Securing the Smart Grid with Software Defined Networking

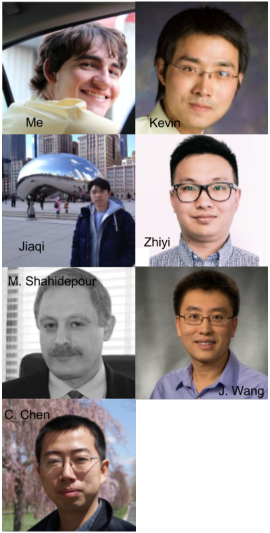
Christopher Hannon

Monday 08-28-2017

Outline

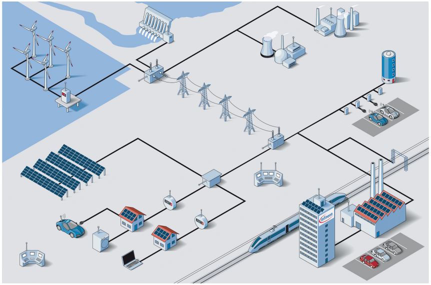
Christopher Hannon

- 3rd year Direct Ph.D Student
- BS CS from IIT in 2015
- Advisor: Kevin Jin
- Collaborators: Jiaqi Yan, Zhiyi Li, M. Shahidepour, J. Wang, C. Chen



Background - Smart Grid

- Utilization of communication network to improve power network
  - Reliability of power
  - Resiliency to failure and attack
  - Flexibility in managing the power grid
  - Efficiency of power usage
- Generation
- Transmission
- Distribution



Background - What Led to SDN

- Software Defined Networking (SDN)
  - What is Software Defined Networking?
  - Why did SDN emerge?
  - Why can SDN be useful for Smart Grid?

Background - First Networks

- Telephone networks
- Circuit switching
- OSI Layer 1
- Time-division multiplexing (virtual circuit switching)

Background - Networking

- Internet emerges
- Subnets
- Routing
- OSI Layer 3

Background - Growth of Networking

- Local Area Networks (LANs)
- Star shaped networks
  - Too complicated
  - Buses and broadcast domains
- Mac addresses (precurser)
  - 8 bit manually set
- OSI Layer 2

Background - Networking

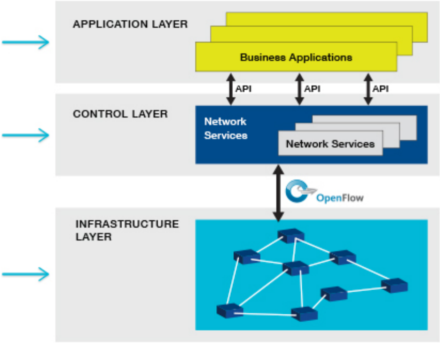
- Large networks
  - Big companies and universities
  - LANs get bigger
- Multiple interconnected buses
  - Single bus is bottleneck
  - Switching and bridging is needed

Background - Why SDN?

- Switching gets really complicated!
  - Spanning tree protocol
  - Layer 2 can not scale to Layer 3
  - DHCP, ARP, layer 2 to layer 3 interfaces+
- Middleboxes
  - Intrusion detection
  - load balancing
  - etc.
- Data Centers!

Background - Software Defined Networking (SDN)

- Enables Easier Control of Communication Network
- Separates the:
  - Data plane - performs packet forwarding according to rules
  - Control plane - makes decisions on how to forward packets
- Provides
  - Centralized management
  - Global view of the network
  - Easier deployment
  - Greater flexibility in network

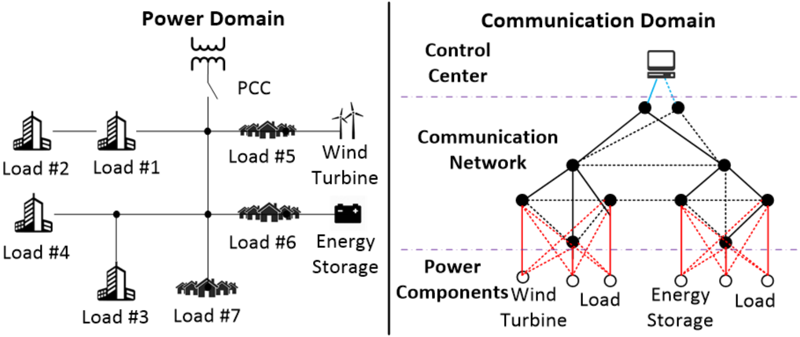


Background - Communication in Power Grids

Different applications have different requirements

- Power Grid Protection
  - Very low latency
  - High priority traffic
- SCADA
  - Millisecond latency
  - Medium priority
- Metering Infrastructure
  - Low priority
- PLC
- Wireless (radio/ cellular)
- Ethernet

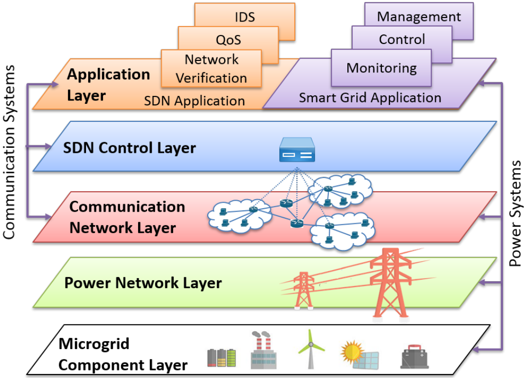
Example



Power Domain
PCC
Load #2
Load #1
Load #5
Wind Turbine
Load #4
Load #6
Energy Storage
Load #3
Load #7
Communication Domain
Control Center
Communication Network
Power Components
Wind Turbine
Load
Energy Storage
Load

