Declarative Policy-based Networking

Boon Thau Loo
University of Pennsylvania
http://netdb.cis.upenn.edu

IEEE POLICY 2010

23 Jul 2010

Outline of Talk

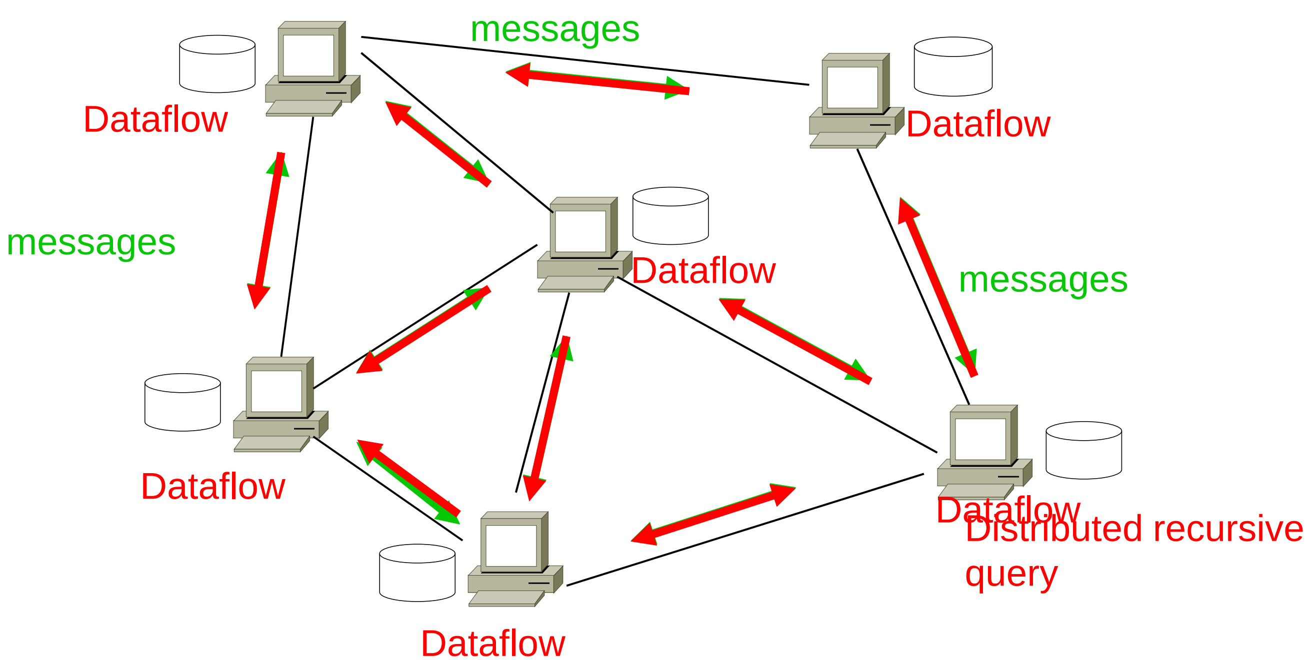
- Overview of declarative networking
- Connections between Distributed Datalog and network routing
- Declarative Secure Networking
  - Security policies in networking
  - Application-aware Anonymity (A3)
- Policy-based Adaptive Routing
  - Policies for hybridizing routing protocols for performance in dynamic networks

Declarative Networking

- A declarative framework for networks:
  - Declarative language: *“ask for what you want, not how to implement it”*
  - Declarative specifications of networks, compiled to distributed dataflows
  - Runtime engine to execute distributed dataflows

- Observation: *Recursive queries* are a natural fit for routing
- Recursive queries:
  - Traditionally for querying graph data structures stored in databases
  - Uses the Datalog language. Designed to be processed using database operators with set semantics.

A Declarative Network



Traditional Networks	Declarative Networks
Network State	Distributed database
Network protocol	Recursive Query Execution
Network messages	Distributed Dataflow

The Case for Declarative

- Ease of programming:
  - Compact and high-level representation of protocols
  - Orders of magnitude reduction in code size
  - Easy customization and rapid prototyping
- Safety:
  - Queries are “sandboxed” within query processor
  - Potential for static analysis and theorem proving techniques on safety
- What about efficiency?
  - No fundamental overhead when executing standard routing protocols
  - Application of well-studied query optimizations

Large Library of Declarative Protocols

- Example implementations to date:
  - Wired routing protocols: DV, LS **[SIGCOMM'05]**
  - Overlay networks: Distributed Hash Tables, multicast overlays **[SOSP'05]**
  - **Secure distributed systems [ICDE'09, NDSS'10, SIGMOD'10]**
  - **Wireless: DSR, AODV, OLSR, HSLS, hybrid protocols [ICNP'09]**
  - Network composition: Chord over RON, i3+RON **[CoNEXT'08]**
  - Distributed provenance [SIGMOD'10]
  - Others: sensor networking protocols **[Sensys'07]**, fault tolerance protocols **[NSDI'08],** replication **[NSDI'09]**, and cloud analytics **[Eurosys'10]**

Outline of Talk

- Overview of declarative networking
- Connections between Distributed Datalog and network routing
- Declarative Secure Networking
- Policy-based Adaptive Routing

Introduction to Datalog

Datalog rule syntax:

<result> ← <condition1>, <condition2>, … , <conditionN>.

Head Body

- Types of conditions in body:
  - Input tables: *link(src,dst)* predicate
  - Arithmetic and list operations
- Head is an output table
  - Recursive rules: result of head in rule body

All-Pairs Reachability

→ R1: reachable(S,D) ← link(S,D)

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

link(a,b) – “there is a link from node *a* to node *b*”

reachable(a,b) – “node *a* can reach node *b*”

“For all nodes S,D,
If there is a link from S to D, then S can reach D”.

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

All-Pairs Reachability

R1: reachable(S,D) ← link(S,D)

➔ R2: reachable(S,D) ← link(S,Z), reachable(Z,D)

"For all nodes S,D and Z,

If there is a link from S to Z, AND Z can reach D, then S can reach D".

