

FSR: Formal Analysis & Implementation Toolkit for Safe Inter-domain Routing

Wenchao Zhou
University of Pennsylvania

Collaboration work with Anduo Wang, Yiqing Ren, Limin Jia, Alexander J.T. Gurney, Boon Thau Loo and Jennifer Rexford

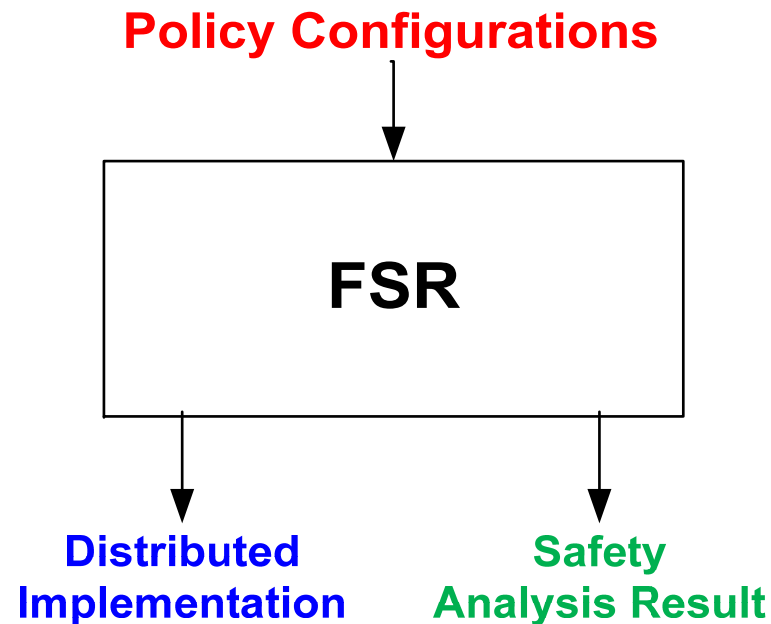


Introduction



- **Today's Internet routing system (BGP) does not guarantee convergence (safety)**
 - Policy configuration largely affect the behavior of BGP
 - Oscillations cause serious performance disruption
- **Limitations of current approaches**
 - **Formal theories:** Manual proofs or counter-examples
 - **Distributed implementation:** Simulations for study protocol overhead and transient behavior during protocol execution
- **Goal of FVR: Bridging the formal theories and distributed implementation**

Architecture (10,000 feet)



- **Input: policy configurations** (as algebra)
 - Examples: Gao-Rexford guideline [SIGMETRICS 00], hop-count
- **Formal side: Safety analysis results**
- **Practical side: Distributed implementation**

Policy as Routing Algebra



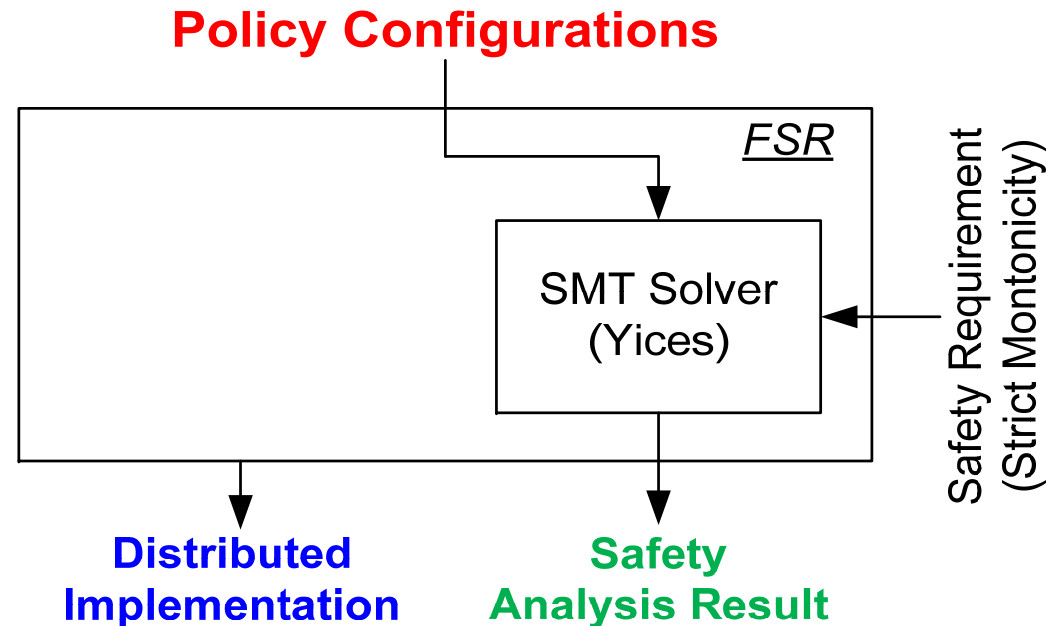
■ An abstract structure $\langle \Sigma, L, <, \oplus \rangle$