Motivation - SDN in Smart Grid

- SDN can be used to create a reliable and resilient power system
- Enable Smart Grid applications to interact with networking and power systems



Motivation

- Real system has high overhead cost and time
- Simulation based tool that combines power system with communication network
- DSSnet - Distribution System Solver network
- Features
  - Power flow analysis
  - Software Defined Network communication network modeling
  - Smart Grid control applications
  - Cyber security evaluation

Combining Simulation and Emulation Systems for Smart Grid Planning and Evaluation

- C. Hannon, J. Yan, and D. Jin. DSSnet: A Smart Grid Modeling Platform Combining Electrical Power Distribution System Simulation and Software Defined Networking Emulation. 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Banff, CA May 2016. First Place IIT Research Day 2016

- C. Hannon, J. Yan, D. Jin, C. Chen, J. Wang. Combining Simulation and Emulation Systems for Smart Grid Planning and Evaluation. Transactions on Modeling and Computer Simulation.

- D. Jin, Z. Li, C. Hannon, C. Chen, J. Wang, M. Shahidehpour. Towards a Resilient and Secure Microgrid Using Software Defined Networking. IEEE Transactions on Smart Grid.

Related Work

- Existing tools
  - FNCS - (Distribution + Transmission + Communication)
  - GECO - Global Event Driven
  - EPOCHS - Agent Based (commercial software)
  - PSLF + ns-2
  - OpenDSS + HiL Testbed
- DSSnet (Distribution System Electric Power Simulation + Communication Network Emulation)

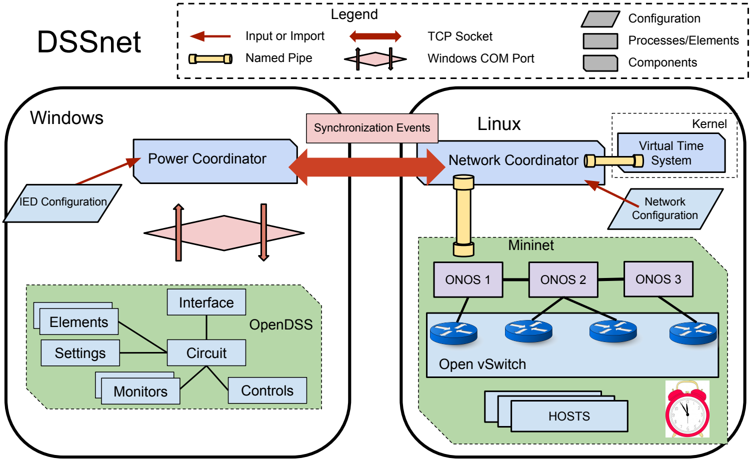
Design - Architecture

- Power Simulation
  - Simulates power flow
- Communication Network Emulation
  - Emulates power devices
    - Communication elements
    - Processing elements
- Emulation Coordinator
- Power Coordinator
- Virtual Time Module
  - Supports synchronization

Background - Simulation & Emulation

- Emulation
  - Virtual machine/ container based emulation
  - Processes execute instructions to advance the clock
  - Inherently continuous
  - Mininet network emulator
    - Hosts & switches
- Simulation
  - Mathematical model of system
  - Solved at timesteps
  - Event queue
    - Clock advances to next event in queue
    - Runs as fast a possible
  - OpenDSS power simulator

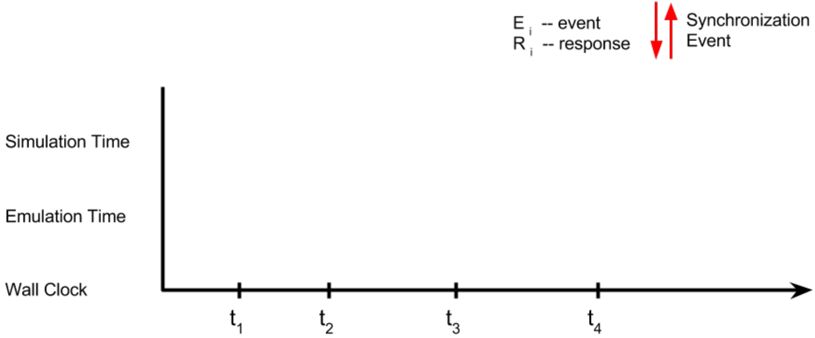
Design - Architecture



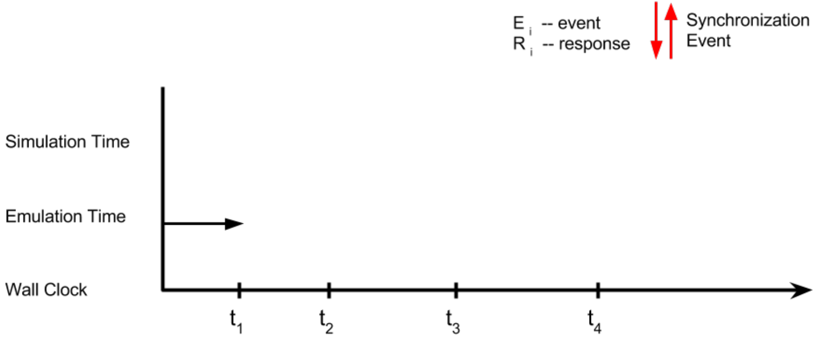
Challenges

- How can the power Simulator be combined with communication network emulator?
  - Emulation is continuous
  - Simulation is solved in discrete time steps (not real time)
- Synchronization is a challenge