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

Network Datalog

Location Specifier "@S"

R1: reachable(@S,D) ← link(@S,D)

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

Query: reachable(@M,N) ← All-Pairs Reachability

Query: reachable(@a,N)

Input table:

link (a)

@S	D
@a	b

link (b)

@S	D
@b	c
@b	a

link (c)

@S	D
@c	b
@c	d

link (d)

@S	D
@d	c

Output table:

reachable (a)

@S	D
@a	b
@a	c
@a	d

Query: reachable(@a,N)

reachable (b)

@S	D
@b	a
@b	c
@b	d

reachable (c)

@S	D
@c	a
@c	b
@c	d

reachable (d)

@S	D
@d	a
@d	b
@d	c

Implicit Communication

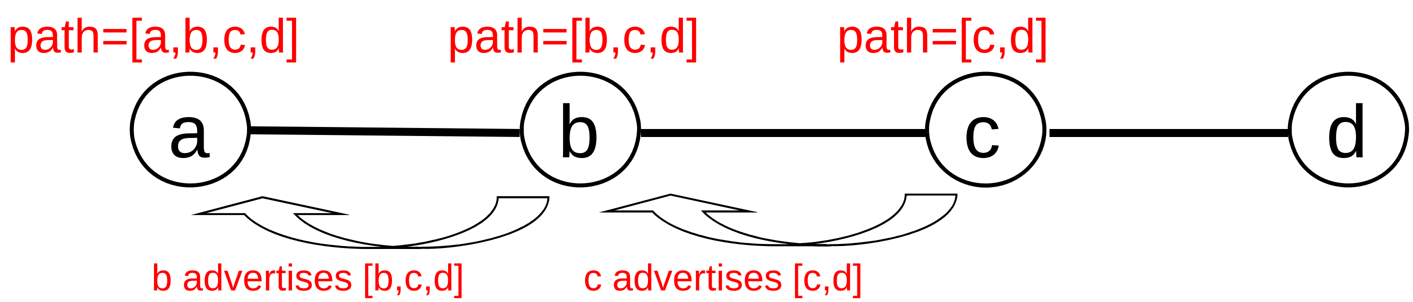
- A networking language with no explicit communication:

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

Data placement induces communication

Path Vector Protocol Example

- Advertisement: entire path to a destination
- Each node receives advertisement, add itself to path and forward to neighbors



Path Vector in Network Datalog

R1: path(@S,D,P) ← link(@S,D), P=(S,D).

R2: path(@S,D,P) ← link(@Z,S),path(@Z,D,P_2), P=S•P_2.

Query: path(@S,D,P)

Add S to front of P_2

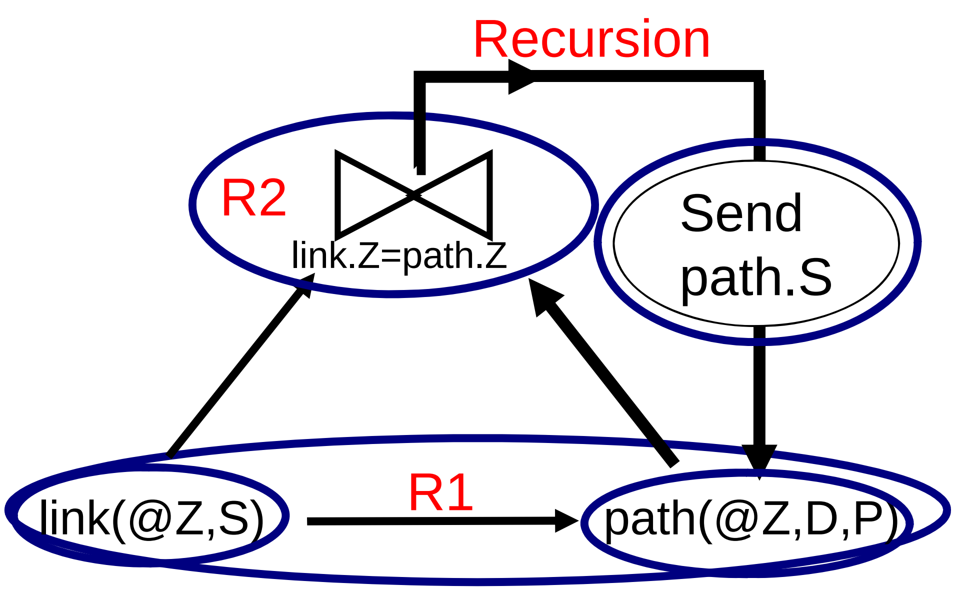
- Input: link(@source, destination)
- Query output: path(@source, destination, pathVector)

Datalog ➔ Execution Plan

R1: path(@S,D,P) ← link(@S,D), P=(S,D).

R2: path(@S,D,P) ← link(@Z,S), path(@Z,D,P_2), P=S • P_2.

Matching variable Z = “Join” ⋈



Query Execution

R1: path(@S,D,P) ← link(@S,D), P=(S,D).

R2: path(@S,D,P) ← link(@Z,S), path(@Z,D,P_2), P=S•P_2.

Query: path(@a,d,P)

Neighbor table:

link (a)

@S	D
@a	b

link (b)

@S	D
@b	c
@b	a

link (c)

@S	D
@c	b
@c	d

link (d)

@S	D
@d	c

Forwarding table:

path (a)

@S	D	P

path (b)

@S	D	P

path (c)

@S	D	P
@c	d	[c,d]

Query Execution

R1: path(@S,D,P) ← link(@S,D), P=(S,D).

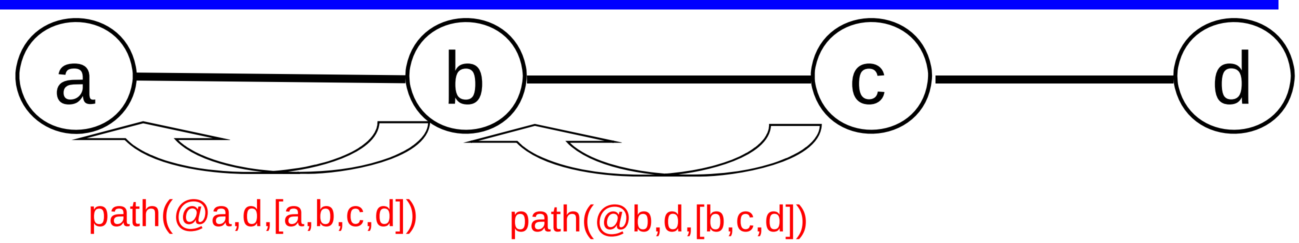
R2: path(@S,D,P) ← link(@Z,S), path(@Z,D,P_2), P=S•P_2.

