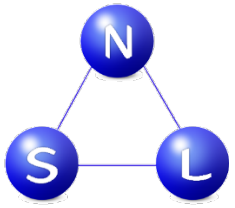



Networked Systems Laboratory

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

# Toward Online Testing of Federated and Heterogeneous Distributed Systems

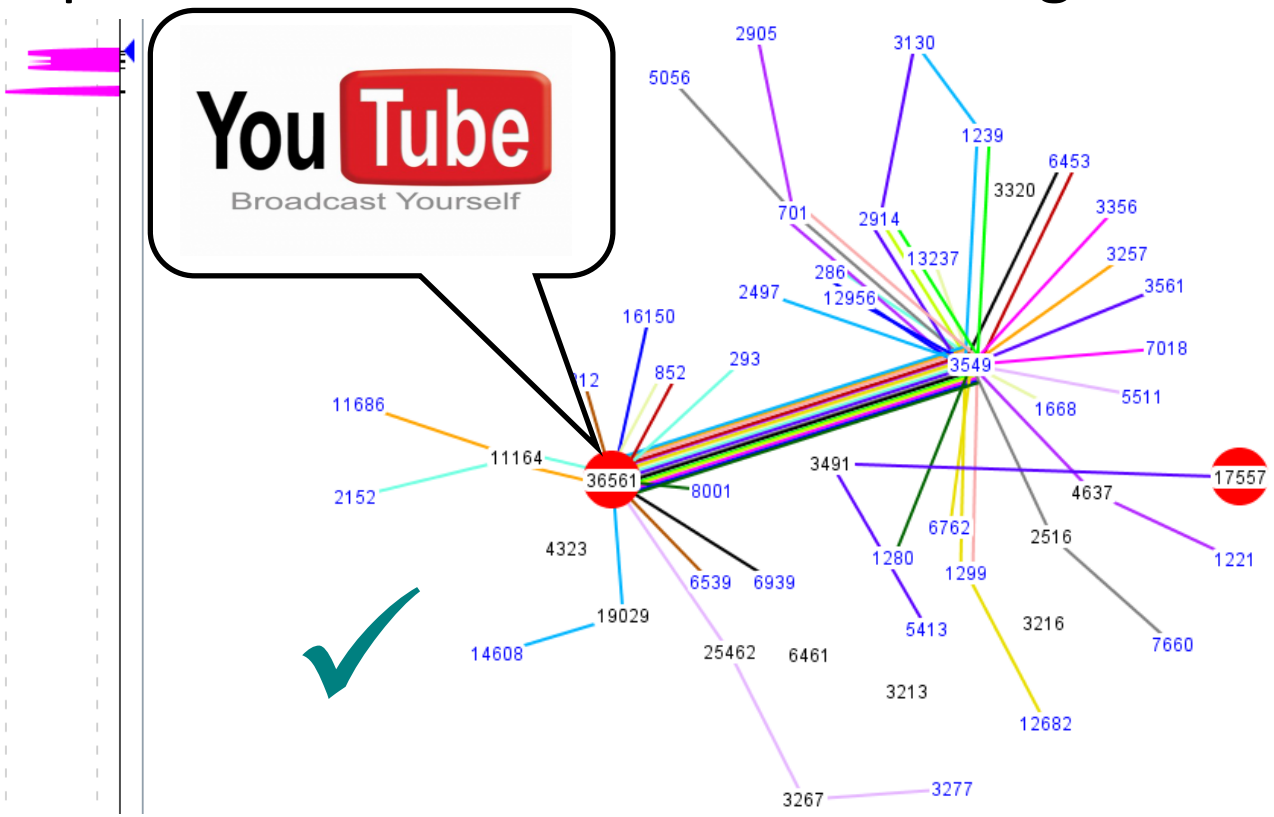
Marco Canini
EPFL, Switzerland

Joint work with: Vojin Jovanović, Daniele Venzano, Boris Spasojević, Olivier Crameri, and Dejan Kostić