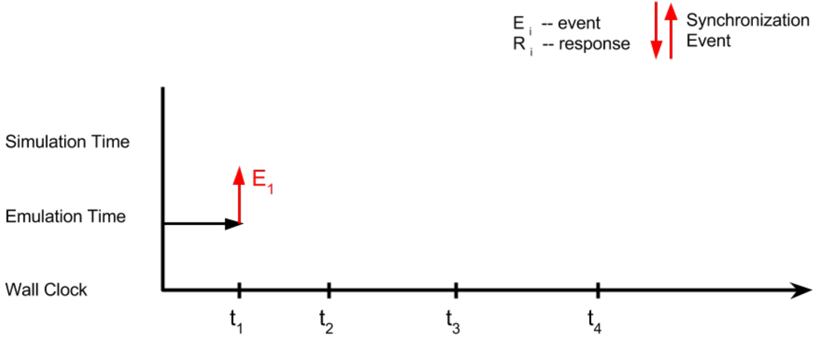
Design - Synchronization



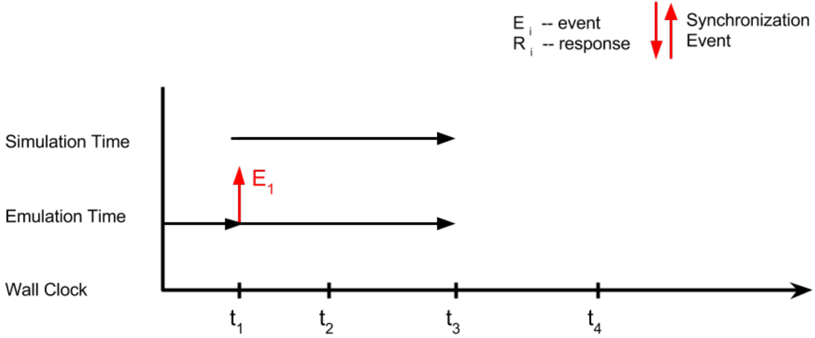
Design - Synchronization



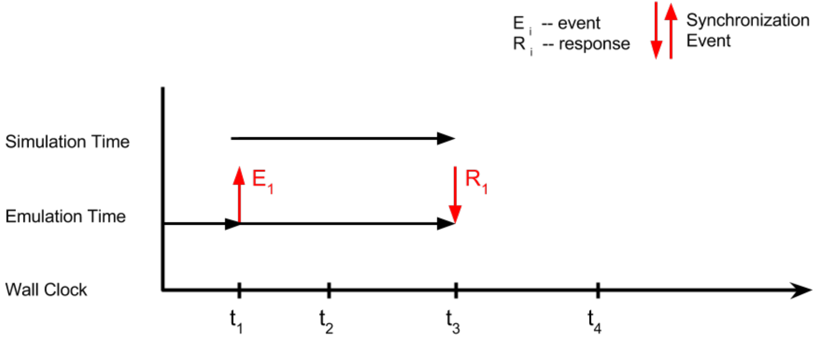
Design - Synchronization



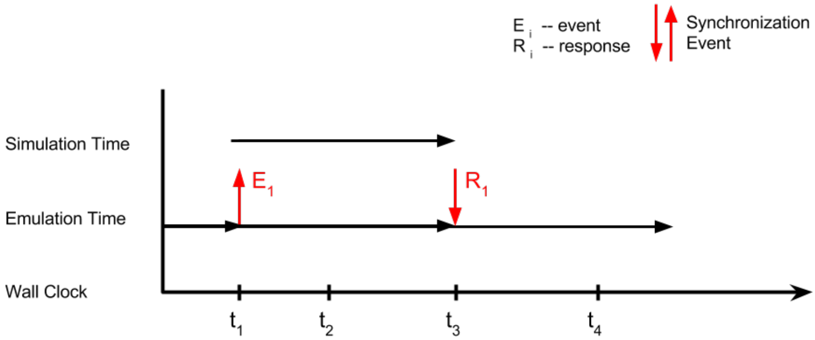
Design - Synchronization



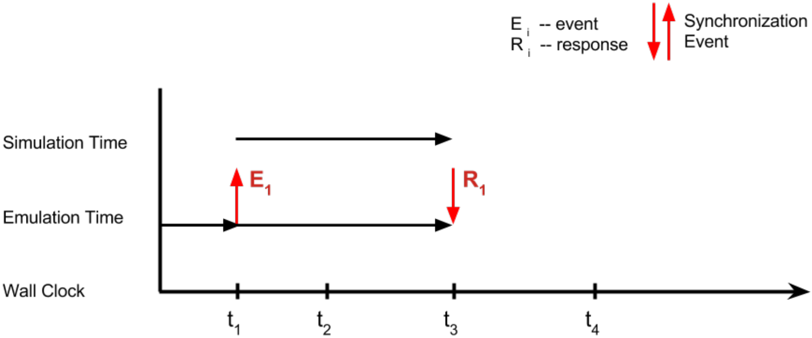
Design - Synchronization



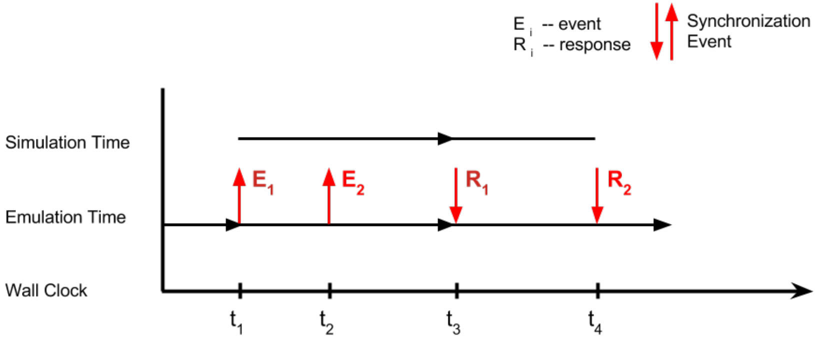
Design - Synchronization



