Declarative Toolkit for Rapid Network Protocol Simulation and Experimentation

Shivkumar C. Muthukumar*, Xiaozhou Li*, Changbin Liu*, Joseph B. Kopena+, Mihai Oprea*, Boon Thau Loo*

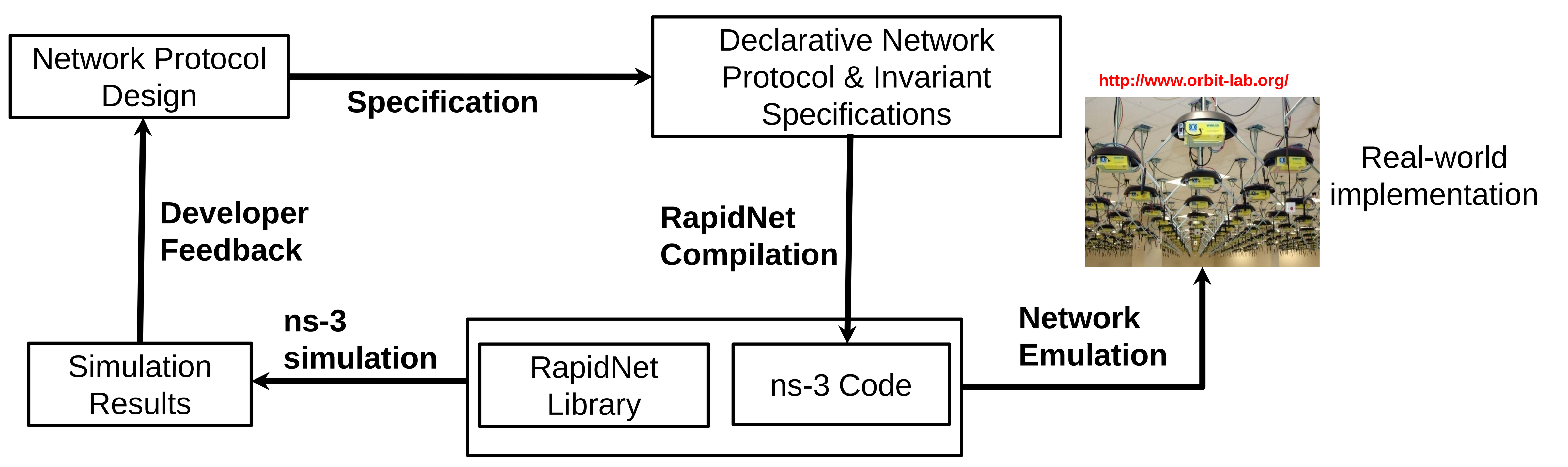
* University of Pennsylvania, +Drexel University

http://netdb.cis.upenn.edu/rapidnet/

Introduction

- **RapidNet Toolkit**
  - A development toolkit for rapid network simulation, prototyping, and experimentation
  - Integrates a declarative networking engine with the ns-3 simulator
  - Develop high-level specifications of network protocols, compiled into ns-3 code for simulation and testbed experimentation
- **Motivation**
  - Proliferation of new protocol designs.
  - One-size-fits-all protocol does not exist:
    - Variability in network connectivity, wireless channels, mobility
    - Wide range of traffic patterns
  - Lack of systematic tools to study protocols under a variety of mobility and traffic settings

Overview of RapidNet



Declarative Networking

- **Declarative Networking**
  - Use a database query language to express declarative specifications of network protocols
  - Declarative specifications are executed by distributed query engine (Click execution model) to implement network protocols
- **Key Advantages**
  - Compact and high-level representation
  - Orders of magnitude reduction in code size
  - Easy customization of existing protocols to derive new ones
  - Potential for verification and correctness checks

Protocols at Demonstration

- **Reactive**
  - DSR (Dynamic Source Routing) (10 rules)
- **Proactive**
  - LS (Link State) (8 rules)
  - HSLS (Hazy Sighted Link State routing) (14 rules)
- **DTN (Delay Tolerant Networks)**
  - Epidemic Routing (15 rules)
- **Wireless Neighbor Discovery**
- **Chord Distributed Hash Table**

Example Specifications

All-pairs reachability protocol

R1 reachable(@S,D) ← link(@S,D)

R2 reachable(@S,D) ← link(@S,Z), reachable(@Z,D)

- Input: link(@source, destination)
- Output: reachable(@source, destination)

Link-state flooding

LS1 lsu(@S,S,N,C,S) :- link(@S,N,C).
LS2 lsu(@M,S,N,C,Z) :- link(@Z,M,C1), lsu(@Z,S,N,C,W), M!=W.