- Σ, L describe route and link attributes
- \oplus specifies how to compute new routes
- $<$ determines how to compare routes

■ Gao-Rexford Guideline

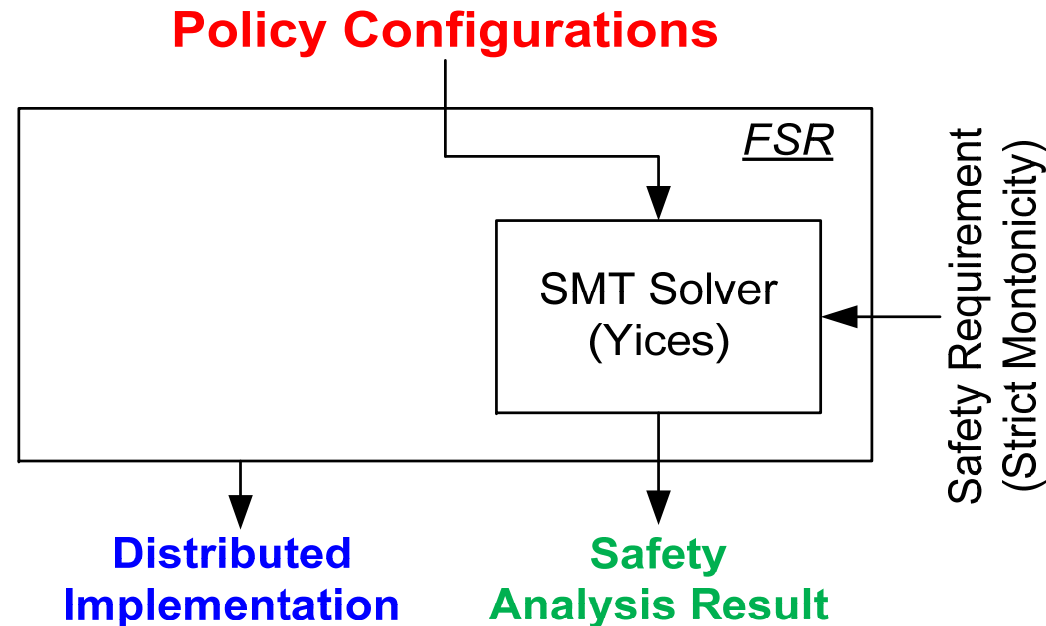
	Explanation	Algebra instance			
Σ	Business relationship: customer, peer, provider	{C, R, P, \emptyset }			
L	Business relationship between neighboring ASes	{c, r, p}			
$<$	Prefer customer (C) over peer(R) / provider(P)	C < R, C < P			
\oplus	How a new route attribute depends on existing routes and links; and whether a node exports/imports certain classes of routes	\oplus	C	R	P
		c	C	\emptyset	\emptyset
		r	R	\emptyset	\emptyset
		p	P	P	P

Architecture (analysis)



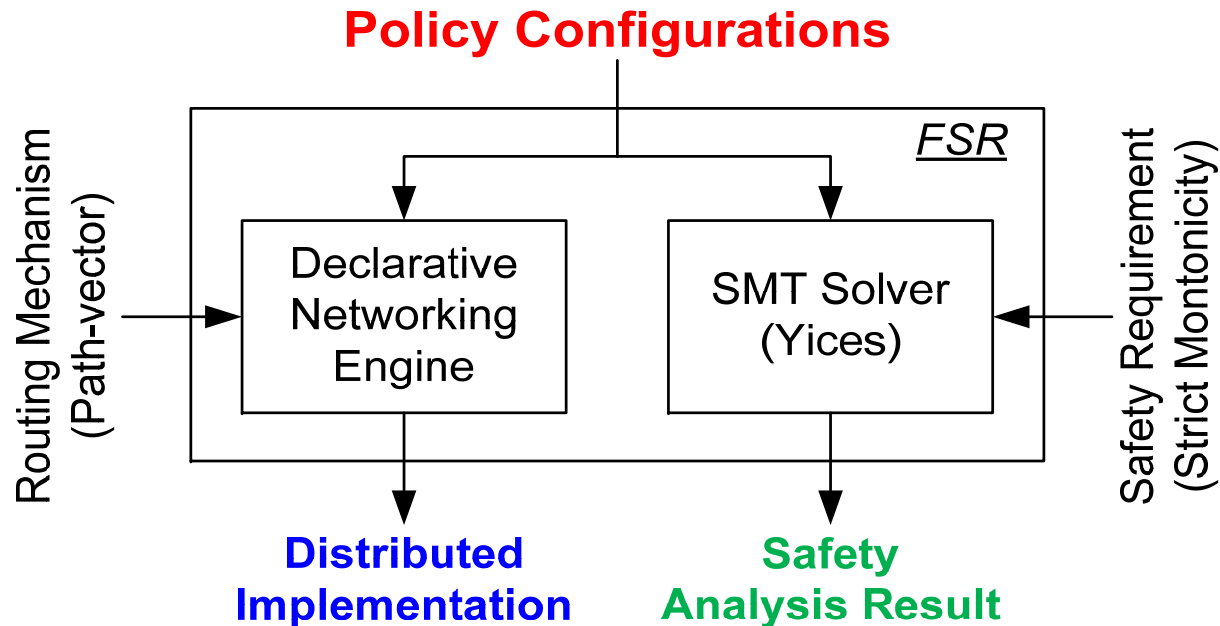
- **Contribution #1: Reduction from safety analysis to SMT solving**
 - Strict monotonicity (SM) implies safety [SIGCOMM 03, 05]
 - Convert policy configuration into Yices (SMT solver) constraints
 - Check if the constraints are satisfiable