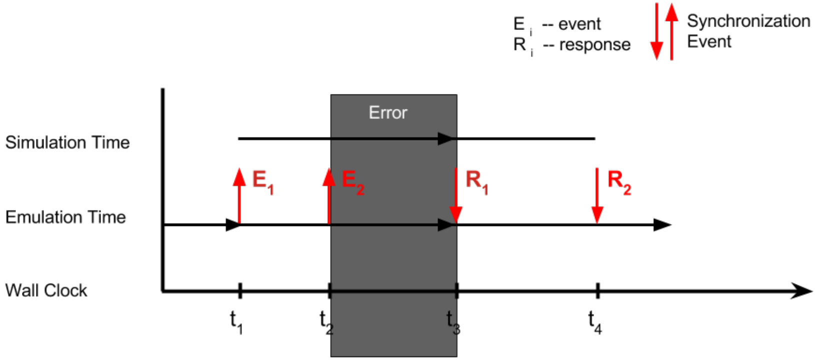
Design - Synchronization



Design - Synchronization

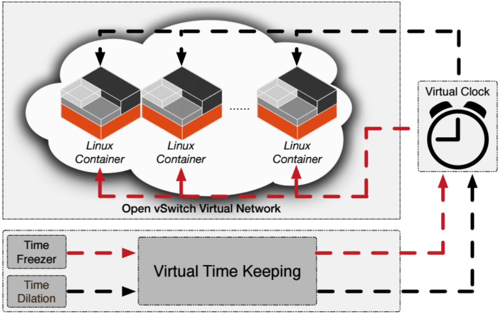


Design - Synchronization

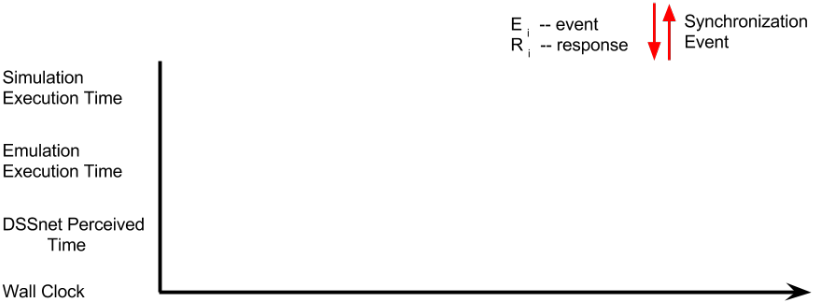


Virtual Time

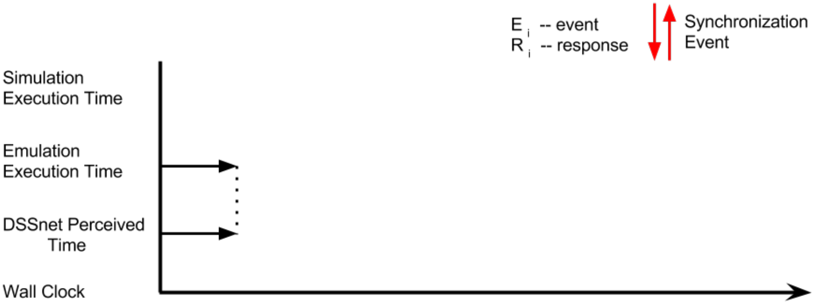
- Each process in the emulation has its own clock
- Virtual time module has the following abilities
  - add processes to virtual time
  - pause processes
  - resume processes



