Survivability Analysis of Networked Systems

Somesh Jha

joint work with Oleg Sheyner and Jeannette Wing

Computer Sciences Department
University of Wisconsin
Madison, WI

Relationship to analyzing compositions of COTS

- A system is typically composed of several COTS components
  - (step 1) analyze components individually
  - (step 2) inputs
    - » vulnerabilities of individual components
    - » connectivity information
  - output
    - » attack graph for the entire system
    - » how can intruder exploit vulnerabilities in individual hosts to achieve their desired goal?

- Analogy to networks
  - hosts ISA components
  - networks ISA system composed of components

Survivability

- What if
  - a cyberhacker brings down the nation's power grid?
  - an act of Mother Nature causes the international banking network to fail?

- Critical infrastructures
  - Utilities: gas, electricity, nuclear, water, …
  - Communications: telephone, networks, …
  - Transportation: airlines, railways, highways, …
  - Medical: emergency services, hospitals, …
  - Financial: banking, trading, …

Survivability

- A system is survivable if it can continue to provide end services despite the presence of faults.

- Faults
  - Accidental or malicious
  - Not necessarily independent

  $\Rightarrow$ Finer-grained reliability analysis is required.

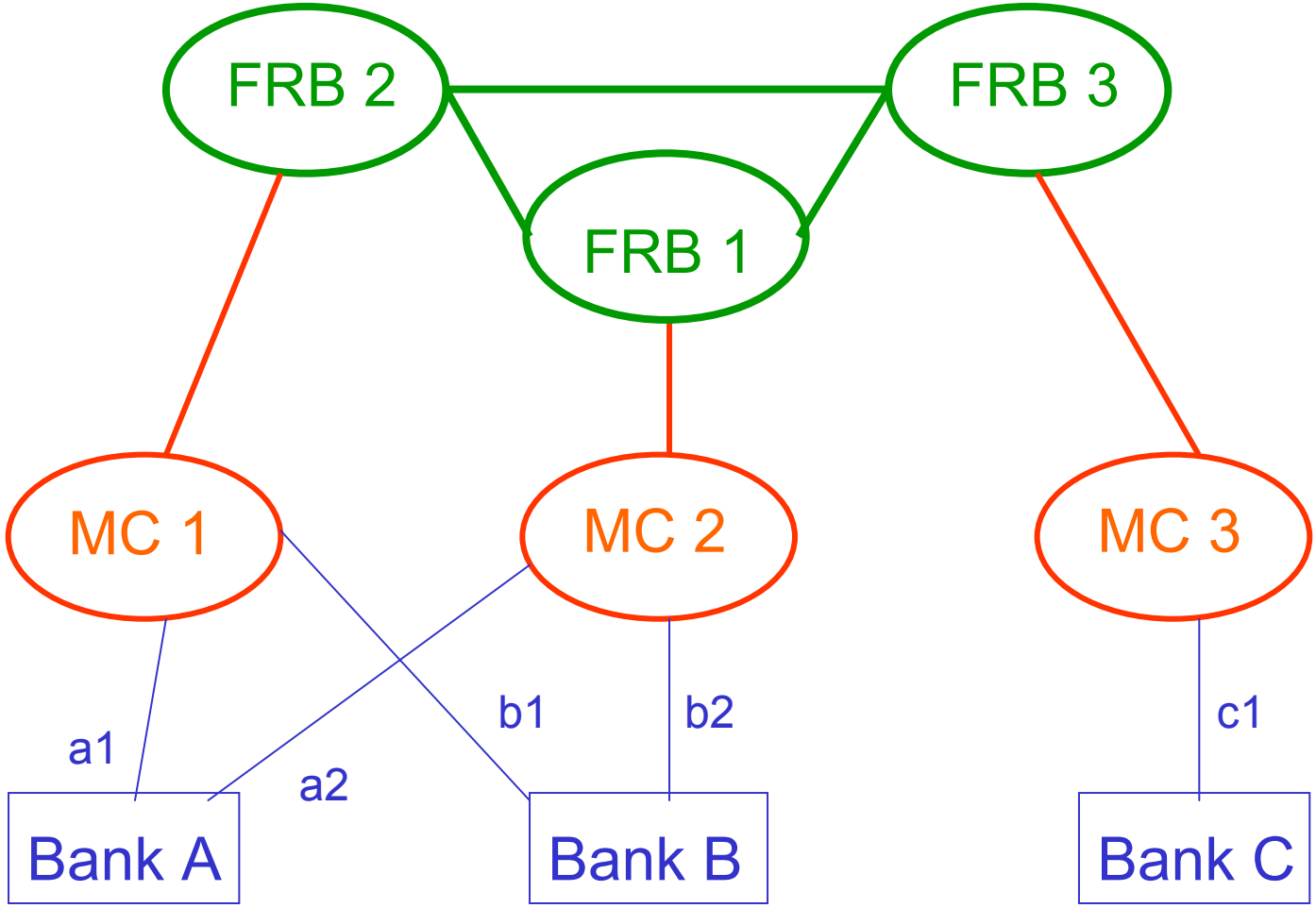
- Service-oriented
  - Exploit semantics of application

  $\Rightarrow$ Not all network nodes and links are treated equally.