Architecture (analysis)



- **Contribution #1:** Reduction from safety analysis to SMT solving
- **Output: Safety analysis result**
 - SAT – *automatically* generated proofs
 - UNSAT – an unsatisfiable core (*pinpoint configuration errors*)

Architecture (implementation)



- **Contribution #2: Provably correct distributed implementation**
 - Generalized path vector + **policy configuration**
 - Generation of *Declarative Networking* program [SIGCOMM 05]
 - Correctness proof for the policy -> NDlog translation

Demo: GR + HopCount

Applications Places System wenchaoz Thu Aug 18, 2:17 AM

RapidNet Demo Visualizer

Protocols NDlog Rules Generated Code Automatic SMT Solver

Simulation Parameters

Protocol	Metarouting
Mobility Model	Constant Position
Nodes Count	40
Arena	800m X 800m
Duration	100
Speed	0m/s to 0m/s

Step

<< >>

Average Bandwidth Utilization

Legend

- Low (Bandwidth)
- Medium
- High
- Very High

45.0

wenchaoz@wenchaoz-... RapidNet Demo Visual... Update Manager

Demo: GR + HopCount

Applications Places System wenchaoz Thu Aug 18, 2:17 AM

RapidNet Demo Visualizer

Protocols NDlog Rules Generated Code Automatic SMT Solver

Simulation Parameters

Protocol	Metarouting
Mobility Model	Constant Position
Nodes Count	40
Arena	800m X 800m
Duration	100
Speed	0m/s to 0m/s