Hazy-sighted link-state flooding

HSLS1 lsu(@S,S,N,C,S,TTL) :- periodic(@S,T),
link(@S,N,C), T=f_pow(2,K)*Te,
TTL=f_pow(2,K), range(@S,K).
HSLS2 lsu(@S,S,N,C,Z,K-1) :- lsu(@Z,S,N,C,W,K),
link(@Z,M,C1), K>0, M!=W.

Neighbor discovery

D1 beaconMsg(@BROADCAST, S) :- periodic(@S,10).
D2 link(@N,S,1) :- beaconMsg(@N,S).

Epidemic exchange

E1 eBitVecReq(@Y,X,V):- summaryVec(@X,V),
eDetectNewLink(@X,Y).
E2 eBitVecReply(@X,Y,V):- eBitVecReq(@Y,X,V1),
summaryVec(@Y,V2),
V=f_vec_AND(V1,f_vec_NOT(V2)).
E3 eNewMsg(@Y,I,S,D):- eBitVecReply(@X,Y,V),
msgs(@X,I,S,D), f_vec_in(V,I)==true.
E4 msgs(@Y,I,S,D):- eNewMsg(@Y,I,S,D).

Ongoing Research

- **Policy-based Adaptive MANETs**
- **Hybrid link-state**
  - HSLS: incurs low bandwidth overhead, scales better
  - LS: quick convergence, perform better in stable network
  - AA: link availability

```
#define THRES 0.5
s1 linkAvail(@M,AVG<AA>) :- lsu(@M,S,N,AA,Z,K).
s2 useHSLS(@M) :- linkAvail(@M,AA), AA<THRES.
s3 useLS(@M) :- linkAvail(@M,AA), AA>=THRES.
```

- **Hybrid Proactive-Epidemic**

```
#define THRES 1.2
pe1 useEpidemic(@M,S,D) :- bestPath(@M,S,D,P,C),
                            C>THRES.
pe2 useLS(@M,S,D) :- bestPath(@M,S,D,P,C), C<=THRES.
```

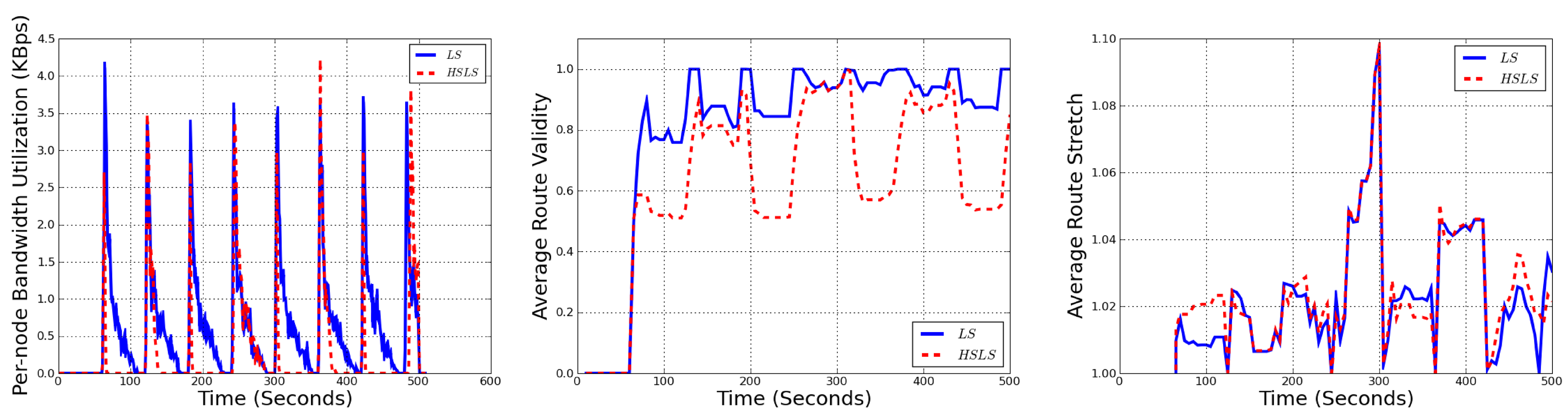
- **Verifiable Networking**
  - Mechanized theorem provers for verifying and generating declarative networks
- **Dynamic Network Composition**

