Large-Scale Network Topology Emulation and Inference

Erik Rye

Naval Postgraduate School
Monterey, CA

31 March 2015

Outline

Ark and Ground Truth

Ark

- Ark is a state-of-the-art system for gathering Internet topologies
- However, as well all know, topology limited by vantage points, filtering, inferences, and heuristics – lots of noise and room for error
- Coming from the math community, this is all foreign and strange!
- Perennial problem in community: no ground truth
- Very unsatistfying for graph/math types

Our basic insight:

1. Let's create our own ground truth
2. And (try to) not fall down the simulation rathole

Ark and Ground Truth

Ark

- Ark is a state-of-the-art system for gathering Internet topologies
- However, as well all know, topology limited by vantage points, filtering, inferences, and heuristics – lots of noise and room for error
- Coming from the math community, this is all foreign and strange!
- Perennial problem in community: no ground truth
- Very unsatistfying for graph/math types

Our basic insight:

1. Let's create our own ground truth
2. And (try to) not fall down the simulation rathole

Ground Truth

If we had ground truth

- Understand how well our inferences are doing
- Understand root causes of traces gathered in Ark
- Develop new probing/inference algorithms (e.g., IPv6)

Hence, we sought to understand how far we could push network emulation for the purpose of creating ground truth

Ground Truth

If we had ground truth

- Understand how well our inferences are doing
- Understand root causes of traces gathered in Ark
- Develop new probing/inference algorithms (e.g., IPv6)

Hence, we sought to understand how far we could push network emulation for the purpose of creating ground truth

Motivation and Methodology

- Powerful confluence:
  - Hardware is cheap and capable +
  - Ability to virtualize router hardware +
  - Run real vendor software images, e.g., Cisco IOS
  - = emulate non-trivial networks
- Why?
  - Emulation reveals crucial implementation details
  - Automation permits experimentation over large parameter space
- For DHS Network Mapping:
  - Create our own "ground truth" to evaluate inference utilities and our own algorithms
  - Automate topology inference from as many vantage points as possible...
  - ...in as many topologies as possible

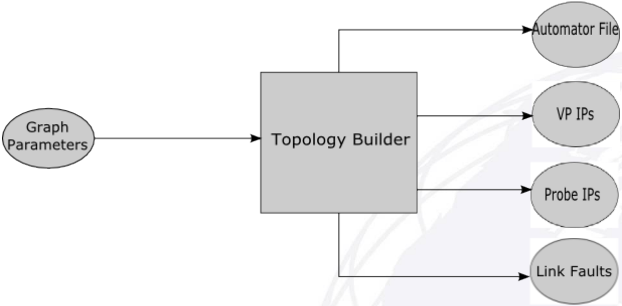
Emulated Router Inference Kit (ERIK)

1. Generate network topologies (Internet-like, reduced, flat, random)
2. Generate each individual router configuration (including IP addressing) based on generated topology (and policy)
3. Configure Dynamips hypervisor to run router images and interconnect virtual routers and switches
4. Run automated testing (e.g. topology inference) exhaustively (e.g. from all vantage points)
5. Automate faults and scenarios
6. Record test/scenario output

A tool for emulated fuzz testing

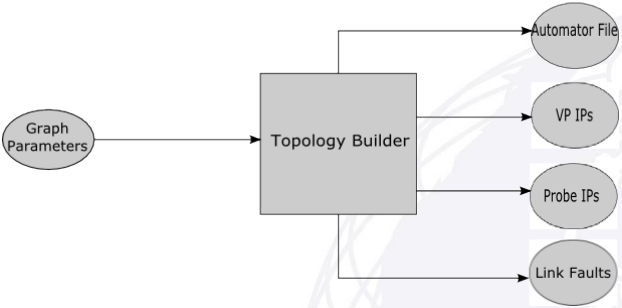
- Focus is on ability to emulate any topology - rather than realism of topologies themselves
  - Objective is not-realism dependent
  - Expose implementation-specific behaviors
- Topology
  - Explore more of the graph space
  - Compare topology generation models
- Vantage Points
  - Evaluate importance and effects of VP selection
  - Single vs. mulitple VPs
- Policy
  - Examine effects of policy implementations/changes
- Faults
  - Study effects of faults on topology inference performance
  - Evaluate resiliancy of topologies under failure scenarios