Work supported by the European Research Council

# Federated, Heterogeneous Distributed Systems

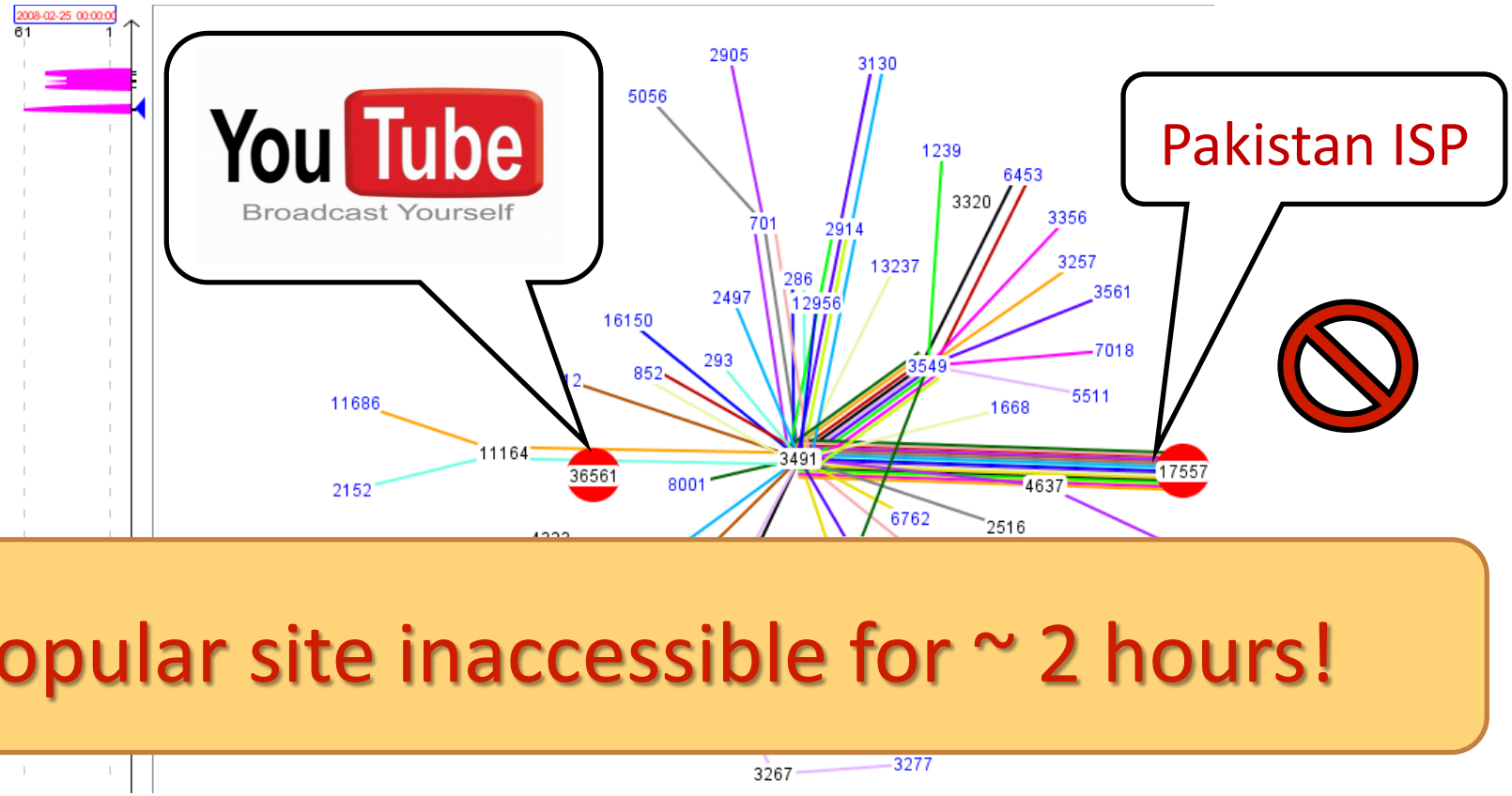
- Multiple administrative domains
- Several interoperable implementations
- Example: Internet inter-domain routing




[BGPlay]

