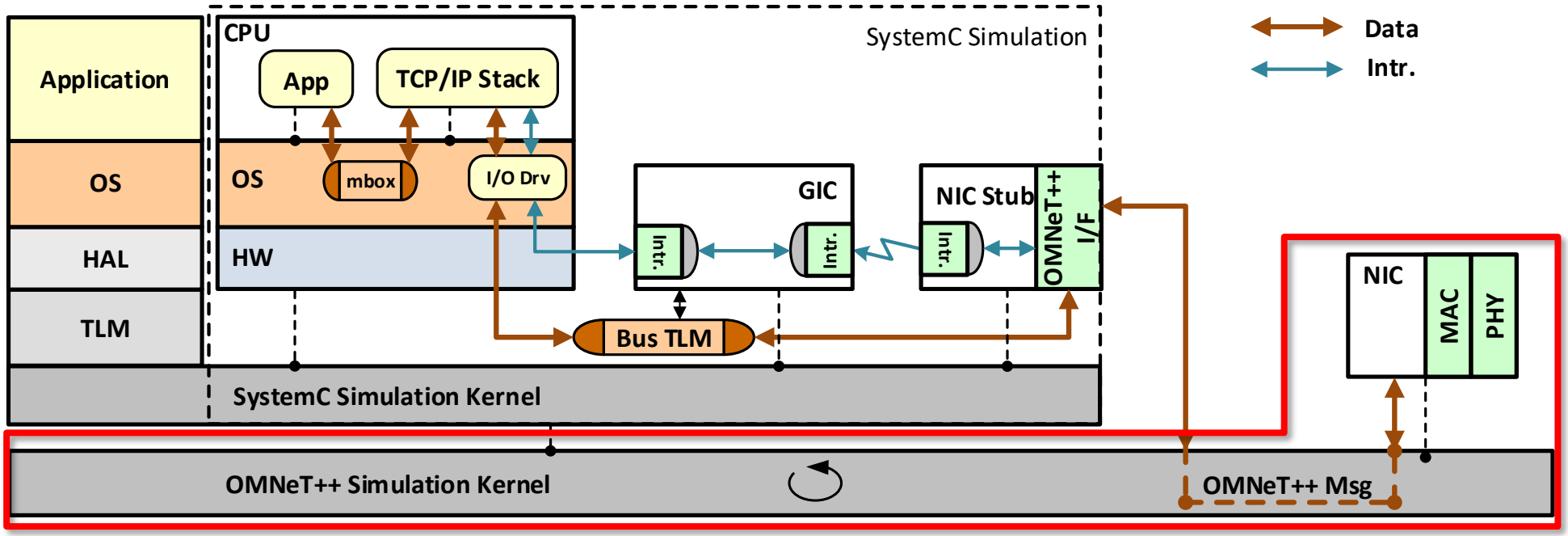
Network Interface Model

- **SystemC network interface card (NIC) stub model**
  - Interfacing with detailed NIC model in OMNET++
- **Generic interruption controller (GIC) model**
  - Distribute interruption signals from NIC upon packet arrival
- **NIC driver**
  - Interrupt service routine (ISR) triggered by GIC
  - Notify lwIP to read network packet data from NIC through bus TLM



Network Simulation Backplane

- **Detailed NIC model in OMNeT++**
  - Media access (MAC) and physical (PHY) layer simulation
  - Interfacing with NIC stub through OMNeT++ message
- **SystemC/OMNeT++ integration**
  - Scheduled and synchronized globally by OMNeT++
  - Multiple device instances in given network topology