ERIK Topology Generation



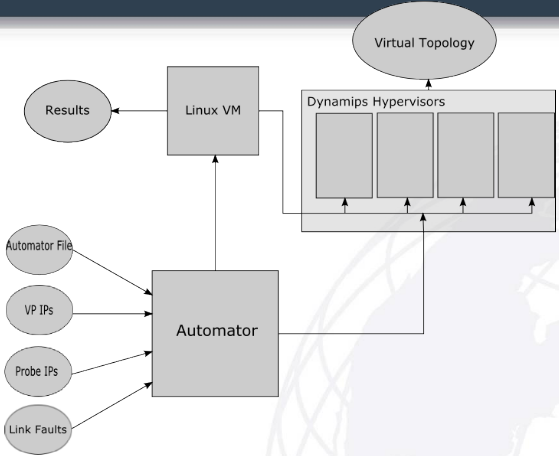
- Topology generation parameters:
  - Topology model - Barabási-Albert, Waxman, Random, Tiered
  - Parameterized Internet-like "tiered"
  - Reduced real graphs (reduce the number of nodes, maintain basic graphic metrics – lots of cool stuff here, ask me about it offline)
- Each AS modeled by a single Cisco 7200 series router.

ERIK Topology Generation



- Tiered model policy: customer > peer > provider
- We implement this policy using route-maps during the configuration-generation

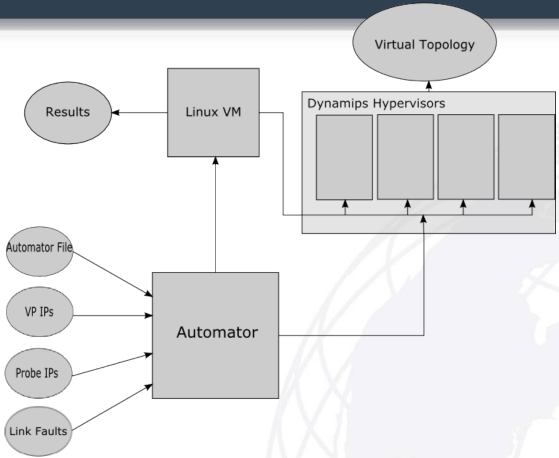
ERIK Automation



- After initialization, the ERIK begins testing by coordinating with a virtual Linux machine
- The VM is connected to an AS in the topology
- In our testing, VM uses *scamper* to probe an IP address in each of the ASes in the topology.

NAVAL POSTGRADUATE SCHOOL

ERIK Automation



- Three rounds: before, during, and after fault injection.
- For added realism and load we carry 50,000 BGP routes
- Faults are links that fail (causes routing churn and behaviors of interest)
- Automation iterates over all vantage points, recording results

ERIK Performance

- We have scaled ERIK up to 300 emulated routers in complex topologies
- ERIK is stable in our environment, but not packaged for redistribution (yet)
  - Hardware-specific parameters for time for routing tables to converge, time to complete initial topology setup