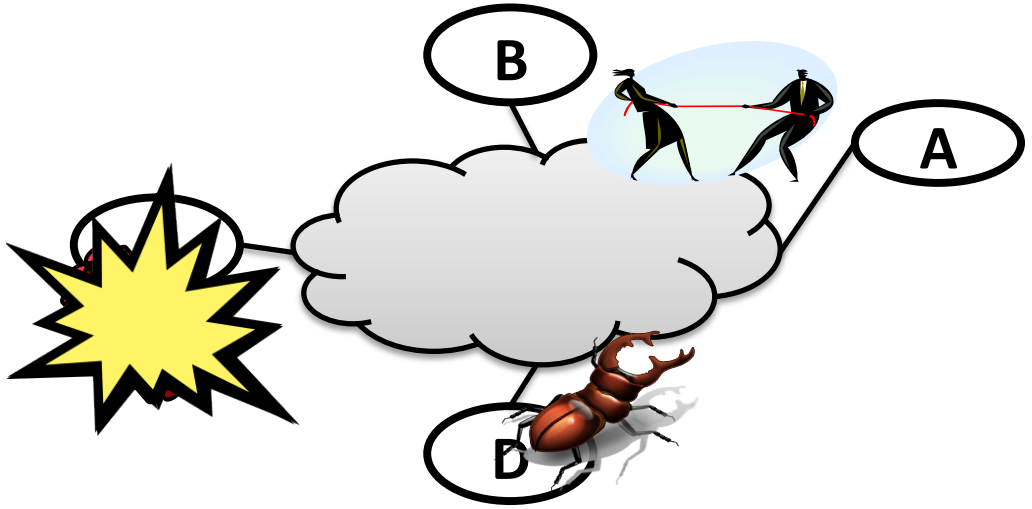
# Internet Routing is Unreliable

- Operator mistakes, bugs, …
  - Origin misconfiguration: Pakistan/YouTube 2008 incident

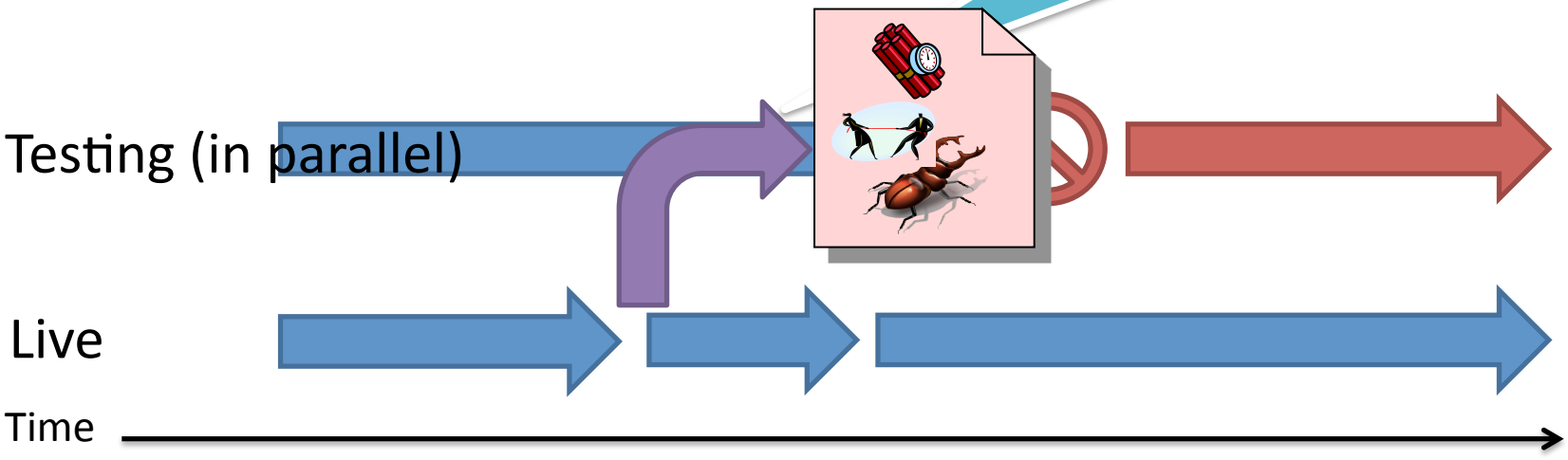



**Popular site inaccessible for ~ 2 hours!**

# How to improve reliability?




Account for current state, code and configuration



# Goal of online testing

- Systematically explore system behavior
  - Detect node actions that lead to faults
  - Continuously and automatically
  - Alongside but in isolation from live environment
  - Accommodate constraints of federated and heterogeneous distributed systems
    - No unrestricted access to node state and configuration
    - Difficult to have source/binary code of all nodes