Recent Publications

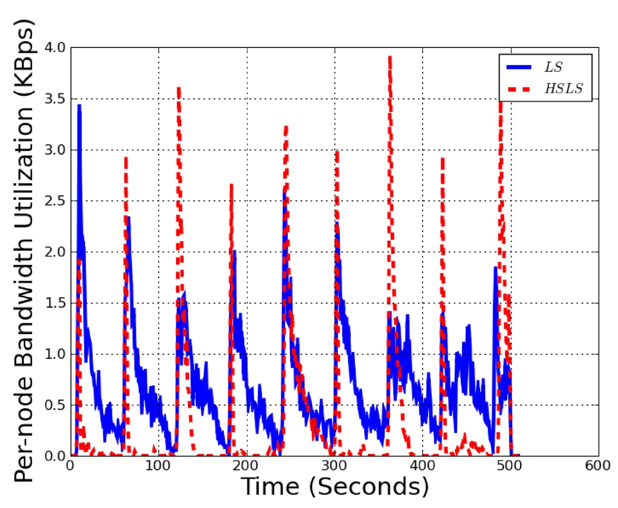
- **Declarative Policy-based Adaptive MANET Routing.** 17th IEEE International Conference on Network Protocols (ICNP 2009)
- **RapidMesh: Declarative Toolkit for Rapid Experimentation of Wireless Mesh Networks.** 4th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2009)
- **Declarative Network Verification** 11th International Symposium on Practical Aspects of Declarative Languages (PADL 2009)
- **MOSAIC: Unified Declarative Platform for Dynamic Overlay Composition.** 4th Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT 2008)

ORBIT Experimentations

ORBIT Testbed: based in Rutgers-WINLAB, 1GHz VIA Nehemiah, 64KB cache 512MB RAM, 802.11 a/b/g ad-hoc mode

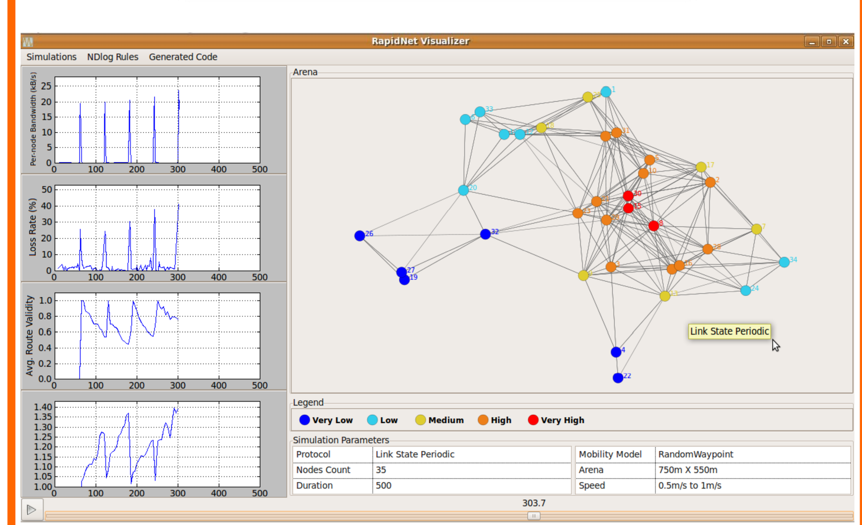


Per-node communication overhead (KBps), average validity and average stretch for periodic LS and HSLS



Per-node communication overhead (KBps) for triggered + periodic LS and HSLS

Visualizer



Visualization of ns-3 traces obtained from declarative networking executions within RapidNet