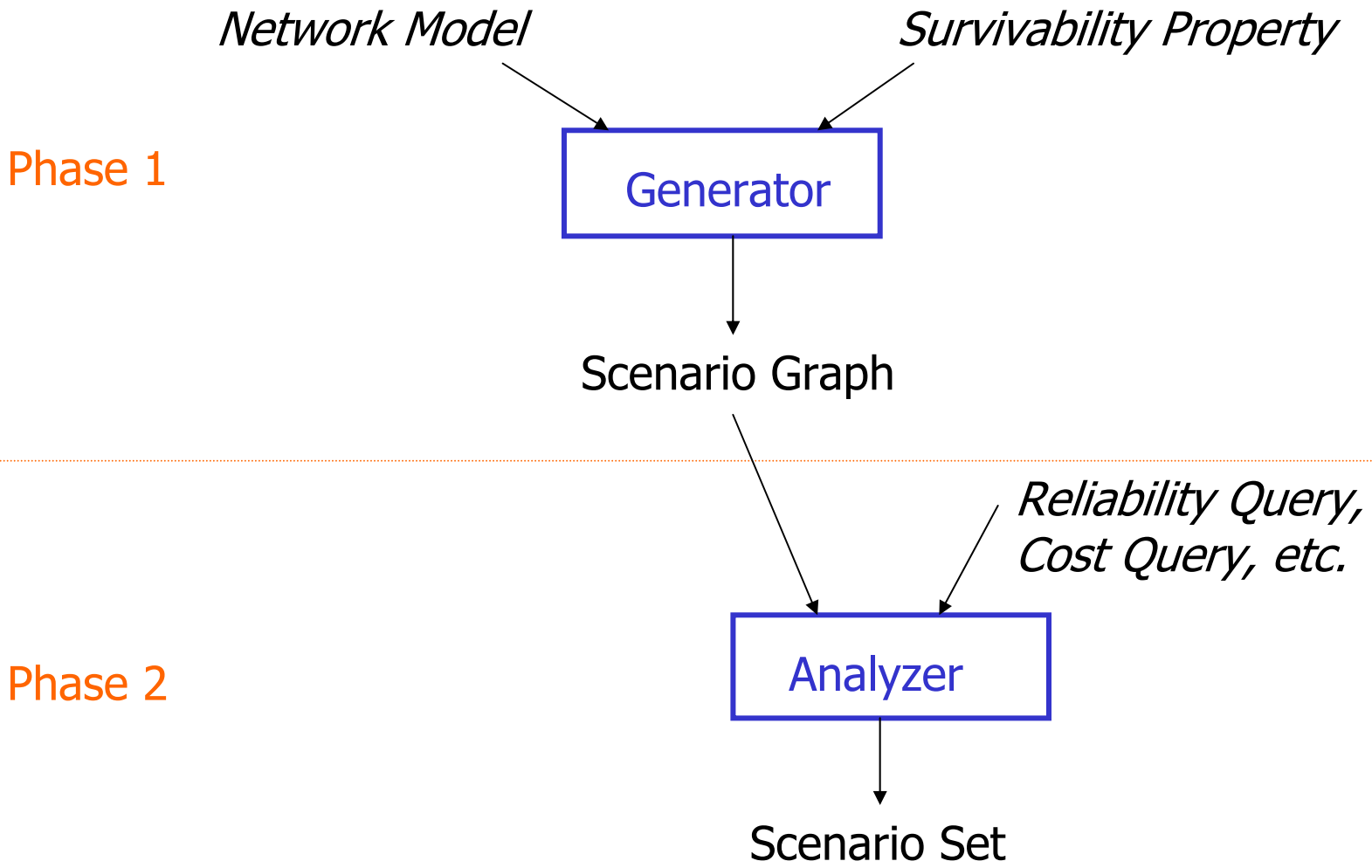
Foundational Questions

- What is the difference between models for survivability and those for
  - Fault-tolerant distributed systems?
  - Secure systems?
- Our starting point:
  - Independence assumption goes out the window.
  - Cost must be included in the equation.