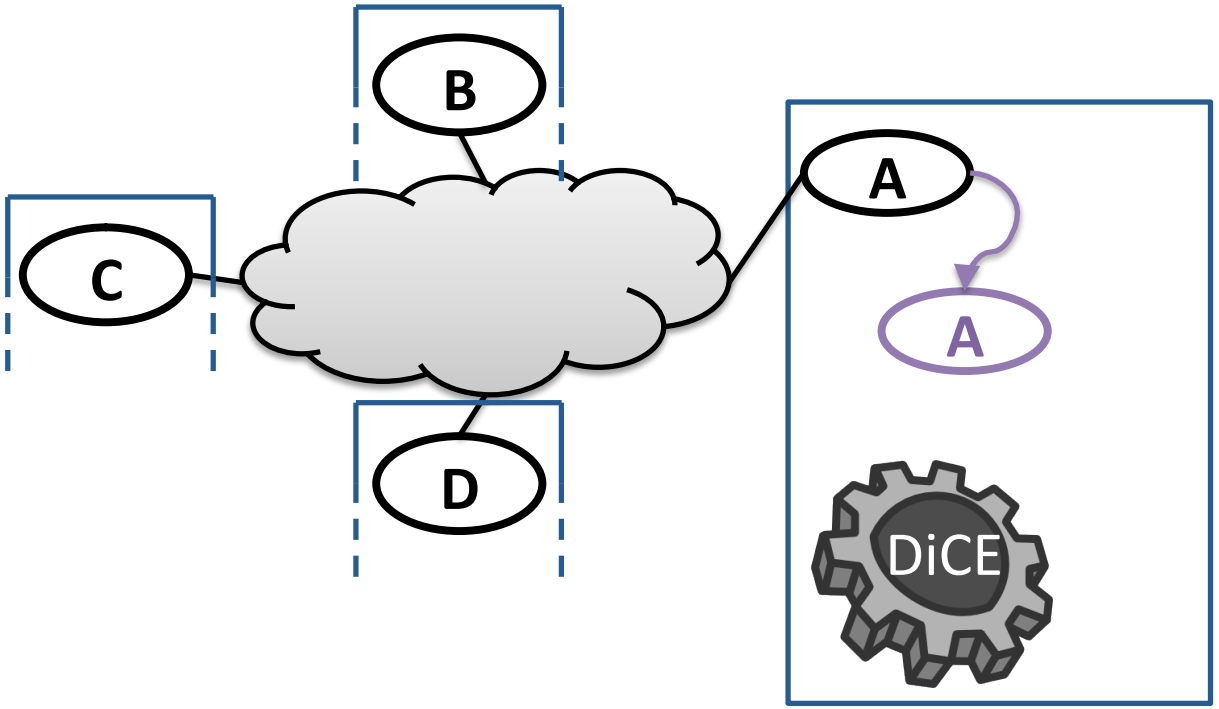
# *Toward reaching* ~~How to reach~~ our goal?

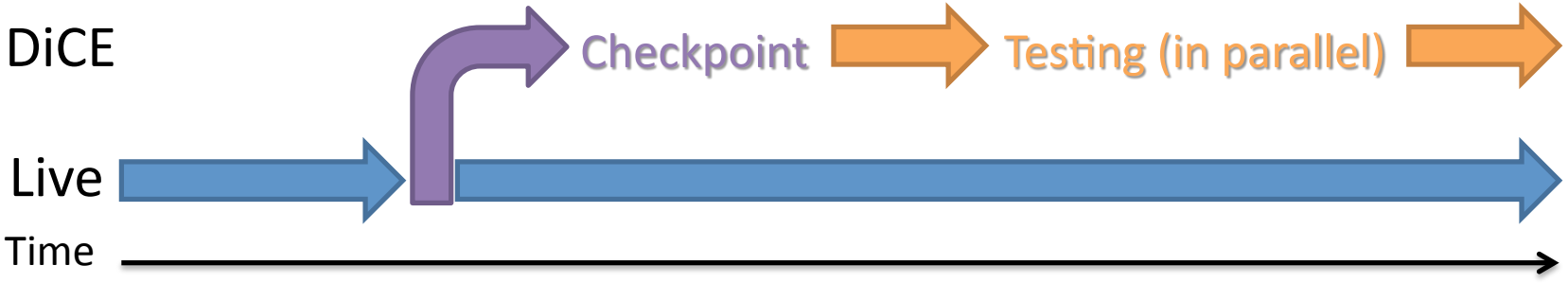
1. Drive system behavior
   - Aggregate result of interleaved node actions
   - Actions driven by paths taken through node's code