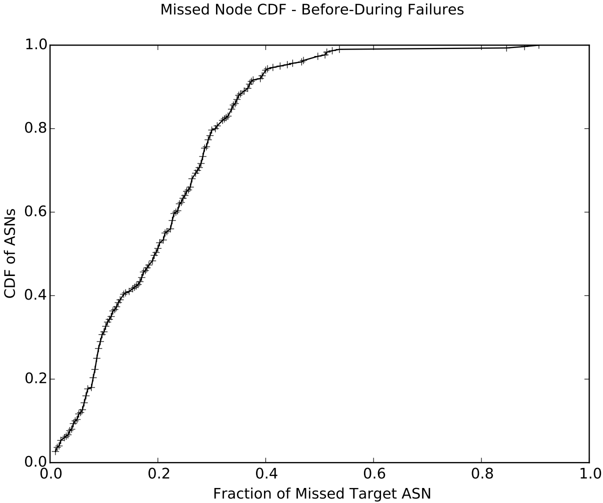
An example topology

- Case study: 15 tier 1, 45 tier 2, 240 customer ASes, connected by 676 edges
- Topology graph is connected physically and by policy, though disconnections occur from failures
- Links selected for failure are the 15 edges in the graph with the highest edge betweenness centrality

Results

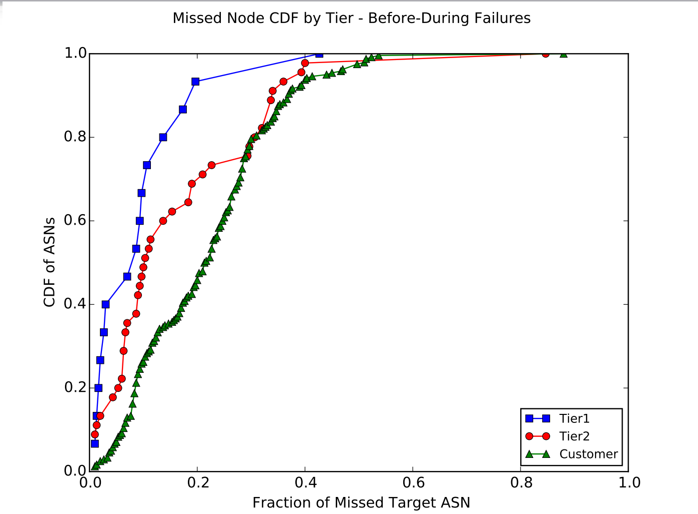
- During the *before-failures* probing rounds, all ASes were discovered from all VPs.
- Inferred graphs not source-based trees from VP
  - Most cycles occur within tier 1 backbone
- During the *failures* scenario, we see a wide variance in the number of ASes that were not discovered by our *scamper* probing.
  - 3 to 272 ASes not discovered; mean of 61 ASes missed.
- In the *after-failures* probing round, disconnections are evident
  - 1 to 285 ASes missed; mean of 5 ASes missed.

Results



- During failures, over 50% of VP probes miss more than 20% of ASes

Results



- By tier, customer nodes miss more topology during failures than others

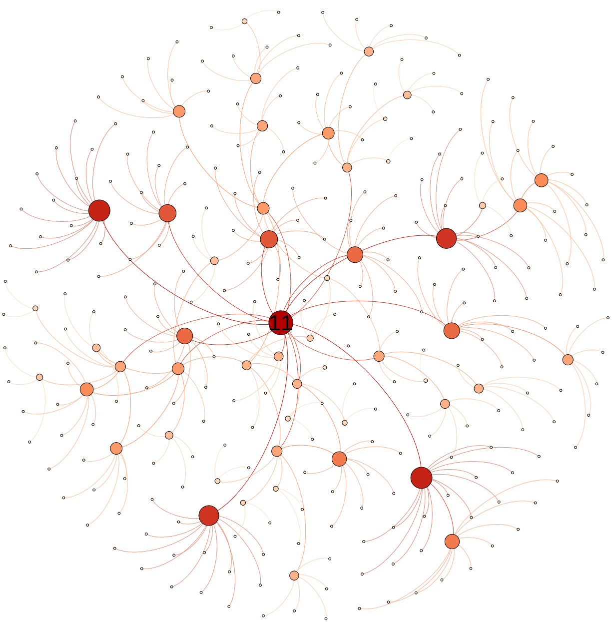
A closer look

AS 11, a tier 1 AS, is an extremely central vertex in the graph - 7 of the top 15 links with highest edge betweenness centrality incident