Query: path(@a,d,P)

Matching variable Z = "Join" ⋈

link link link link

Communication patterns are identical to those in the actual path vector protocol



Forwarding table:

path

@S	D	P
@a	d	[a,b,c,d]

path

@S	D	P
@b	d	[b,c,d]

path

@S	D	P
@c	d	[c,d]

Outline of Talk

- Overview of declarative networking
- Connections between Distributed Datalog and network routing
- Declarative Secure Networking
- Policy-based Adaptive Routing

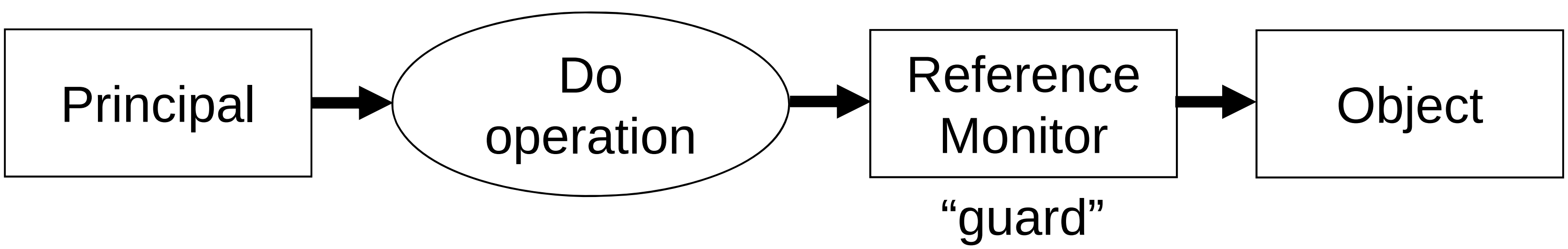
Unified Declarative Platform for Secure Networked Information Systems.
Wenchao Zhou, Yun Mao, Boon Thau Loo, and Martín Abadi.
25th International Conference on Data Engineering (ICDE), Apr 2009.

A3: An Extensible Platform for Application-Aware Anonymity.
Micah Sherr, Andrew Mao, William R. Marczak, Wenchao Zhou, Boon Thau Loo, and Matt Blaze
17th Annual Network & Distributed System Security Symposium (NDSS), 2010.

SecureBlox: Customizable Secure Distributed Data Processing
William R. Marczak, Shan Shan Huang, Martin Bravenboer, Micah Sherr, Boon Thau Loo, and Molham Aref.
ACM SIGMOD International Conference on Management of Data, 2010.

Background: Access Control

- Central to security, pervasive in computer systems
- Broadly defined as:
  - Enforce security policies in a multi-user environment
  - Assigning credentials to principals to perform actions
  - Commonly known as ***trust management***
- Model:
  - objects, resources
  - requests for operations on objects
  - sources for requests, called principals
  - a reference monitor to decide on requests



Background: Access Control

- **Access control languages:**
  - *Analyzing* and *implementing* security policies
  - Several runtime systems based on distributed Datalog/Prolog
- **Binder** [Oakland 02]**: a simple representative language**
  - **Context:** each principal has its own context where its rules and data reside
  - **Authentication:** "says" construct (digital signatures)

    At alice:

    b1: access(P,O,read) :- good(P).

    b2: access(P,O,read) :- bob says access(P,O,read).
  - "In alice's context, any principal P may access object O in read mode if P is good (b1) or, bob says P may do so (b2 - delegation)"
- Several languages and systems: Keynote [RFC-2704], SD3 [Oakland 01], Delegation Logic [TISSEC 03], etc.

Comparing the two

- Declarative networking and access control languages are based on logic and Datalog
- Similar observation:
  - Martín Abadi. "*On Access Control, Data Integration, and Their Languages*."
  - Comparing data-integration and trust management languages
- Both extend Datalog in surprisingly similar ways
  - Context (location) to identify components (nodes) in a distributed system
  - Suggests possibility to unify both languages
  - Leverage ideas from database community (e.g. efficient query processing and optimizations) to enforce access control policies
- Differences
  - Top-down vs bottom-up evaluation
  - Trust assumptions

Secure Network Datalog (SeNDlog)

- Rules within a context
  - Untrusted network
  - Predicates in rule body in local context
- Authenticated communication
  - "says" construct
  - *Export predicate:* "X says p@Y"
    - X exports the predicate p to Y.
  - *Import predicate:* "X says p"
    - X asserts the predicate p.

r1: reachable(@S,D) :- link(@S,D).
r2: reachable(@S,D) :- link(@S,Z),
reachable(@Z,D).

⇩ *localization rewrite*

At S:
s1: reachable(@S,D) :- link(@S,D).
s2: linkD(D,S)@D :- link(S,D).
s3: reachable(Z,D)@Z :- linkD(S,Z),
reachable(S,D).

⇩ *authenticated communication*

At S:
s1: reachable(S,D) :- link(S,D).
s2: S says linkD(D,S)@D :- link(S,D).
s3: S says reachable(Z,D)@Z :-
Z says linkD(S,Z),
W says reachable(S,D).