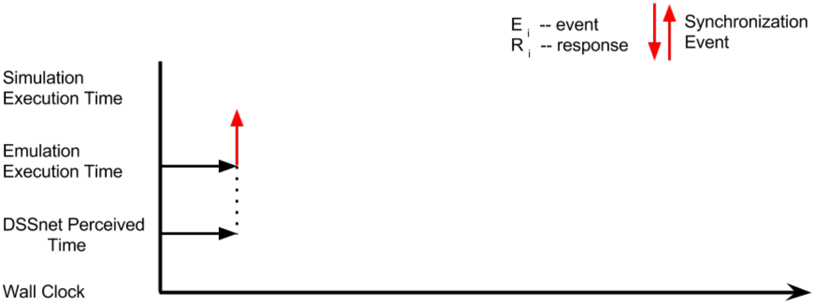
Design - Synchronization



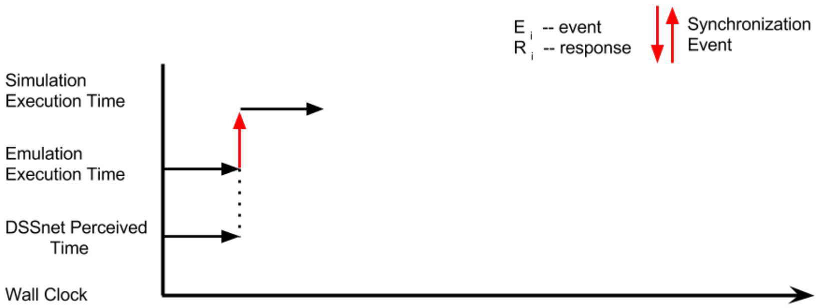
Design - Synchronization



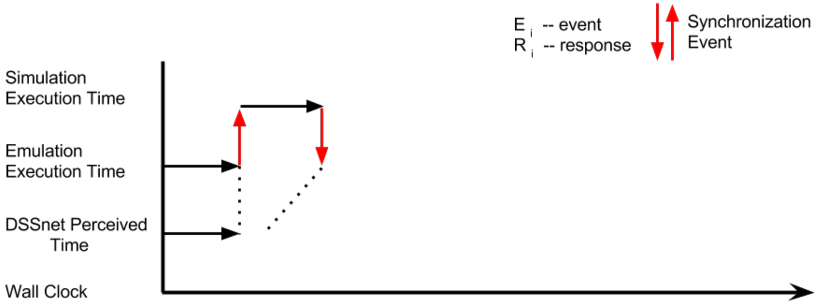
Design - Synchronization



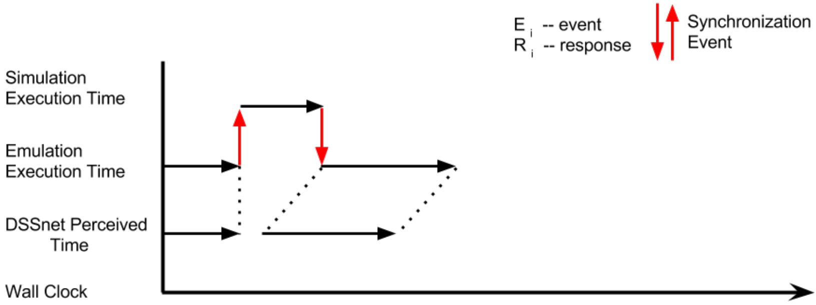
Design - Synchronization



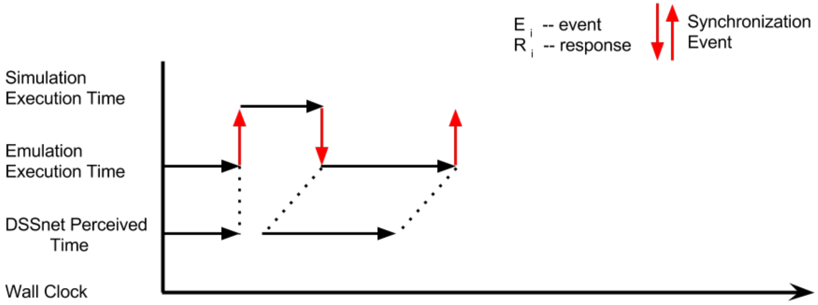
Design - Synchronization



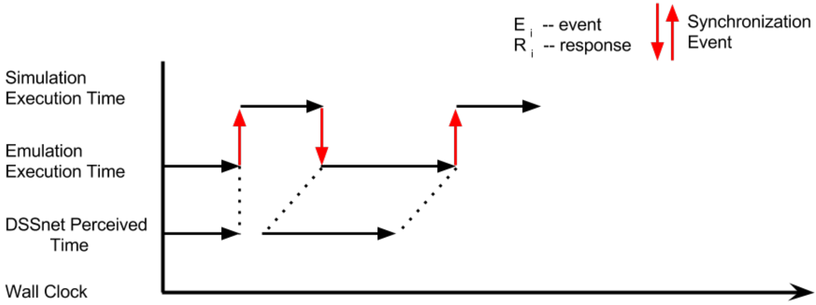
Design - Synchronization



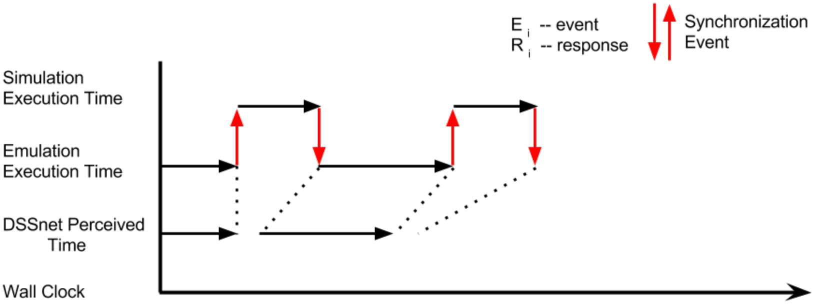
Design - Synchronization



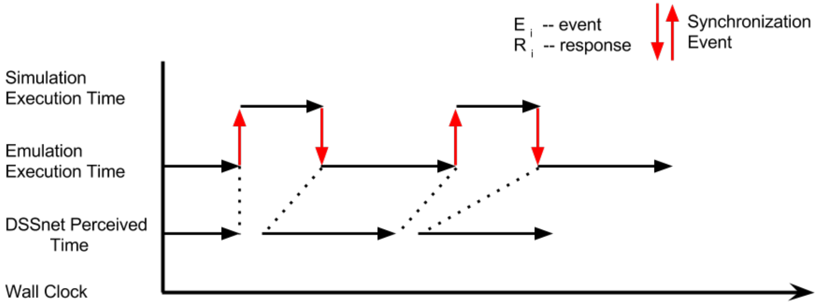
Design - Synchronization



Design - Synchronization



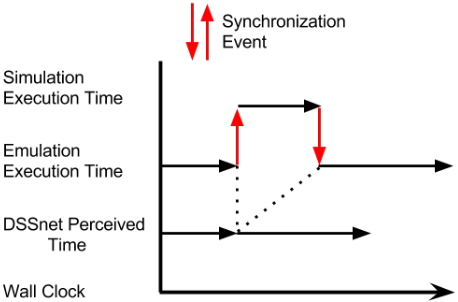
Design - Synchronization