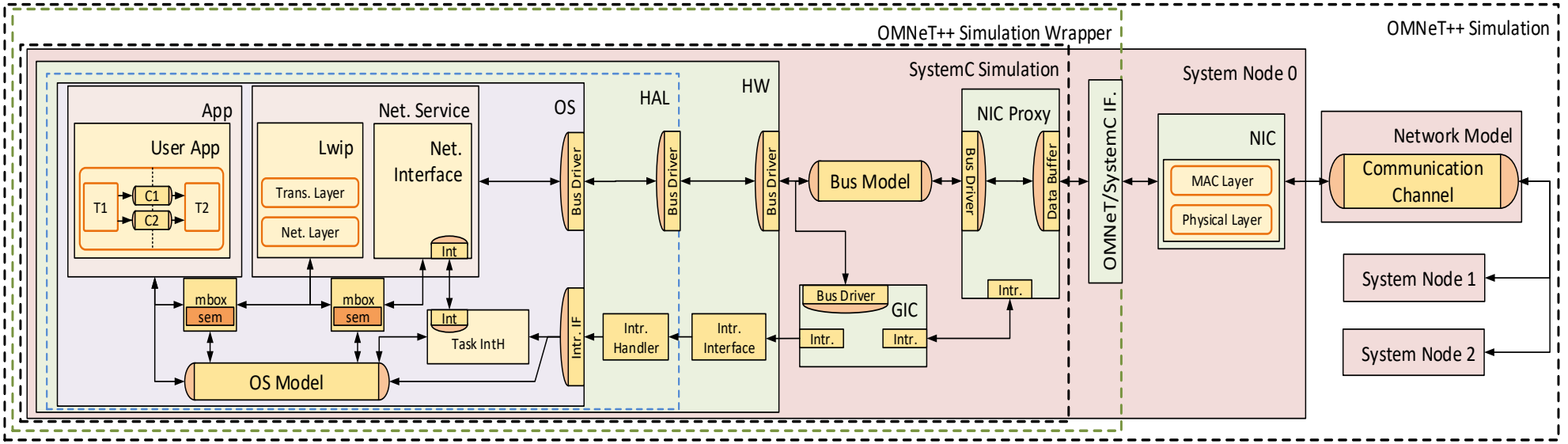


Experimental Setup

- **IoT application case studies**
  - Vision graph discovery
  - ECG diagnosis
- **Target platform**
  - Raspberry Pi 3 with 1.2 GHz ARM Cortex A-53
  - Raspberry Pi 0 with 1.0 GHz ARM11
- **WLAN client-server topology**
- **Design space exploration parameters**
  - Application task offloading levels
  - System configurations
    - Single core client (C:1×)
    - Single/dual core server (S:1×, S:2×)
    - Device type: Pi0, Pi3, Pi4 (Pi3 with twice the frequency)
  - 802.11 WLAN (b@11Mbps, g@9Mbps, g@54Mbps)

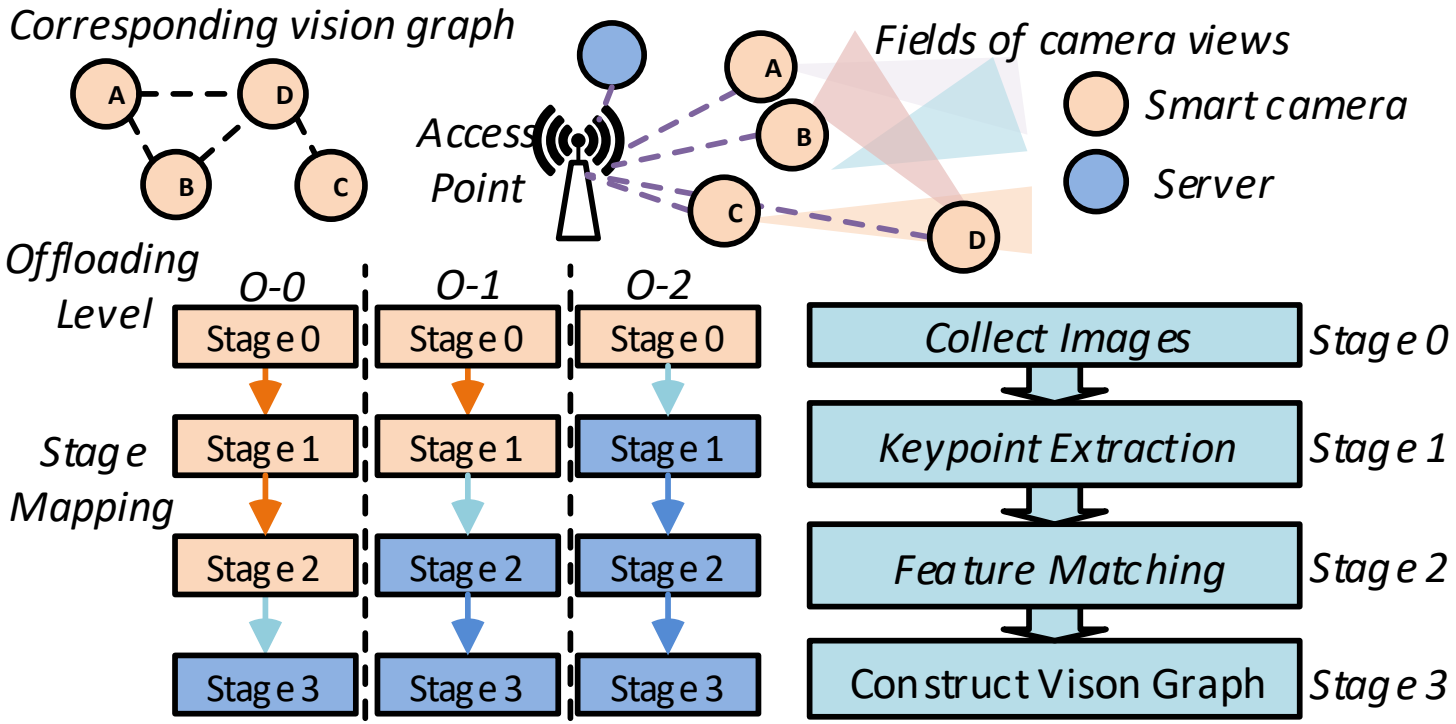
NoS Simulation Setup

- **SystemC system instances**
  - Profiling using HW counters [PAPI]
  - FIFO scheduler in OS model
- **INET/OMNeT++**
  - Network-related timing information
  - Star-like client-server topology