   ⟹ **Subject node's code to inputs that exercise possible actions**

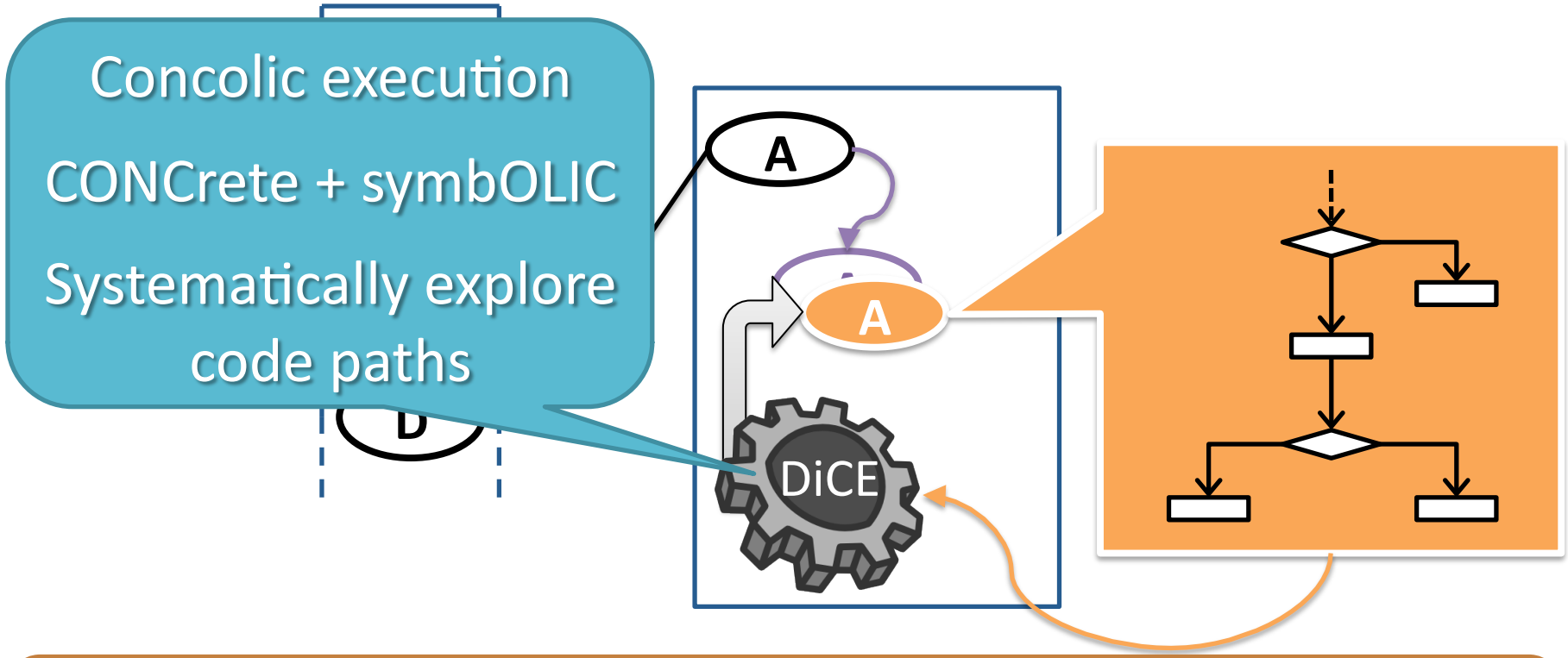
2. Observe consequences of node actions
   - System-wide perspective
   - Check faults while preserving confidentiality