Authenticated Path Vector Protocol

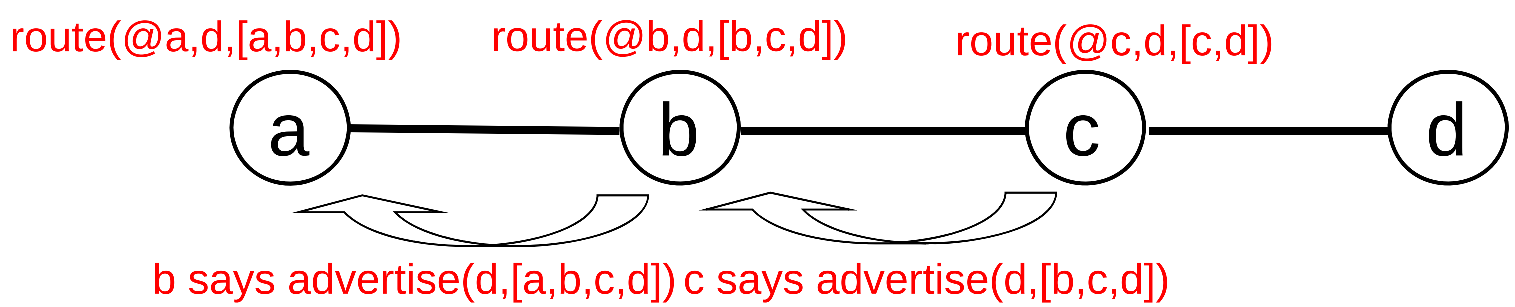
```
At Z,
    z1 route(Z,X,P) :- neighbor(Z,X), P=f_initPath(Z,X).
    z2 route(Z,Y,P) :- X says advertise(Y,P), acceptRoute(Z,X,Y).
    z3 advertise(Y,P1)@X :- neighbor(Z,X), route(Z,Y,P),
                               carryTraffic(Z,X,Y), P1=f_concat(X,P).
```

- Import and export policies
- Basis for Secure BGP
  - Authenticated advertisements
  - Authenticated subpaths (provenance)
  - Encryption (for secrecy) with cryptographic functions

Authenticated Path Vector Protocol

At Z,

z1 route(Z,X,P) :- neighbor(Z,X), P=f_initPath(Z,X).
z2 route(Z,Y,P) :- X says advertise(Y,P), acceptRoute(Z,X,Y).
z3 advertise(Y,P1)@X :- neighbor(Z,X), route(Z,Y,P),
carryTraffic(Z,X,Y), P1=f_concat(X,P).



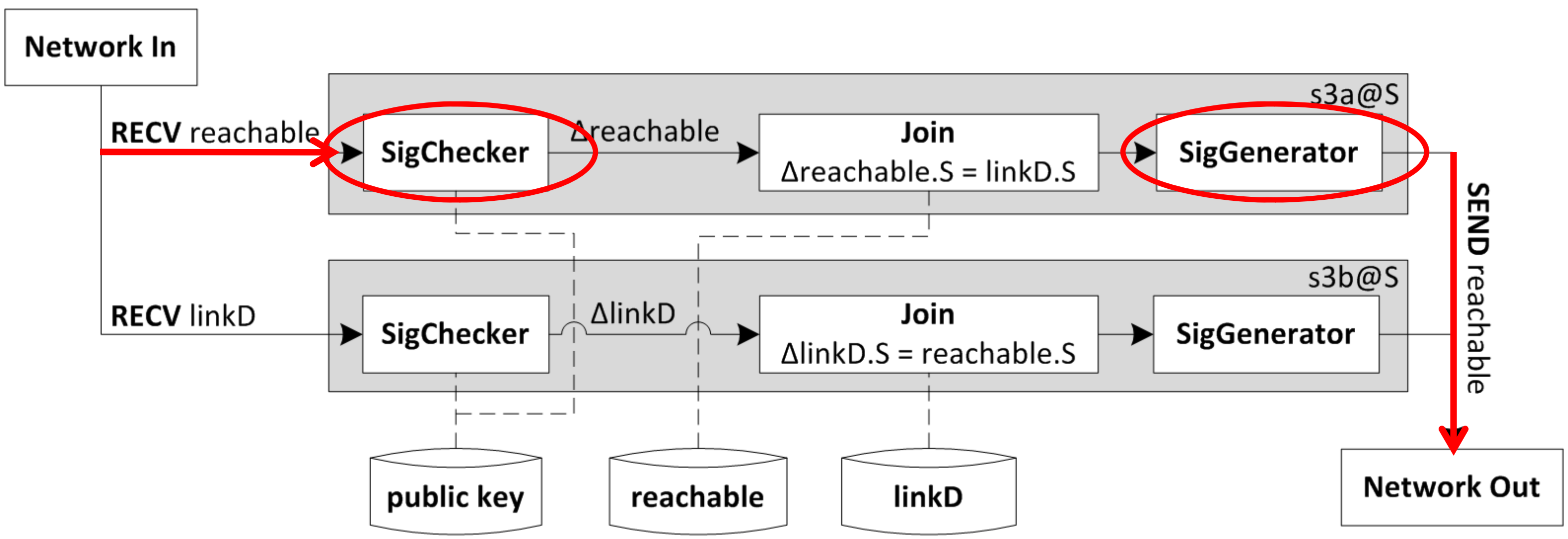
Example Protocols in SeNDlog

- **Secure network routing**
  - Nodes import/export signed route advertisements from neighbors
  - Advertisements include signed sub-paths (*authenticated provenance*)
  - Building blocks for secure BGP
- **Secure packet forwarding**
- **Secure DHTs**
  - Chord DHT – authenticate the node-join process
  - Signed node identifiers to prevent malicious nodes from joining the DHT
- **Customizable anonymous routing**
  - Application-aware Anonymity (http://a3.cis.upenn.edu)
- **Customizable distributed data processing**
  - Integration with LogicBlox (http://www.logicblox.com) **[SIGMOD'10]**

Execution Plan

- Pipelined semi-naive evaluation [SIGMOD'06]
  - Asynchronous communication in distributed settings
- Each delta rule corresponds to a "rule strand"
- Additional operators to support authenticated communication

At S, reachable(Z,D)@Z :- Z says linkD(S,Z), W says reachable(S,D).