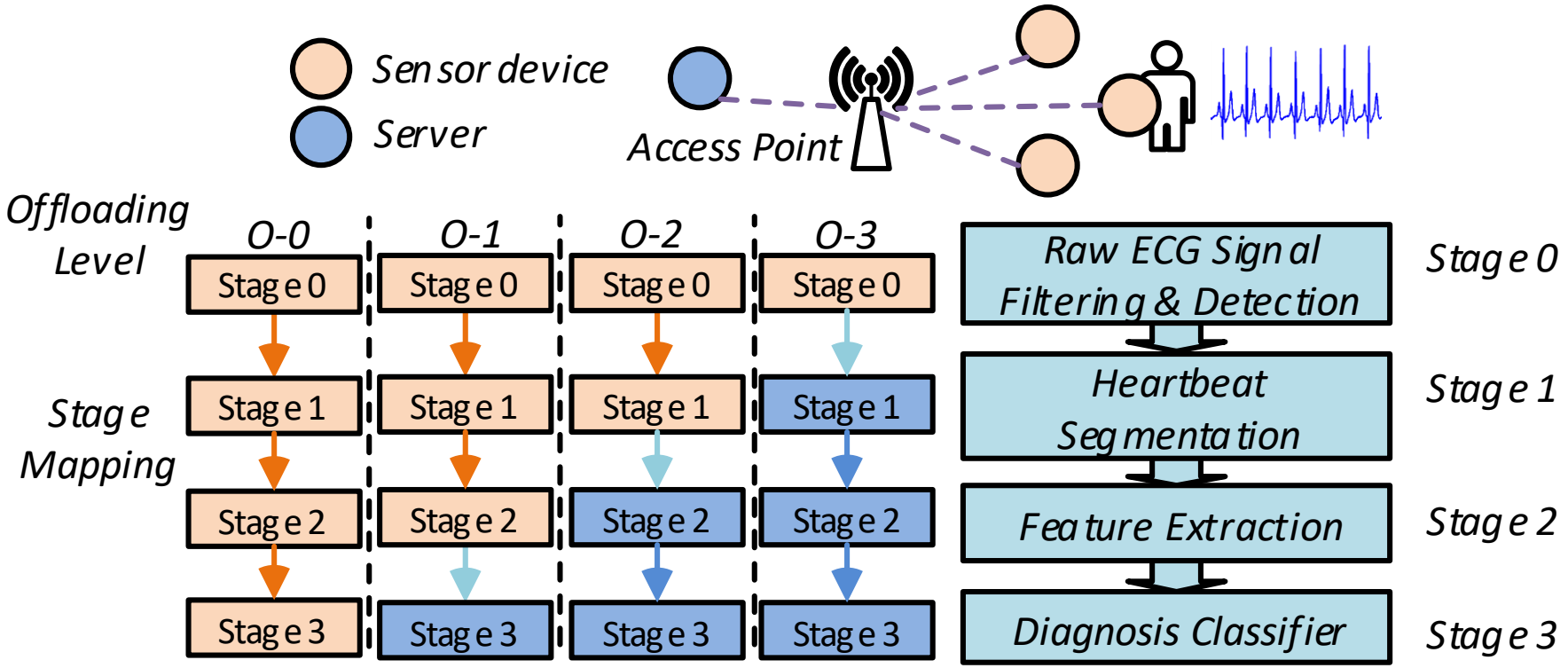
Vision Graph Discovery

- **Vision graph discovery in smart camera networks**
  - Discover view overlaps among a set of cameras
  - 4 execution stages, 3 offloading levels, 2-6 clients
  - Implemented in C++ with OpenCV
  - Performance metrics: throughput, client core utilization