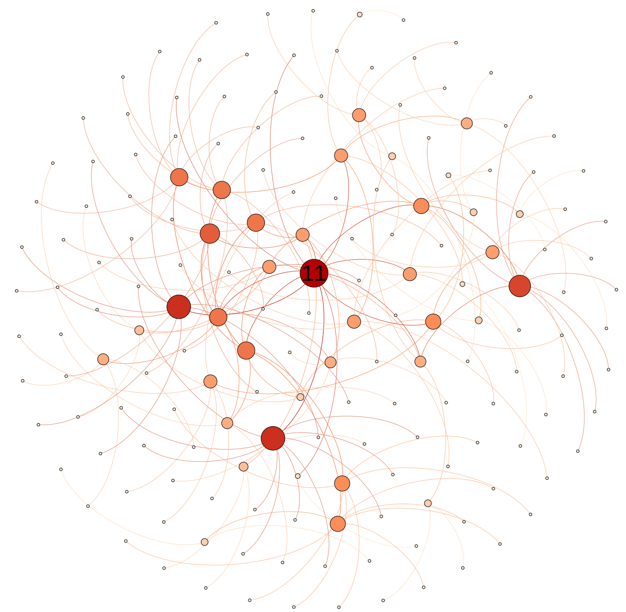
- Before failures scenario, all 300 ASes reached
- During failures scenario, nearly half of preferred routes must be updated. Only 172 AS destinations discovered.
- After failures scenario, 297 ASes reached.
- 97 different edges in the after-failures inferred graph than in the before-failures inferred graph.
- Though the number of vertices inferred before and after failures scenario are close, the resultant inferred graphs are quite different

NAVAL POSTGRADUATE SCHOOL

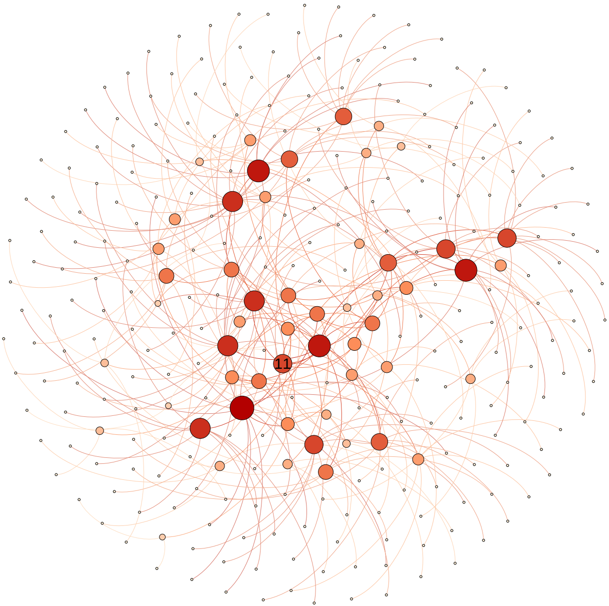
Before Failures Scenario



During Failures Scenario



After Failures Scenario



Way ahead

- Many opportunities (for us and others) to leverage ERIK going forward:
    - Scalability - using clusters of machines, can the number of emulated routers be increased by an order of magnitude?
    - Intra-AS/Inter-AS combined topologies
    - Emulation of JunOS topologies to enable direct IOS/JunOS comparisons (do we obtain the same topologies? what about under faults?)
    - Customer cone and BCP38 source address validation (anti-spoofing)
    - Validate and reproduce results obtained from Ark probing in a controlled environment
    - Explore IPv6 topology inference

ERIK Conclusions

- ERIK has the potential to be an efficient and effective tool for automating network topology testing
  - Cover much more of the graph space with arbitrary policy complexity.
  - Understand router/scenario implementation specific details that influence results
  - Model hypothetical scenarios to observe topology resilience to failures
  - Can be adapted to problems beyond topology inference