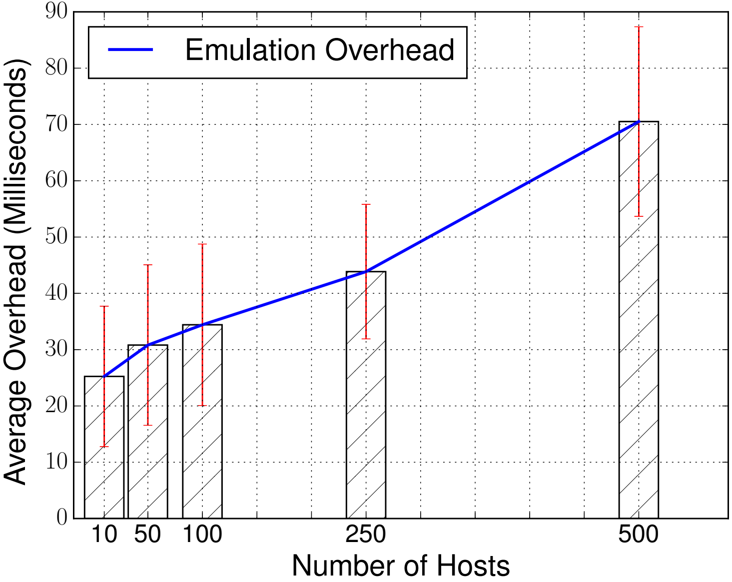
Design - Synchronization

- $$Time_{wall\_clock} = \sum t_{E\_i} + \sum t_{S\_i}$$
- $$Time_{DSSnet} = \sum t_{E\_i} + \sum t_{S'\_i}$$
- $$ret = \frac{t_{S'\_i}}{t_{S\_i}}$$
  - $(1,\infty)$
  - $(0,1]$
  - 0

Design - Synchronization



Evaluation

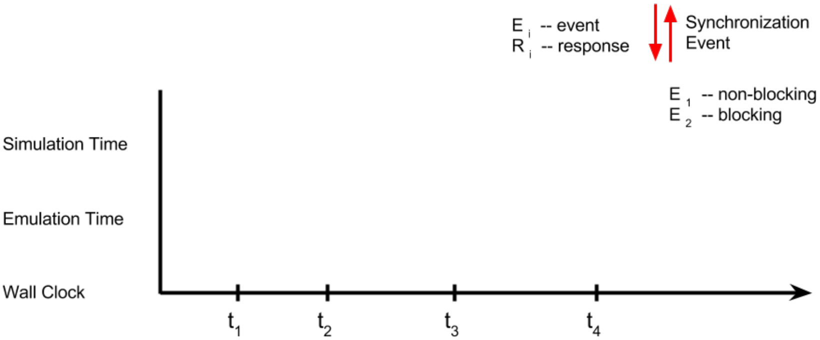


Emulation Overhead
Average Overhead (Milliseconds)
0
10
20
30
40
50
60
70
80
90
10
50
100
250
500
Number of Hosts

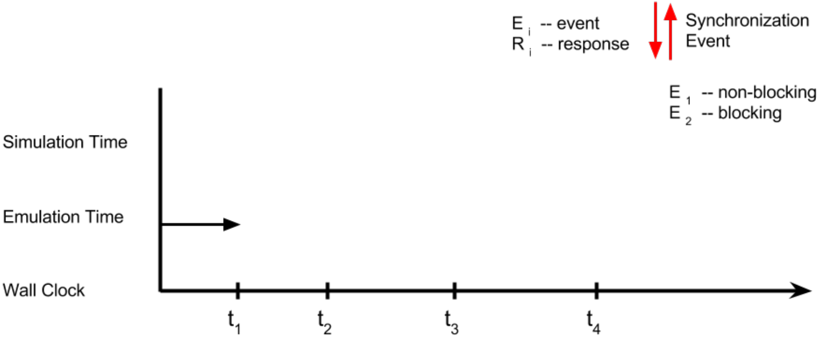
Design - Parallel Execution

- Minimize pause and unpause
- Multiple event queue design
  - Blocking (forces pause)
  - Non-blocking (enables the systems to run in parallel)