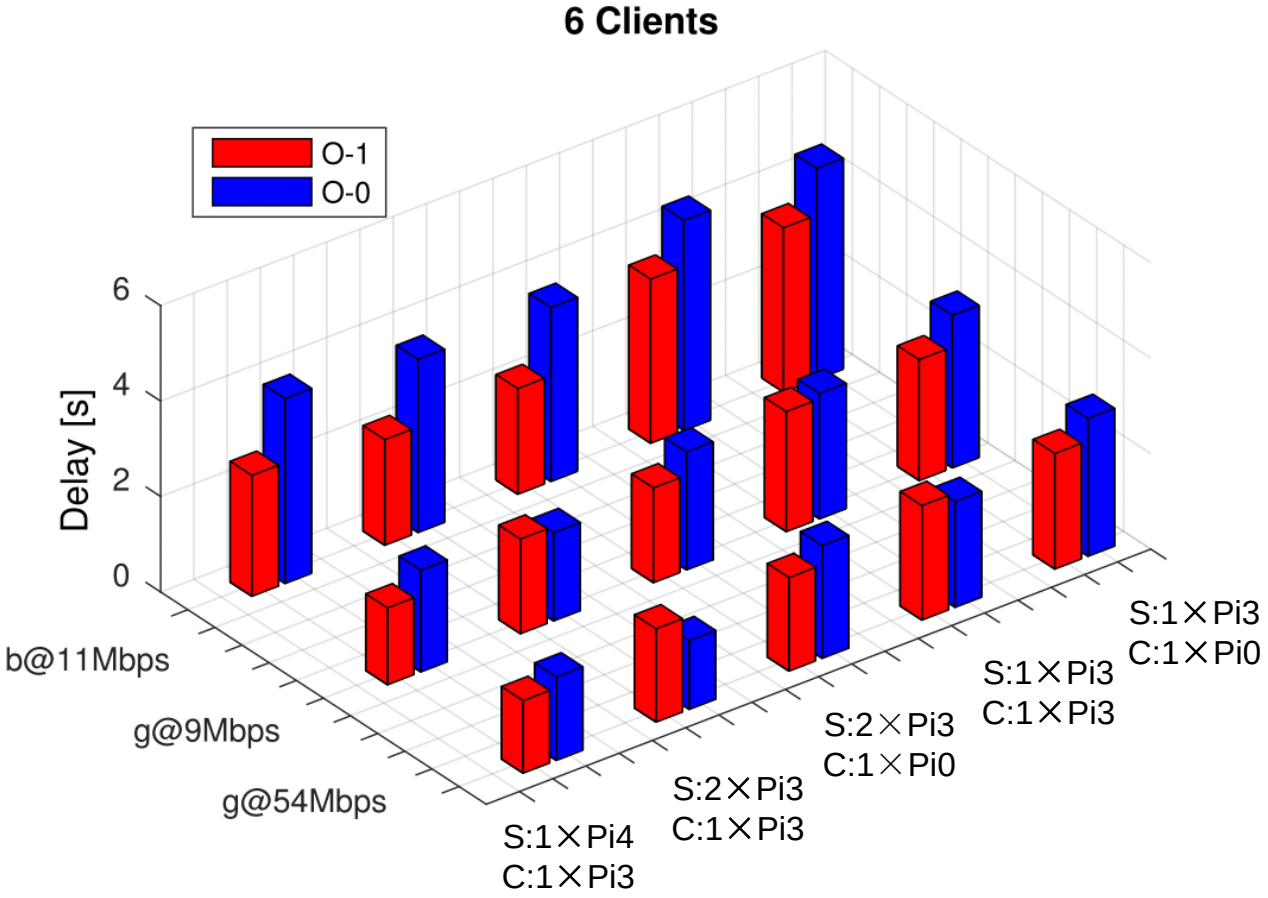
ECG Diagnosis

- **ECG monitoring**
  - Recognize abnormal heartbeat from raw ECG signals
  - 4 execution stages, 4 offloading levels, 2-6 clients
  - Implemented in C
  - Performance metrics: throughput, client core utilization



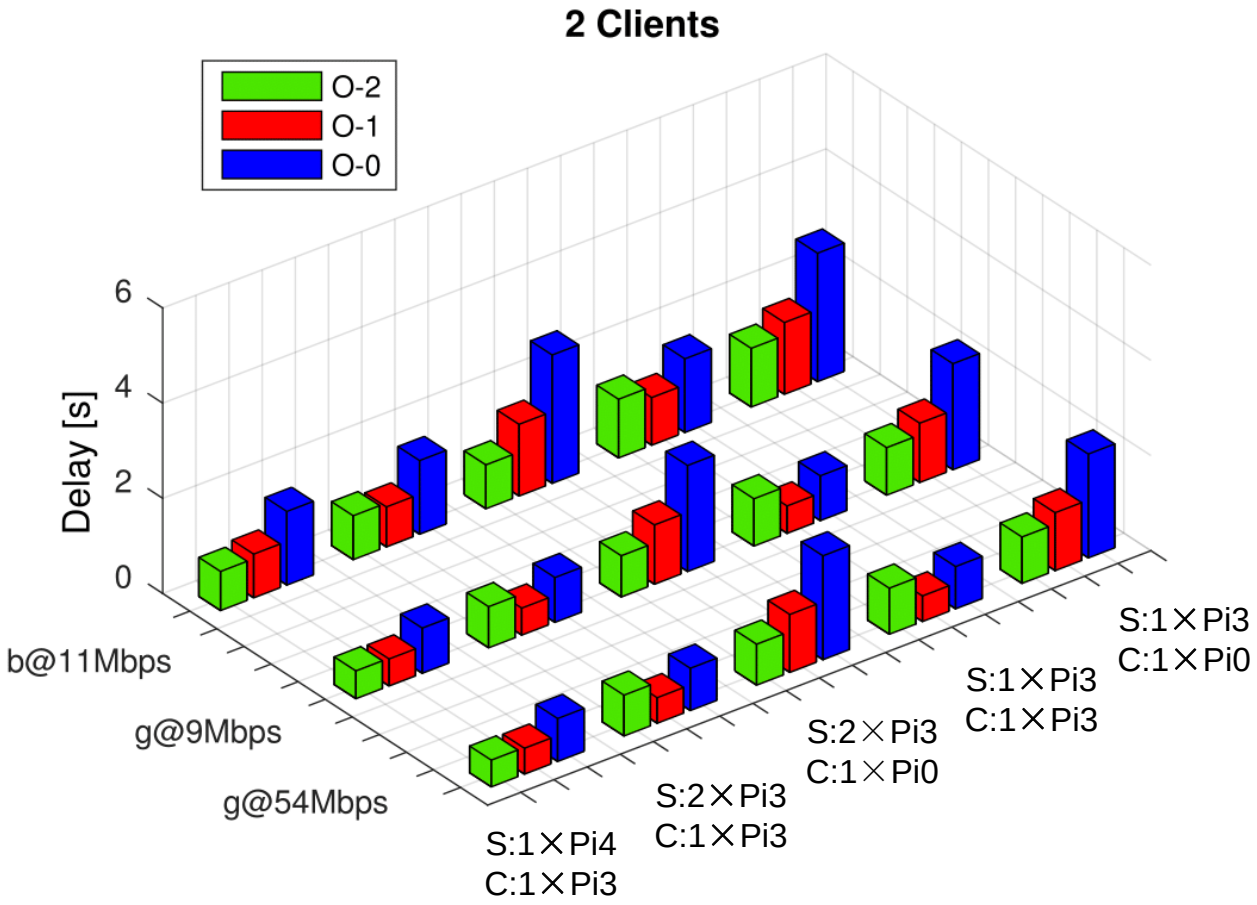
Vision Graph Discovery Throughput

- **Output-to-output delay with 6 clients**
  - Server- and computation-bound at O-2
  - Balanced at O-0 and O-1