Step

<< >>

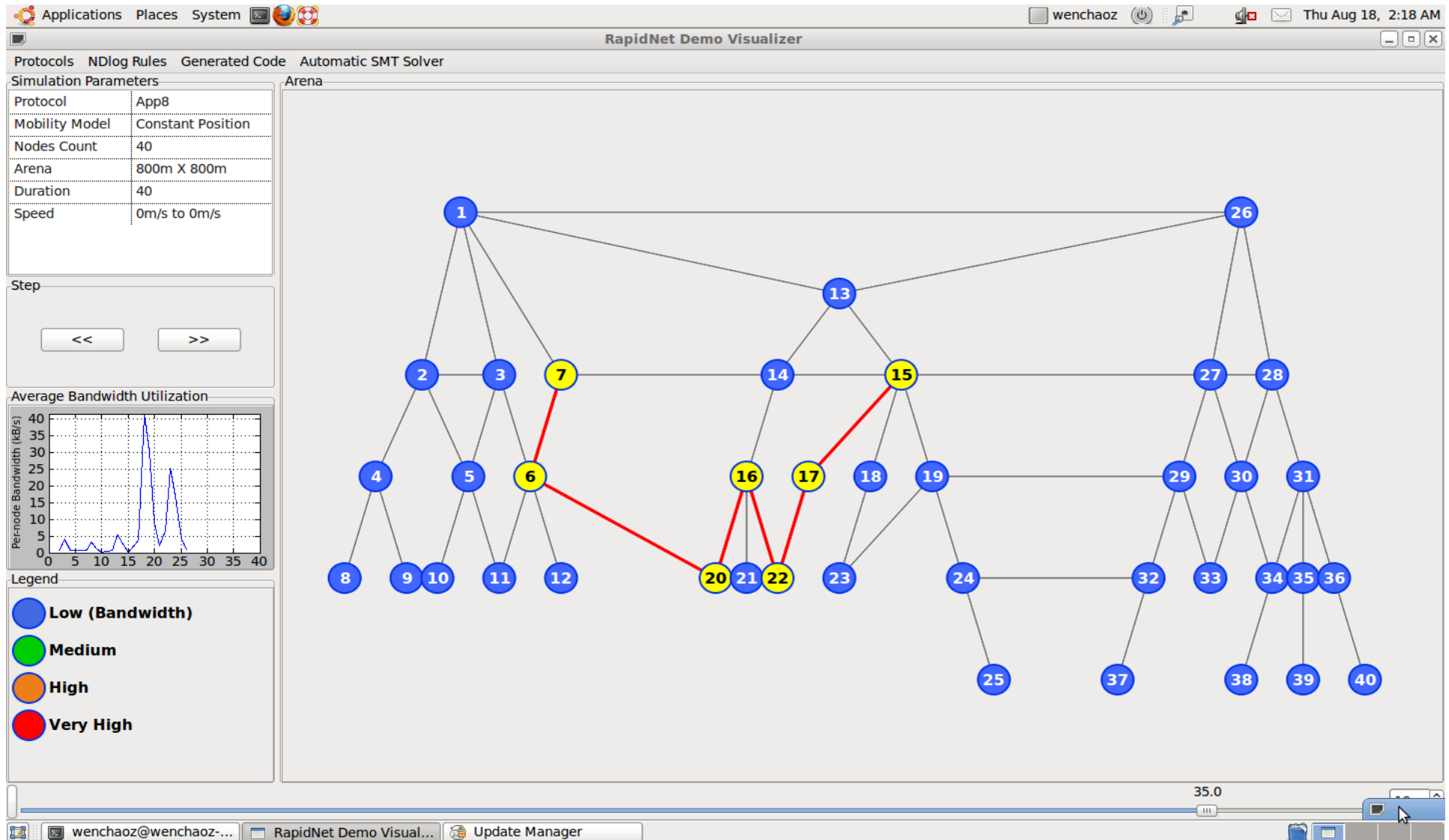
Average Bandwidth Utilization

Legend

- Low (Bandwidth)
- Medium
- High
- Very High

45.0

Demo: GR (wrong) + HopCount



Demo: Pinpoint Mis-configuration

The image shows a desktop environment with a window titled "RapidNet Demo Visualizer" and a terminal window titled "Yices Result".

The "RapidNet Demo Visualizer" window displays a map of Europe with a network graph overlaid. Nodes are represented by numbered circles. Nodes 7, 27, and 32 are highlighted in pink. The map shows connections between various cities across Europe, including London, Paris, Berlin, and Rome.

The "Yices Result" terminal window shows the output of the Yices solver. The output includes the following text:

```
./yices -e smt/02InstanceAnalysis/01AS1755 Gadget/Metarouting.y > yic
./parse_result smt/02InstanceAnalysis/01AS1755 Gadget/Metarouting.y |
wenchaoz@wenchaoz-laptop:/home/netdb/project/sigcomm-demo/rapidnet_vis
tanceAnalysis/01AS1755 Gadget/Metarouting.y > yices_result.txt
wenchaoz@wenchaoz-laptop:/home/netdb/project/sigcomm-demo/rapidnet_vis
2InstanceAnalysis/01AS1755 Gadget/Metarouting.y | more
unsat
unsat core : Node 7 9 375
Node 7:
(assert+ (< (sn0n10n12n99) (sn0n10n12n99) )) ;; 98
(assert+ (< (sn0n10n12n99) (sn0n10n12n99) )) ;; 295
Node 27:
(assert+ (< (sn26n21n20n22n99) (sn26n21n20n22n99) )) ;; 116
(assert+ (< (sn26n21n20n22n99) (sn26n21n20n22n99) )) ;; 349
Node 32:
(assert+ (< (sn31n30n8n74n99) (sn26n31n30n8n74n99) )) ;; 140
(assert+ (< (sn31n6n0n10n12n99) (sn31n30n8n74n99) )) ;; 375
wenchaoz@wenchaoz-laptop:/home/netdb/project/sigcomm-demo/rapidnet_vis
```



Summary



- **Bridging formal theories and implementation**
 - Unified policy specifications (based on routing algebra)
 - Automated safety analysis (reduced to constraint solving)
 - Provably correct distributed implementations

Thank you...

**FSR: Formal Analysis & Implementation Toolkit for
Safe Inter-domain Routing**



<http://netdb.cis.upenn.edu/fvr/>