Outline of Talk

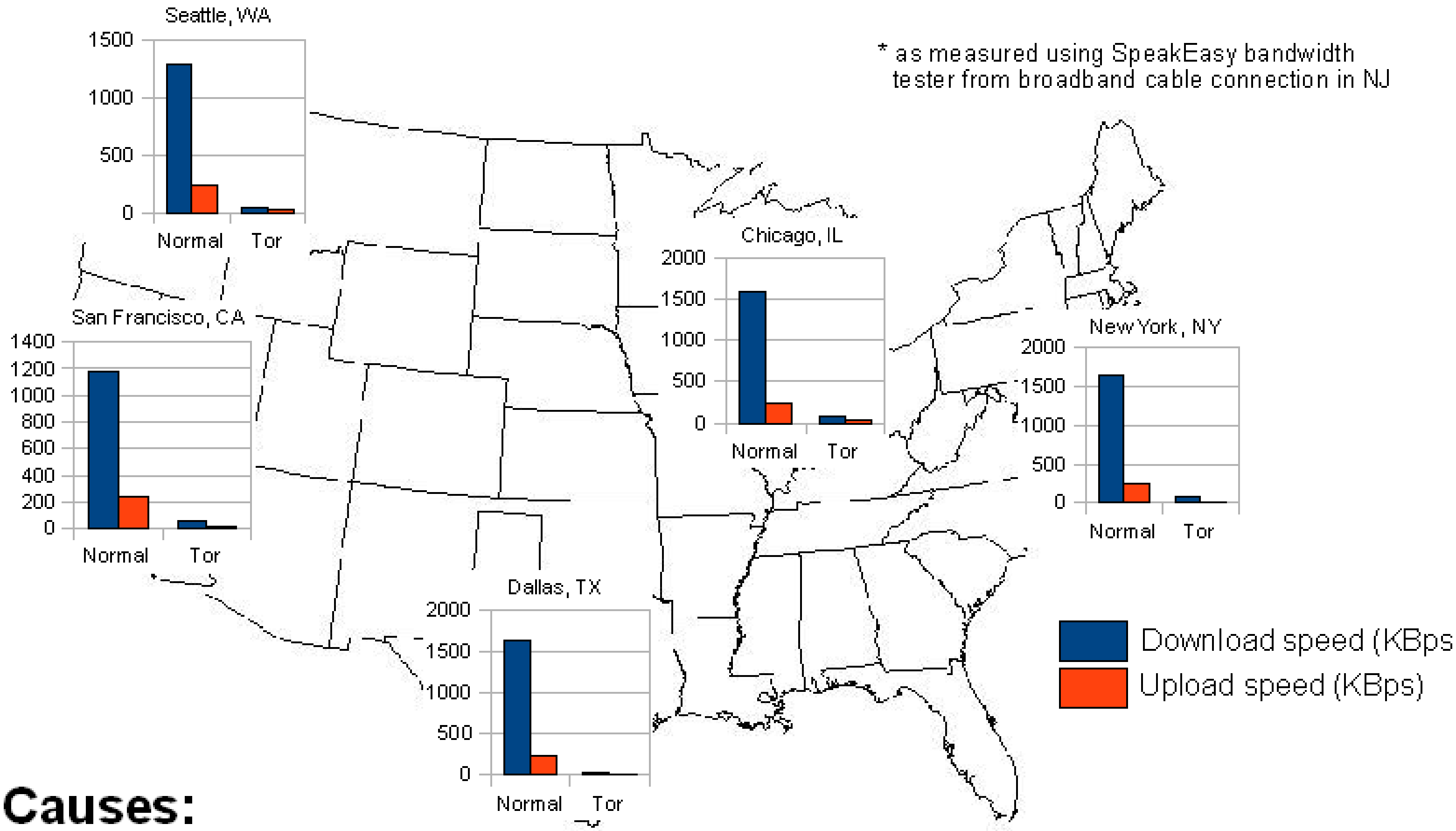
- Overview of declarative networking
- Connections between Distributed Datalog and network routing
- Declarative Secure Networking
- Policy-based Adaptive Routing

Unified Declarative Platform for Secure Networked Information Systems.
Wenchao Zhou, Yun Mao, Boon Thau Loo, and Martín Abadi.
25th International Conference on Data Engineering (ICDE), Apr 2009.

A3: An Extensible Platform for Application-Aware Anonymity.
Micah Sherr, Andrew Mao, William R. Marczak, Wenchao Zhou, Boon Thau Loo, and Matt Blaze
17th Annual Network & Distributed System Security Symposium (NDSS), 2010.

SecureBlox: Customizable Secure Distributed Data Processing
William R. Marczak, Shan Shan Huang, Martin Bravenboer, Micah Sherr, Boon Thau Loo, and Molham Aref.
ACM SIGMOD International Conference on Management of Data, 2010.

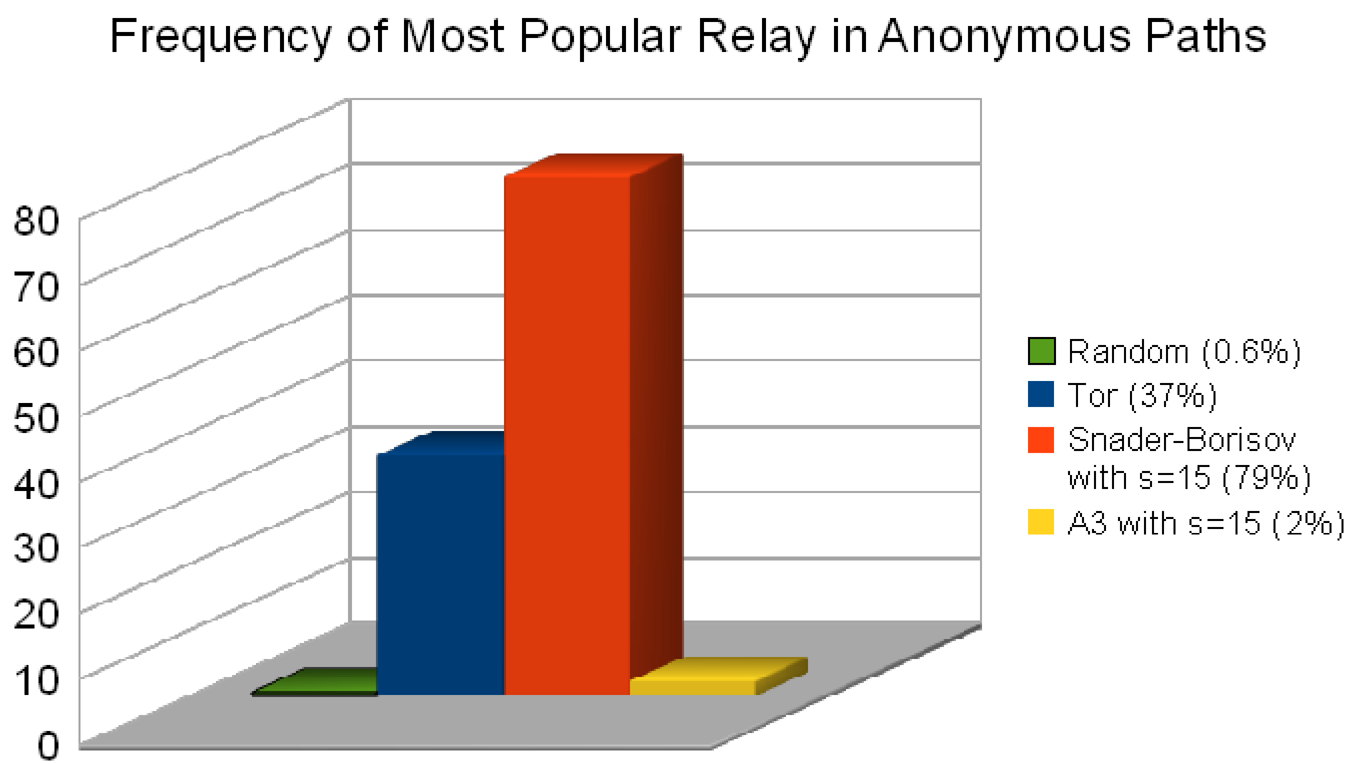
Observation:
Existing Anonymity Systems are Slow