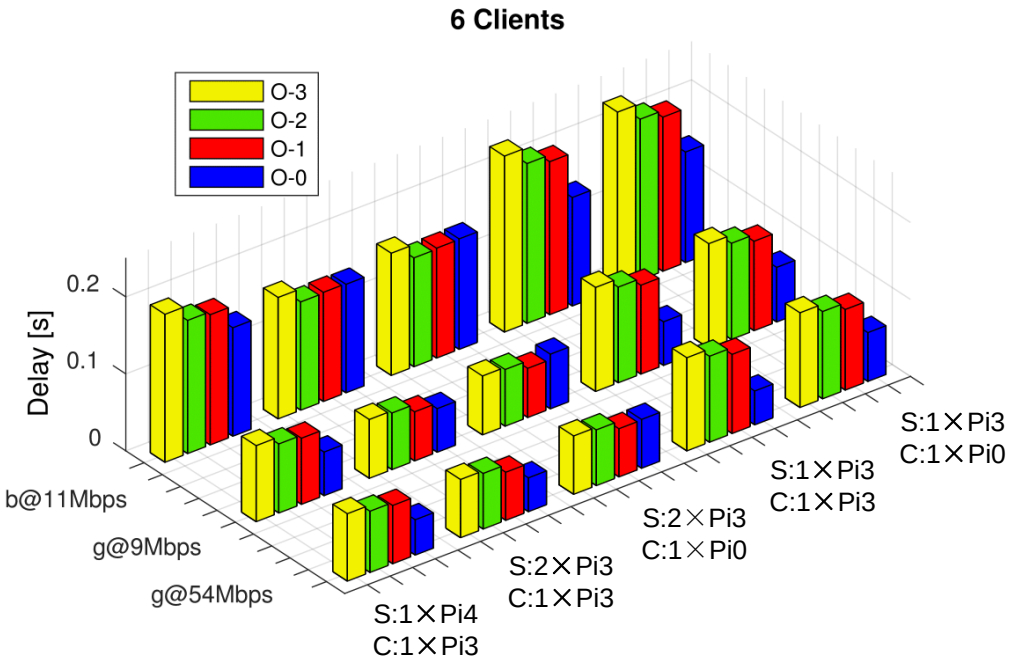
Vision Graph Discovery Throughput

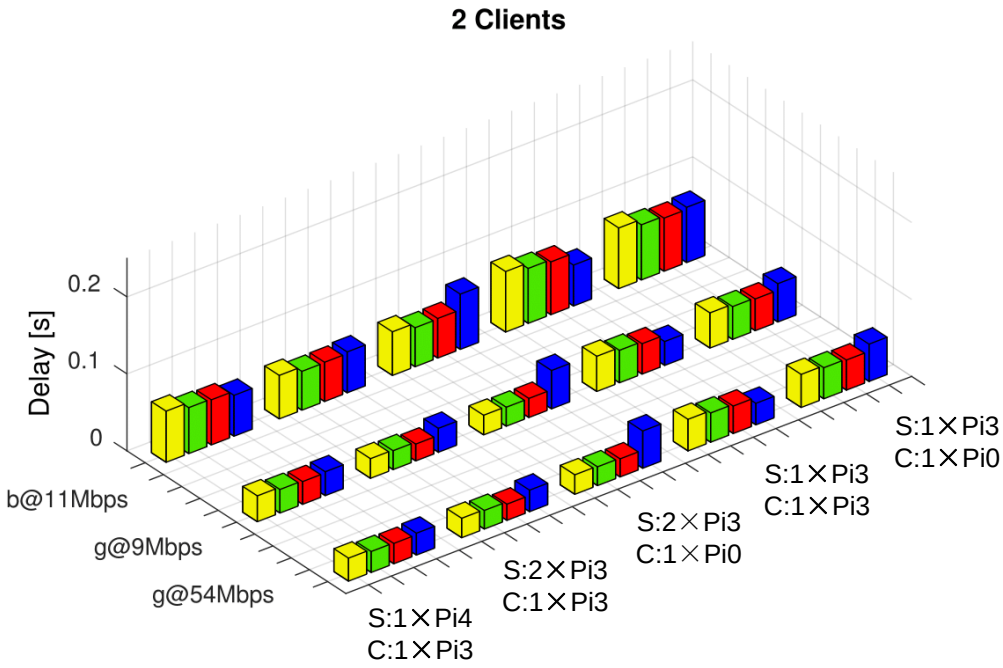
- **Output-to-output delay with 2 clients**
  - Computation and communication balanced
  - Client-bound at O-0 and O-1