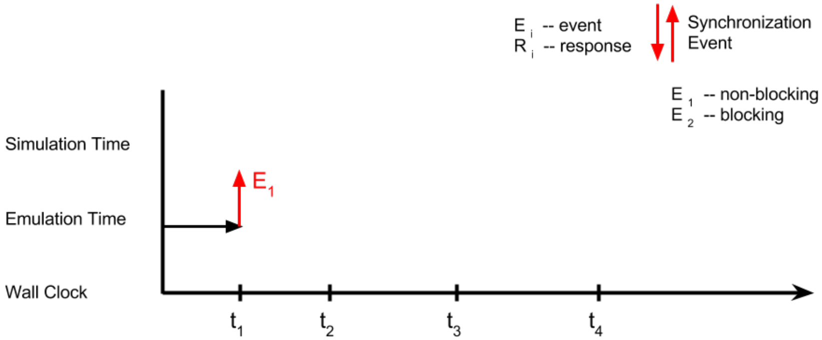
Design - Parallel Execution



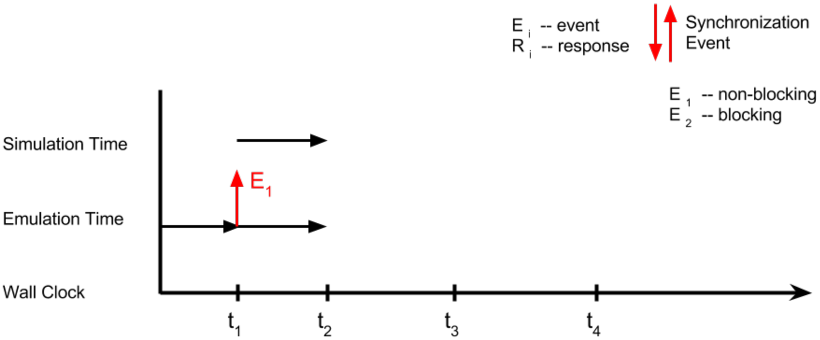
Design - Parallel Execution



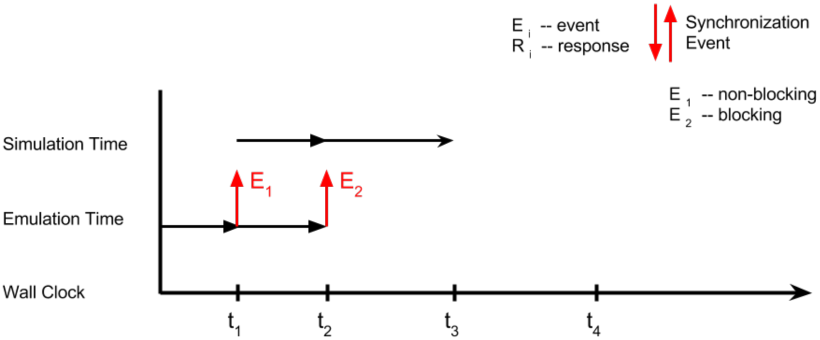
Design - Parallel Execution



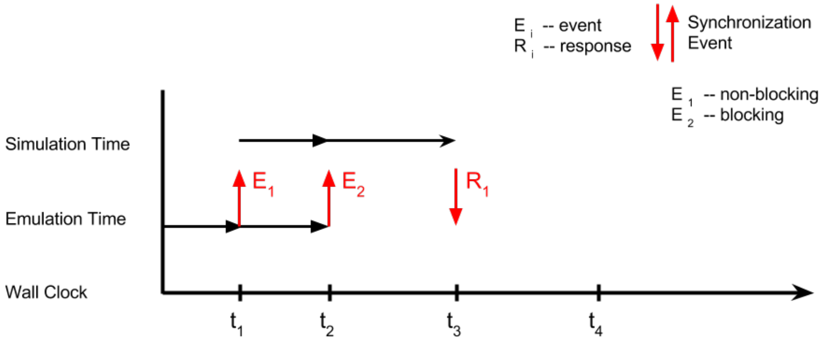
Design - Parallel Execution



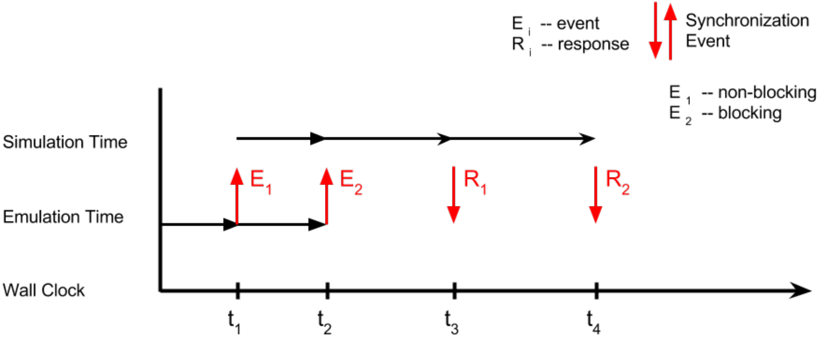
Design - Parallel Execution



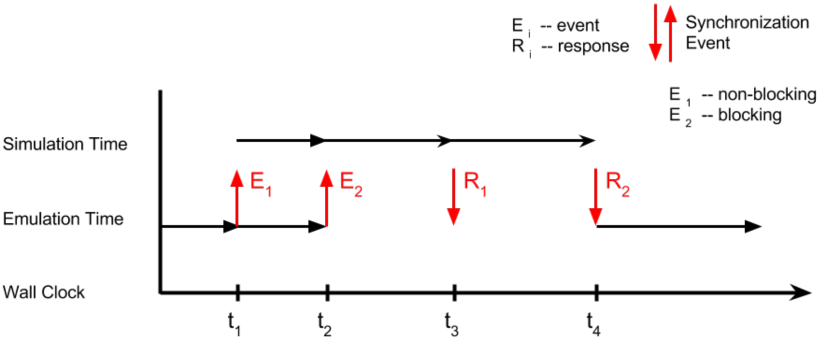
Design - Parallel Execution



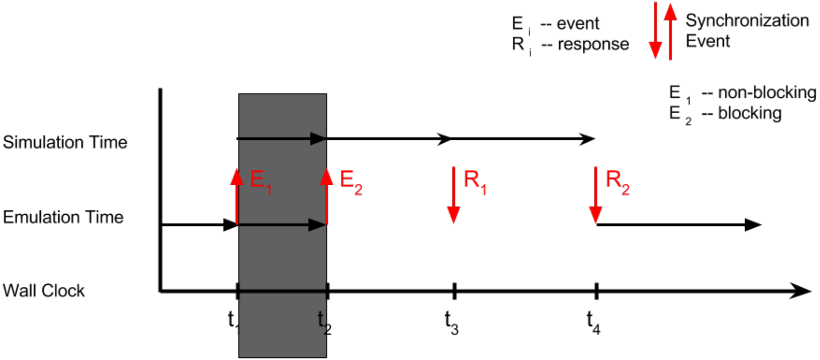
Design - Parallel Execution



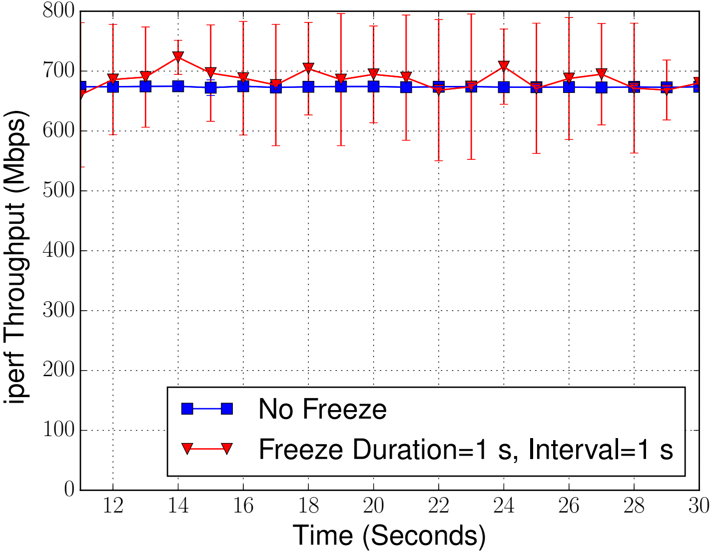
Design - Parallel Execution



Design - Parallel Execution

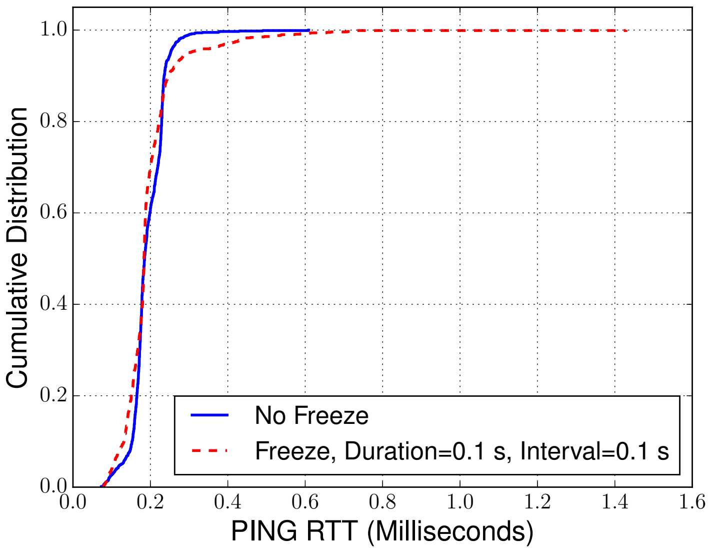


Evaluation



800
700
600
500
400
300
200
100
0
iperf Throughput (Mbps)
12
14
16
18
20
22
24
26
28
30
Time (Seconds)
No Freeze
Freeze Duration=1 s, Interval=1 s

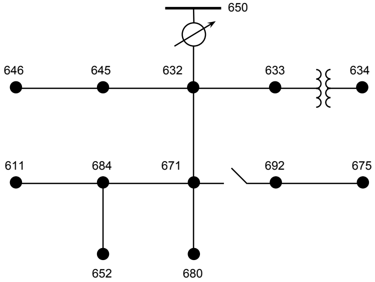
Evaluation



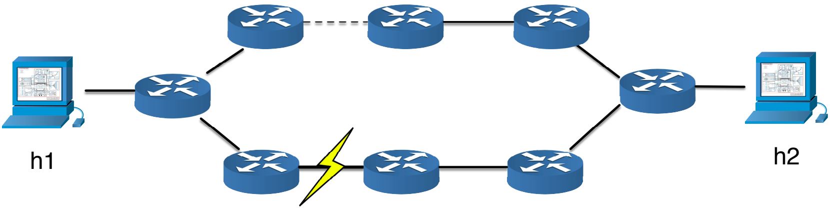
Cumulative Distribution
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
PING RTT (Milliseconds)
No Freeze
Freeze, Duration=0.1 s, Interval=0.1 s

Use Case

- Demand Response Application
  - Dynamic Wind Generation