Causes:

- Congestion (1,500 relays for 100,000+ clients)
- Lack of scalability (centralized directory servers)
- Traffic (BitTorrent represents 40% of Tor traffic [McCoy-PETS08])

Observation: Existing Anonymity Systems are Vulnerable



Causes:

- Relay selection algorithms biased by <u>self-reported</u> ***node characteristics*** (i.e., bandwidth)

  An attractive (high-bandwidth) node is attractive to all clients

Observation: “Performance” depends on the application

High bandwidth

Low latency

Low jitter

Network diversity

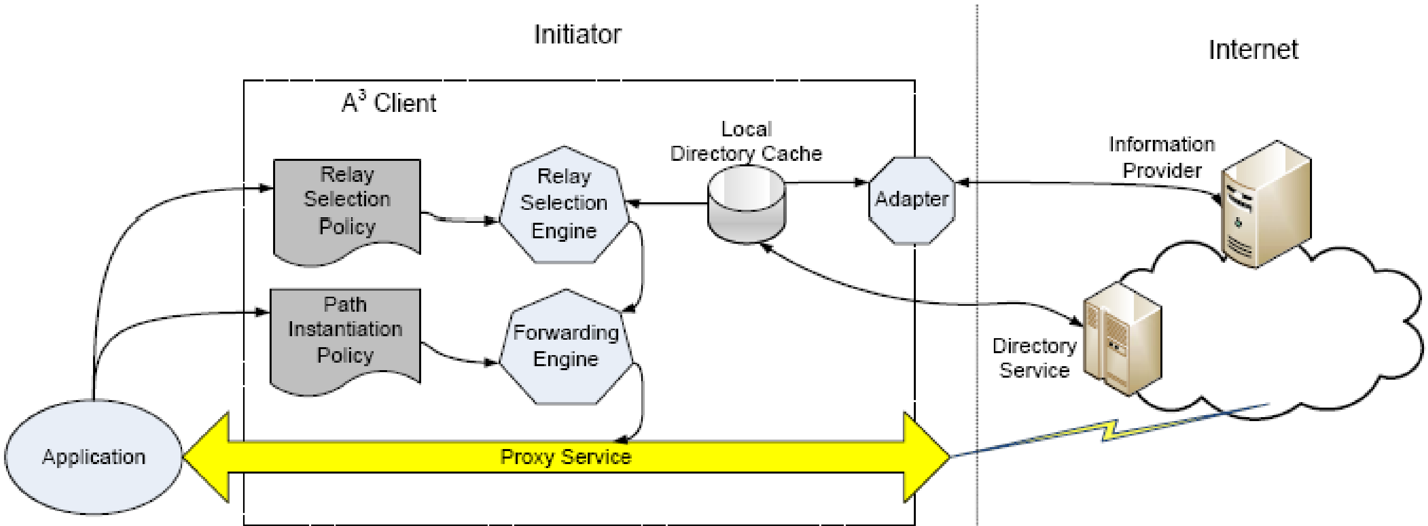
Relay Selection Techniques

Technique	Description	Benefits	Example
Uniform	Select uniformly at random	Stronger anonymity	Email mixing
Tor	Bias based on bandwidth	High bandwidth and utilization	Web browsing
Snader-Borisov	Tunable bias towards bandwidth	Tunable anonymity and performance	File transfers
Weighted	Bias based on link metrics	Versatility and expressiveness	Streaming multicast
Hybrid	Combines above techniques	Supports diverse requirements	Video conferencing
Constraint	Meet specific e2e requirements	Supports real-time demands	VoIP

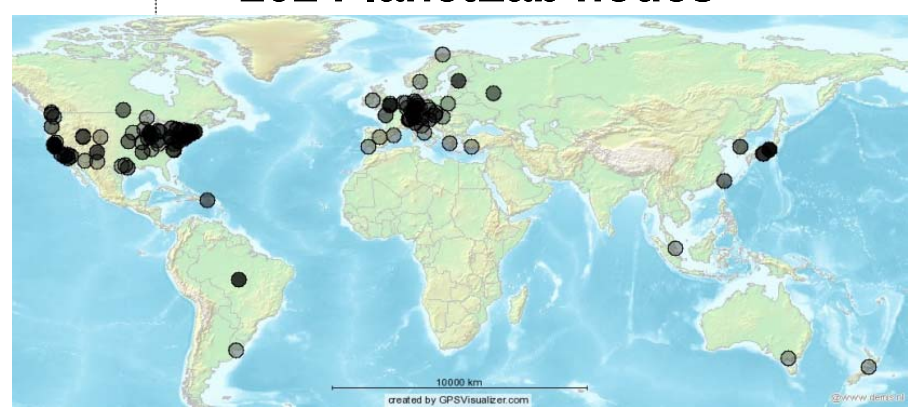
Link-based relay selection [PETS’09]
Path instantiation policies: Onion routing, Tor incremental telescoping strategy, Crowds

A3 on PlanetLab **http://a3.cis.upenn.edu**

A3: An Extensible Platform for Application-Aware Anonymity. NDSS'09



202 PlanetLab nodes



Contributions of A3:

- Tunable relay selection strategies that meet diverse performance requirements
- SeNDlog-based policy language for specifying relay selection and path construction
-Veracity: vote-based network coordinates (USENIX'09)

Outline of Talk

- Overview of declarative networking
- Connections between Distributed Datalog and network routing
- Declarative Secure Networking
- Policy-based Adaptive Routing

Declarative Policy-based Adaptive MANET Routing
Changbin Liu, Richardo Correa, Xiaozhou Li, Prithwish Basu, Boon Thau Loo, and Yun Mao.
17th IEEE International Conference on Network Protocols (ICNP), Princeton, New Jersey, Oct, 2009.