ECG Diagnosis Throughput

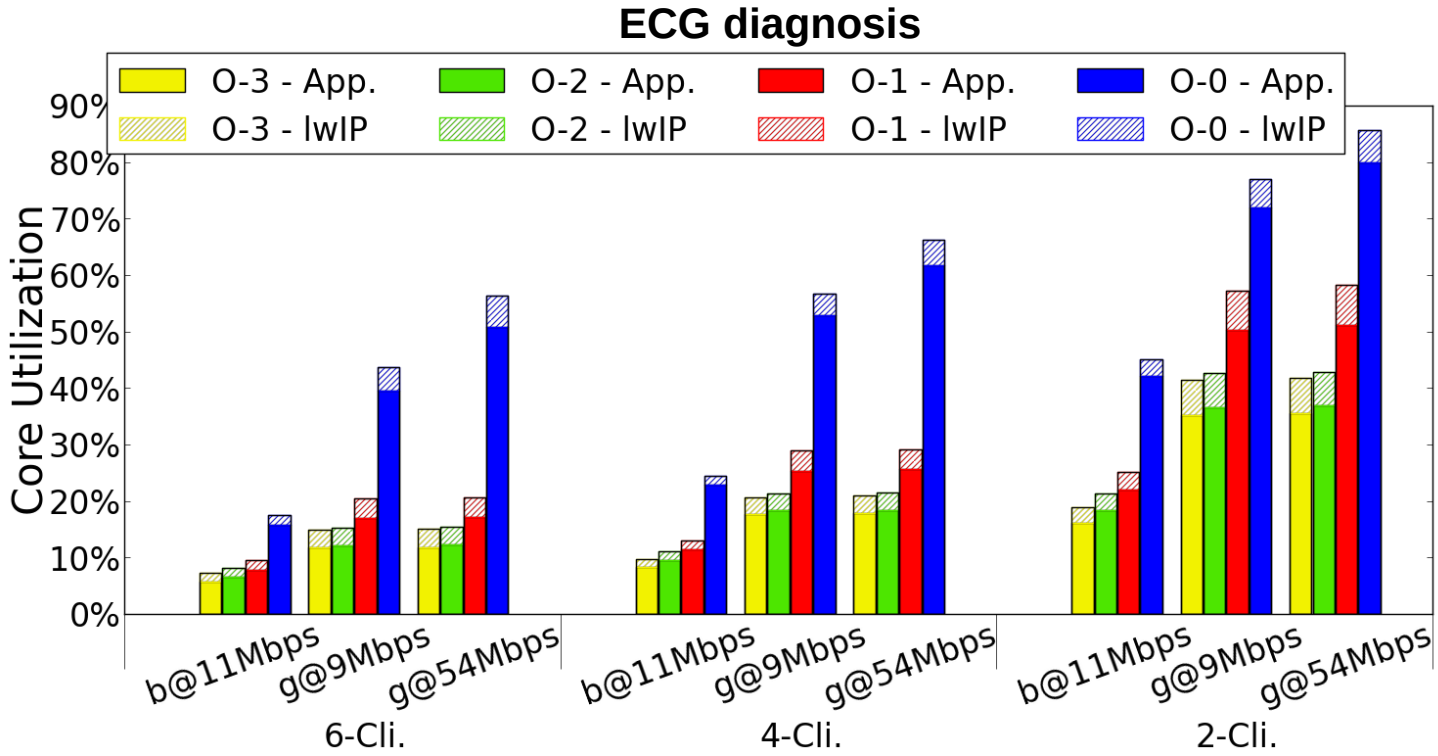
- **Output-to-output delay with 2 and 6 clients**
  - Always computation and communication balanced
  - Server-bound at O-1,2,3
  - Client-bound at O-0