# DiCE



# DiCE

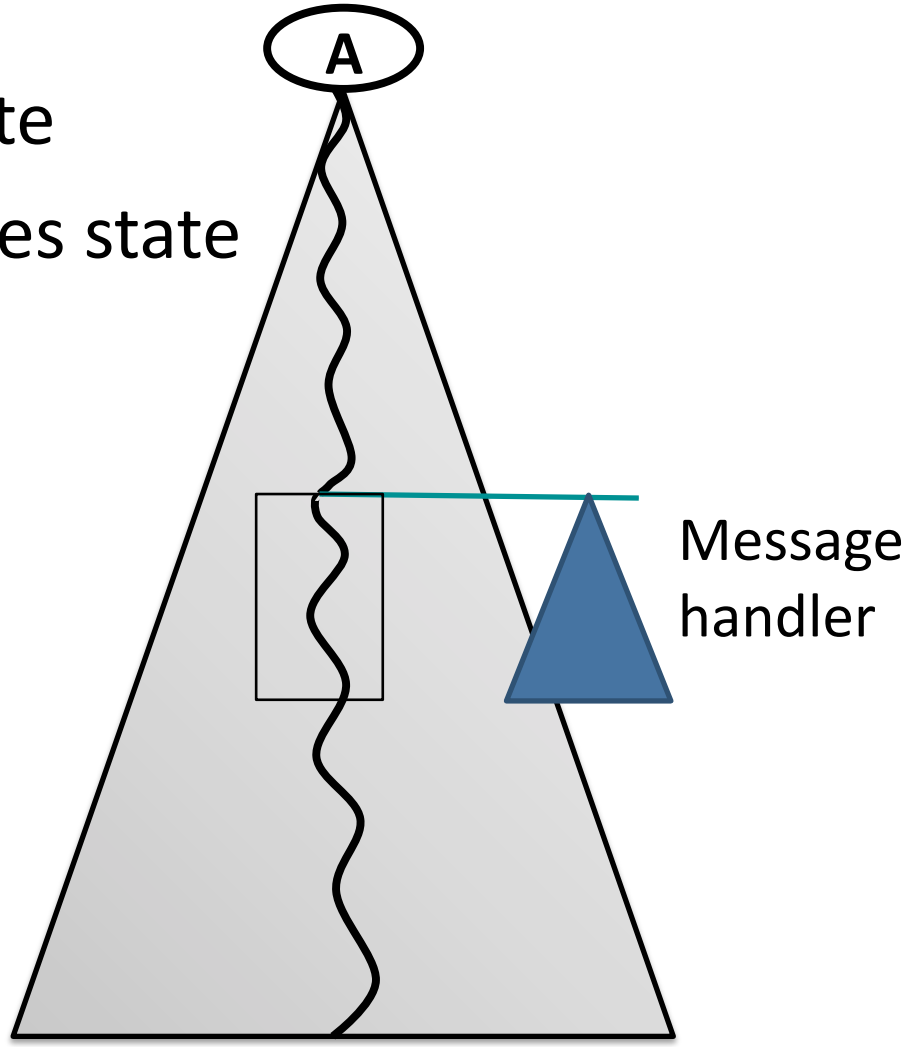



Exponential number of paths!

Time

# Managing Path Explosion

- Explore from current state
- Localize code that changes state
  - e.g., message handlers
- Inject small-sized inputs