Motivation

- Mobile ad-hoc network (MANET) or heterogeneous wired/wireless environment
- Variety of MANET routing protocols
  - Reactive (DSR, AODV)
  - Proactive (LS, OLSR, HSLS)
  - Epidemic
  - Hybrid (ZRP, SHARP)
- However, a *one-size-fits-all* routing protocol does not exist:
  - Variability in network connectivity, wireless channels, mobility
  - Wide range of traffic patterns

Policy-based Adaptive Routing

- **Using the declarative networking framework**
  - Implement a wide range of MANET protocols
  - Hybrid protocol composed from any number of known protocols
  - Generic set of policies for selecting and switching among different routing protocols due to network/traffic conditions
  - Policies also specified in declarative language
- **Examples**
  - Hybrid link state
  - Hybrid proactive-epidemic

Declarative MANET protocols

- Reactive
  - DSR (Dynamic Source Routing) (10 rules)
- Proactive
  - LS (Link State) (8 rules)
  - HSLS (Hazy Sighted Link State routing) (14 rules)
  - OLSR (Optimized Link State Routing) (27 rules)
- Epidemic
  - Summary Vector based (16 rules)

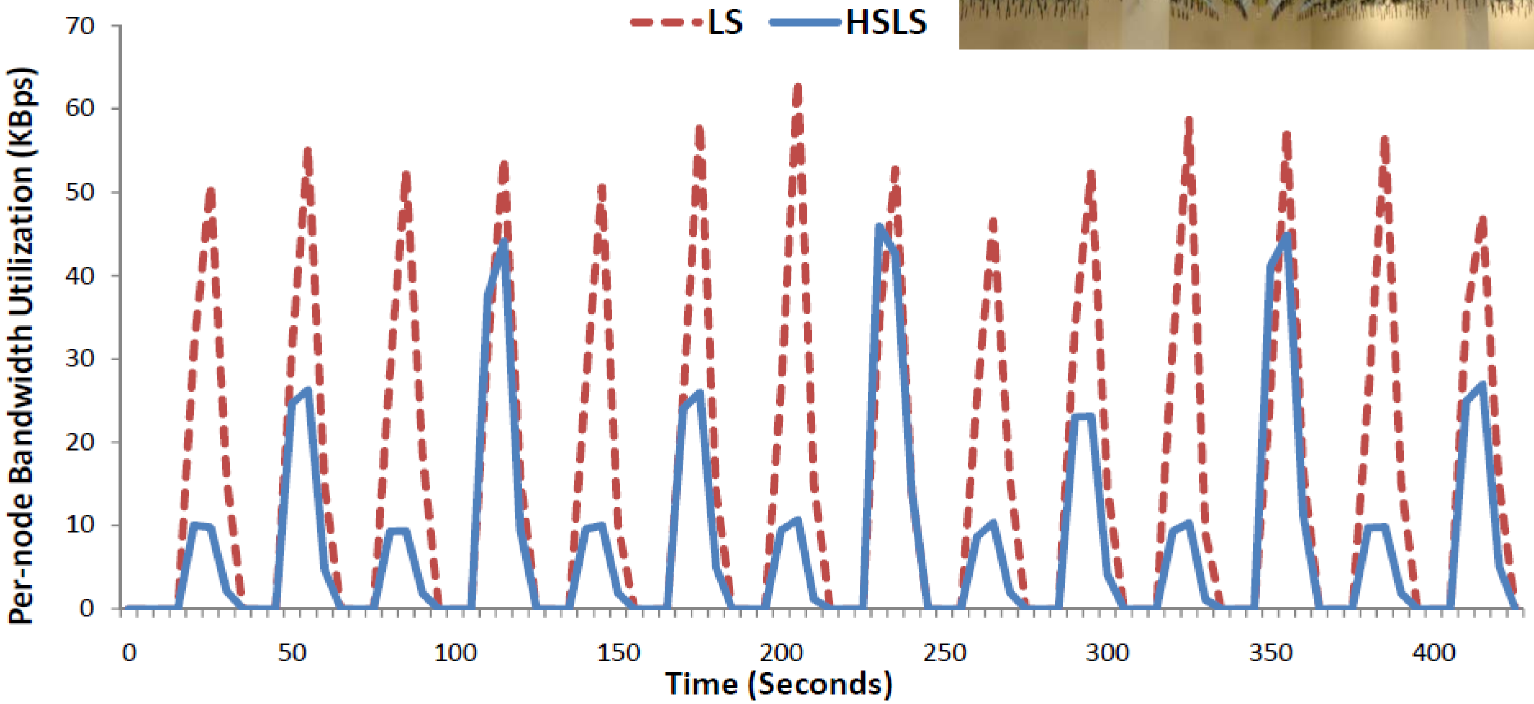
Measurements on ORBIT Wireless Testbed

ORBIT wireless testbed at Rutgers University

1 GhZ VIA Nehemiah, 64 KB cache, 512 MB RAM

Atheros AR5212 chipset 802.11 a/b/g ad hoc mode

33 nodes in a 7m x 5m grid



Example(1): Hybrid Link State

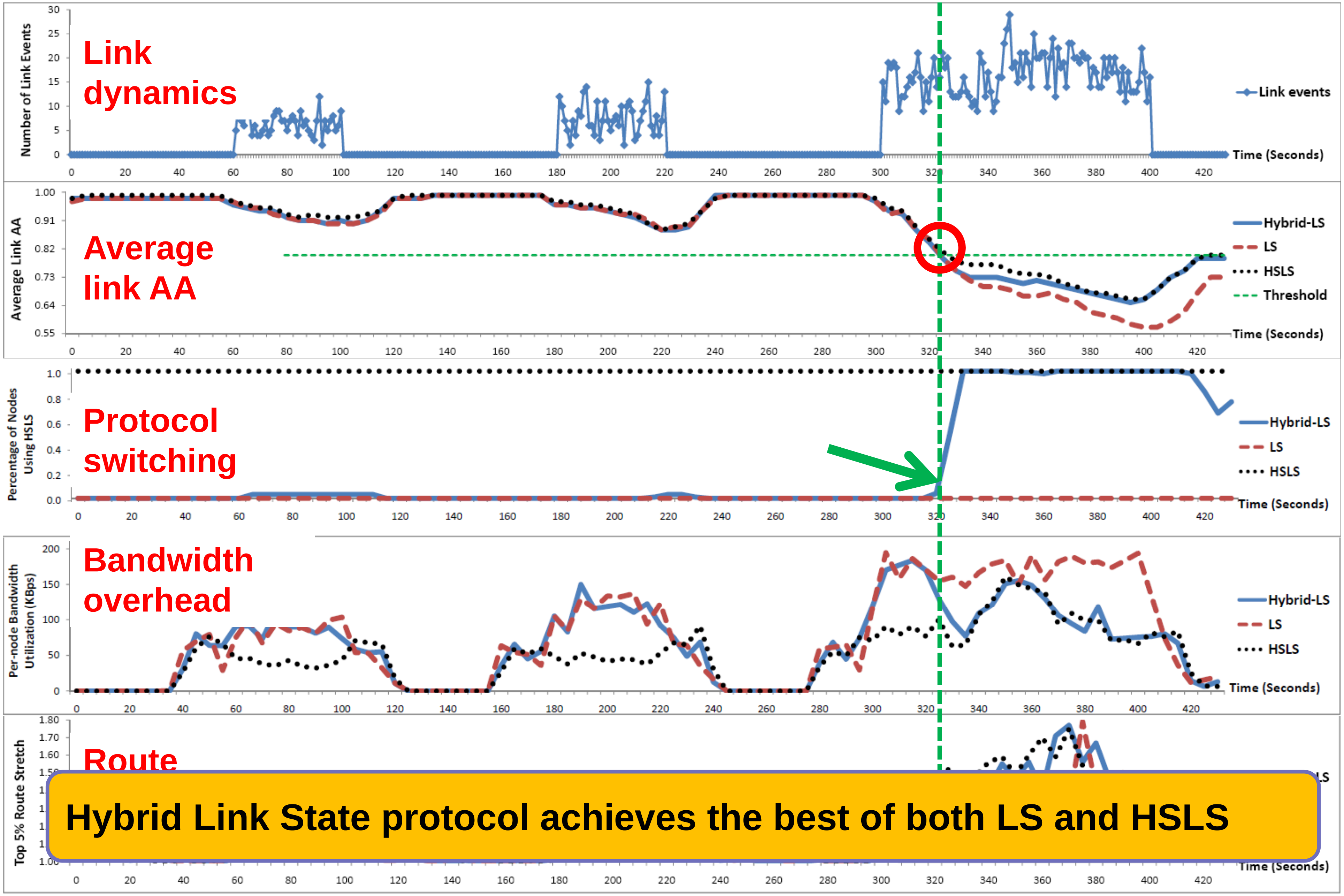
- LS: quick convergence, may perform better in stable network
- HSLS: incurs low bandwidth overhead, scales better

- Adapt between LS and HSLS
  - Low mobility: LS
  - High mobility: HSLS
  - Mobility measurement: link average availability (AA), i.e. percentage of time when link is up