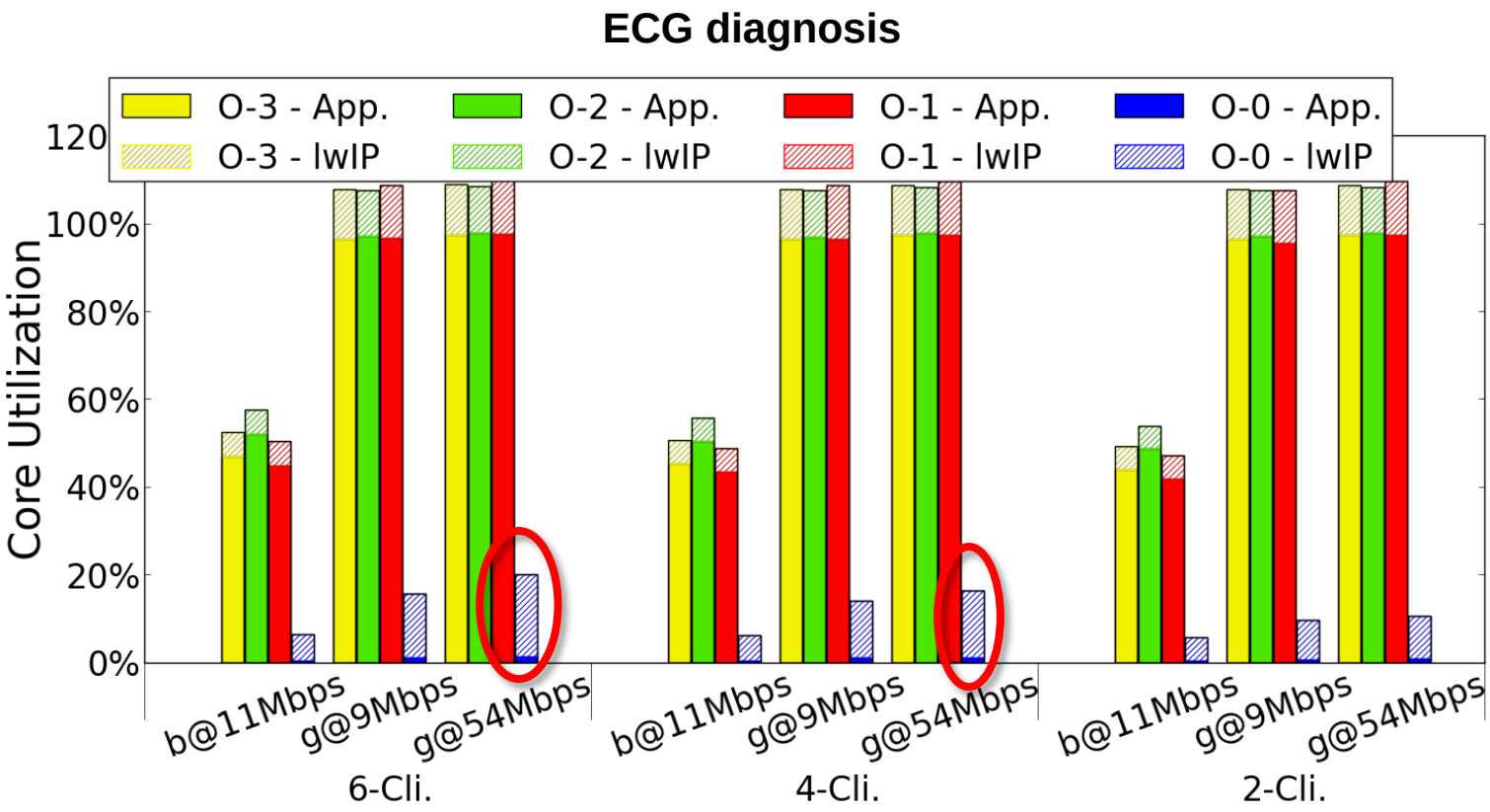
Client Core Utilization

- **Mainly affected by offloading level and client count**
  - Lower offloading increases client load
  - Lower client count increases throughput
- **Network configuration impact**
  - ECG diagnosis: 13.7% - 31.6%
  - Vision graph discovery: 11.5% - 19.5%



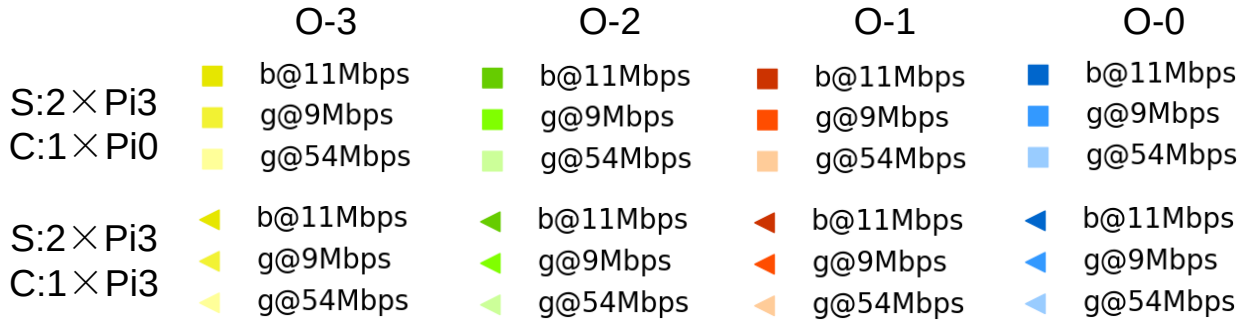
Server Core Utilization

- **Mainly affected by offloading level**
  - Higher offloading increases server load
  - Client count affects throughput and server load
- **Up to 20% core utilization for lwIP stack**