  - Energy Storage Device
- Sensor Monitors Generation and Sends to Control Center
- Control Center Sends Commands to the Energy Storage Device



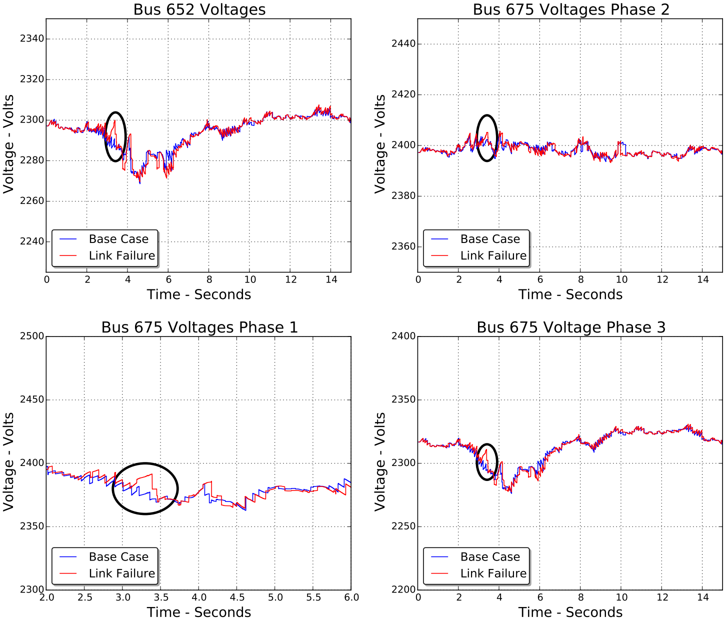
Use Case



Use Case

- Using default SDN reactive forwarding application
- Link failure on primary path at t=3.1
- Network self-heals automatically to take alternative path

Use Case



Bus 652 Voltages
Voltage - Volts
Time - Seconds
Base Case
Link Failure
Bus 675 Voltages Phase 2
Voltage - Volts
Time - Seconds
Base Case
Link Failure
Bus 675 Voltages Phase 1
Voltage - Volts
Time - Seconds
Base Case
Link Failure
Bus 675 Voltage Phase 3
Voltage - Volts
Time - Seconds
Base Case
Link Failure

Future Work

- Hardware-in-the-Loop testbed
  - using embedded Linux hardware
  - modify the scheduler
- Create a distributed emulator environment for scalability
- Develop SDN Smart Grid Applications
  - Context aware intrusion detection and mitigation
  - Application based self-healing

Conclusion

- DSSnet
  - Electric Power Simulation
  - Communication Network Emulation
- For Smart Grid Planning and Evaluation