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

Evaluation of Hybrid Link State

- 33 wireless nodes on 7m x 5m grid on **ORBIT testbed** that communicate over 802.11a
- Linux *iptables* to filter packets from non-neighbors
- Emulate 2-dimensional random waypoint model
- Random jitter and desynchronized broadcasting to alleviate packet collision
- Alternate at 60 seconds interval of:
  - Moderate speed: nodes move at 0.06 m/s
  - Fast speed: nodes move at 0.15m/s



Hybrid Link State protocol achieves the best of both LS and HSLS

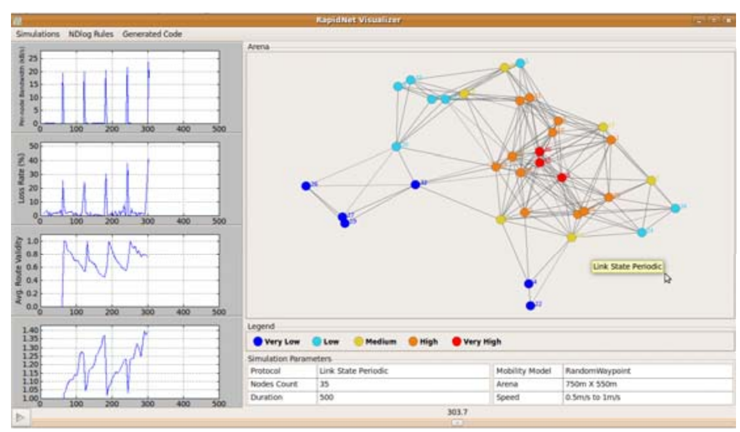
Example(2): Hybrid Proactive-Epidemic

- LS: good performance for well connected network
- Epidemic: for DTN, reliable message delivery in the sacrifice of high bandwidth
- Adapt between LS and Epidemic
  - Well connected network: LS
  - Disrupted network: Epidemic
  - Network connectivity measurement: path length or cumulative AA
- Refer to our paper for more details about evaluation

Declarative framework makes it easier to express policies for runtime adaptation of routing protocols

Conclusion

- Declarative networking –network protocols using a declarative language
- Two instances of declarative policy-based networking
  - Declarative Secure Networking
  - Adaptive routing
- Ongoing work :
  - Policy-based wireless channel selection + routing
  - Secure cloud data management, secure network provenance [SIGMOD'10]
  - Formal network verification
- RapidNet declarative networking system
  - http://netdb.cis.upenn.edu/rapidnet
  - Code available for download



[SIGCOMM'09 demonstration]

Thank You …

Visit us at http://netdb.cis.upenn.edu