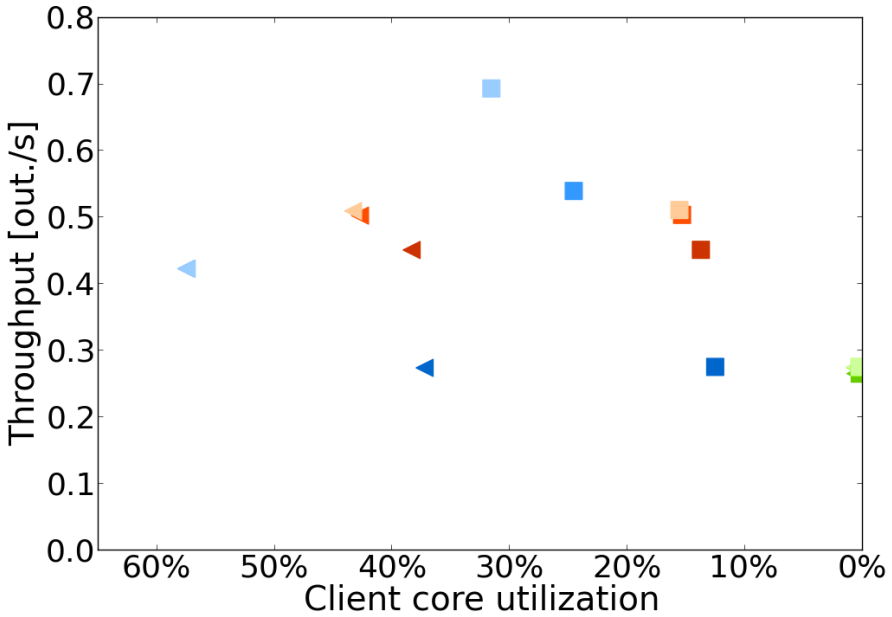


NoS Design Space

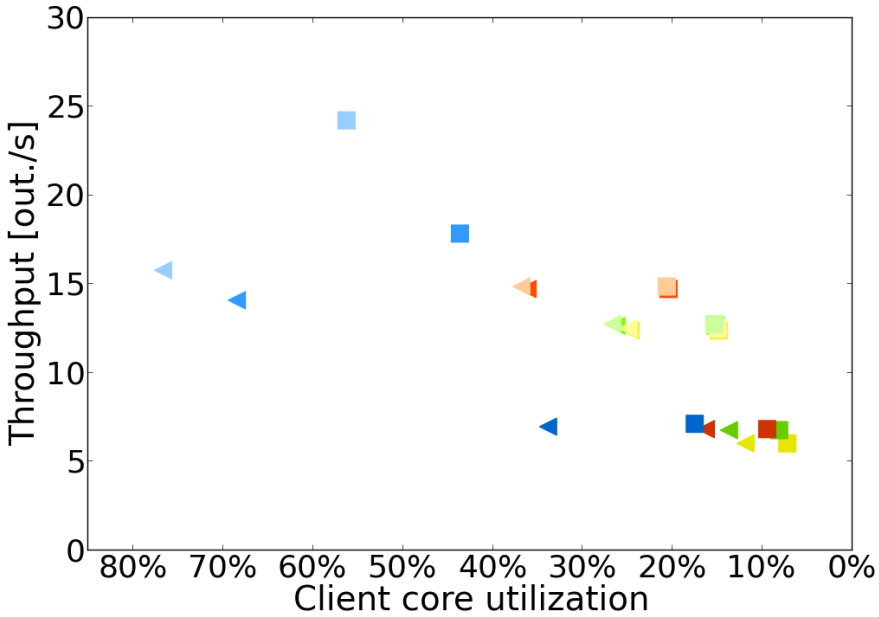
- **IoT design space exploration**
  - 6-client setup with different app., system and net. config.
  - ➢ Non-obvious parameter interactions



Vison graph discovery

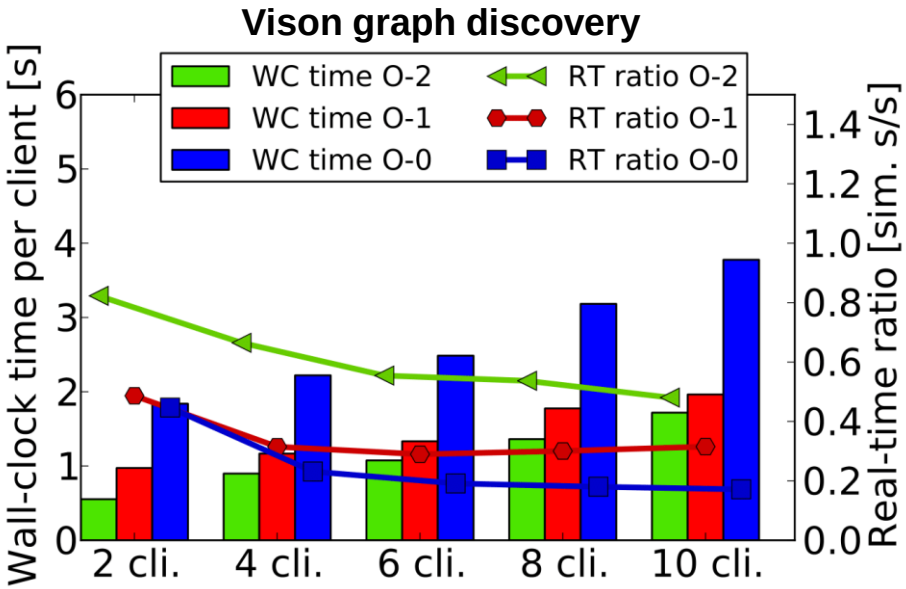


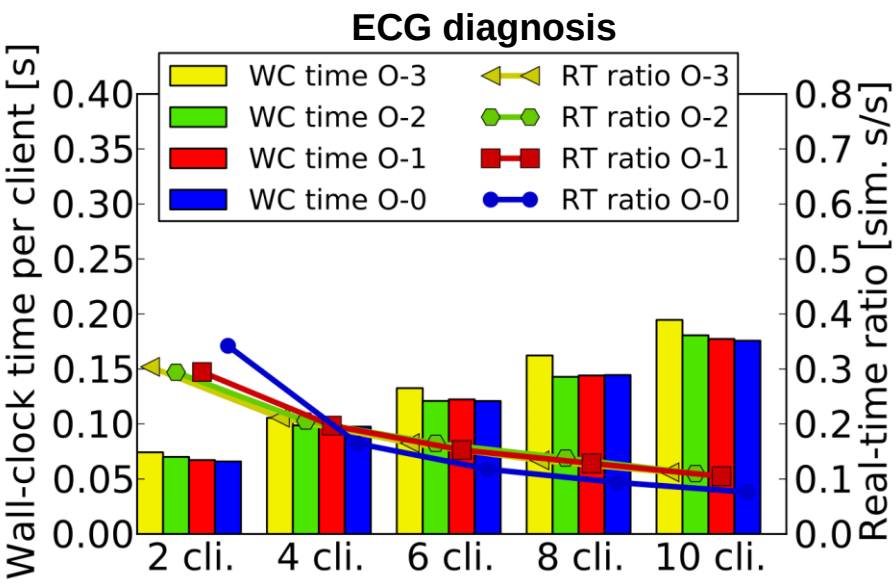
ECG diagnosis



Simulation Speed and Accuracy

- **Simulation speed**
  - Vision graph discovery: 0.39 sim. seconds/s
  - ECG: 0.18 sim. seconds/s
  - ➢ Larger communication requires more simulation events
- **Simulation accuracy**
  - >95% for system simulator





Summary & Conclusions

- **Networks-of-Systems (NoS) simulator**
  - SystemC based host-compiled system model
  - INET/OMNeT++ integration
  - ➢ Fast, comprehensive and flexible NoS simulator

- **NoS design space exploration**
  - IoT application case studies
  - ➢ Significant network/system interactions

- **Future work**
  - Accuracy validation against more real world setups
  - Various metrics such as quality and reliability
  - Network/system co-optimization