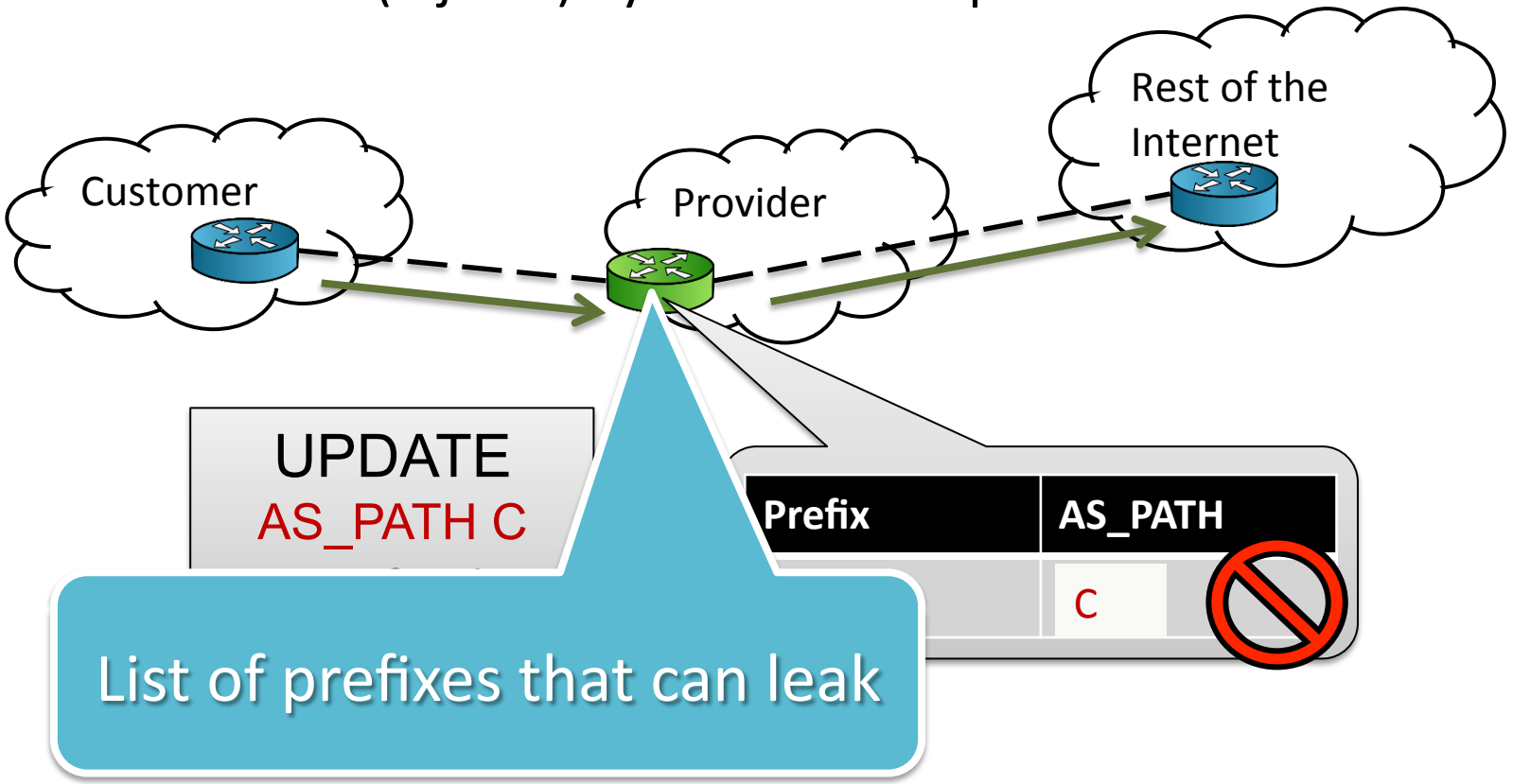


# DiCE prototype for BGP

- Integrated DiCE in BGP module of BIRD 1.1.7
  - Open source router, coded in C
- Use fork() to take/clone checkpoints
- Exploring BGP behavior
  - UPDATE messages main drivers of state change
    - Announced routes
    - Path attributes
- Concolic execution instruments code
  - Use only for testing → negligible impact on live system

# Detecting origin misconfiguration

- Check: routing tables polluted in external ASes?
  - Route leaks (hijacks) by customer or provider



# Going forward

- Initial building block

  $\Longrightarrow$ **Ability to explore node actions (in isolation)**

- Next

  $\Longrightarrow$ **Observe consequences on system-wide state**

  – Isolated online testing harness
  – Check states w/o exposing private information
  – More info in our position paper [LADIS ’10]

- Thank you! Questions?

# Micro benchmarks

- CPU overhead
  - Metric: BGP updates per second
  - Stress test during RIB load
    - Baseline: 15.1 – W/ exploration: 13.9 – Impact 8%
  - Realistic test during trace replay
    - Negligible impact
- Memory overhead
  - Cloned process has 37% overhead on average
    - We did not attempt to minimize instrumentation