Simple Example: A Banking System



Overview of Our Method



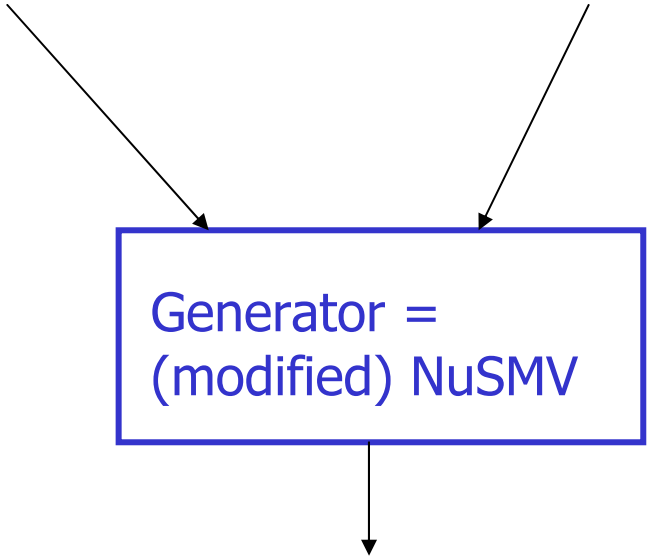
Phase 1

Network Model =

A set of concurrently executing Finite State Machines.

Survivability Property =

A predicate in CTL.



Scenario Graph =

A set of related examples.

Network Model

- Processes
  - Nodes and links are processes (i.e., FSMs)
    - banks, money centers, federal reserve banks, and links
  - Communication via shared variables (i.e., finite queues)
    - representing channels, and hence interconnections.
- Failures
  - Faults represented by special state variable
    - fault:{normal, failed, intruded}
  - Links and banks can fail at any time
    - Failed link blocks all traffic.
    - Failed bank routes all checks to an arbitrarily chosen money center.
  - Money centers and federal reserve banks do not fail.

Survivability Properties

- Fault-related
  - Money never deposited into wrong account.
    - AG($\neg$error)
- Service-related
  - A check issued eventually clears.
    - AG(checkIssued $\rightarrow$ AF(checkCleared))

Scenario Graphs

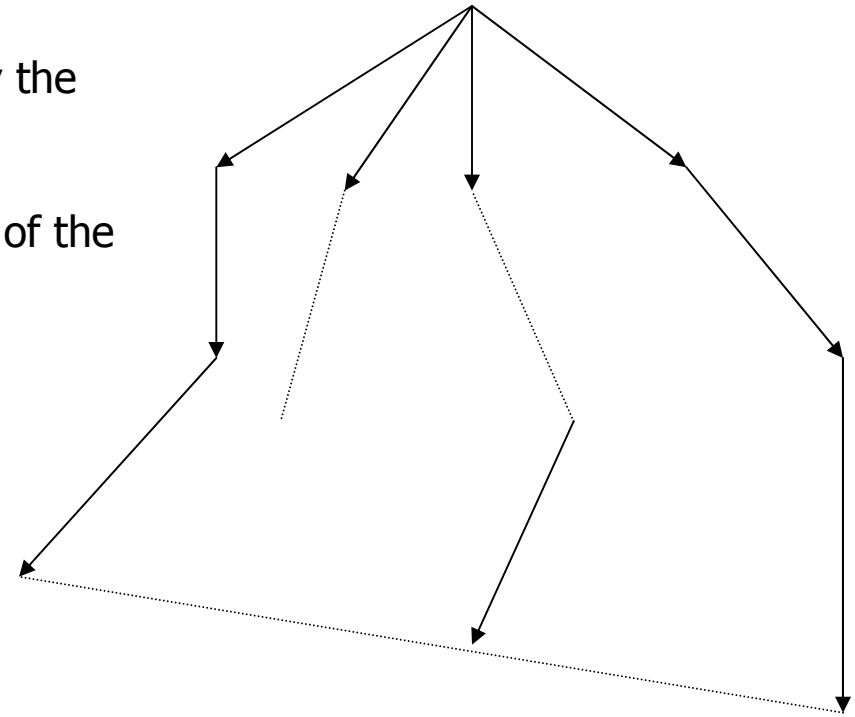
- Given a state machine, M, and a property, P, a scenario graph is a concise representation of the set of traces of M with respect to P.
  - P = fault property
    - A fault scenario graph represents all system traces that end in a state that does not satisfy P.
  - P = service property
    - A service success (fail) scenario graph represents all system traces in which an issued service successfully finishes (fails to finish).

Output: *Fault* Scenario Graph

Intuition:

- Each "counterexample" spit out by the model checker is a scenario.

- Survivability property gives a slice of the model.



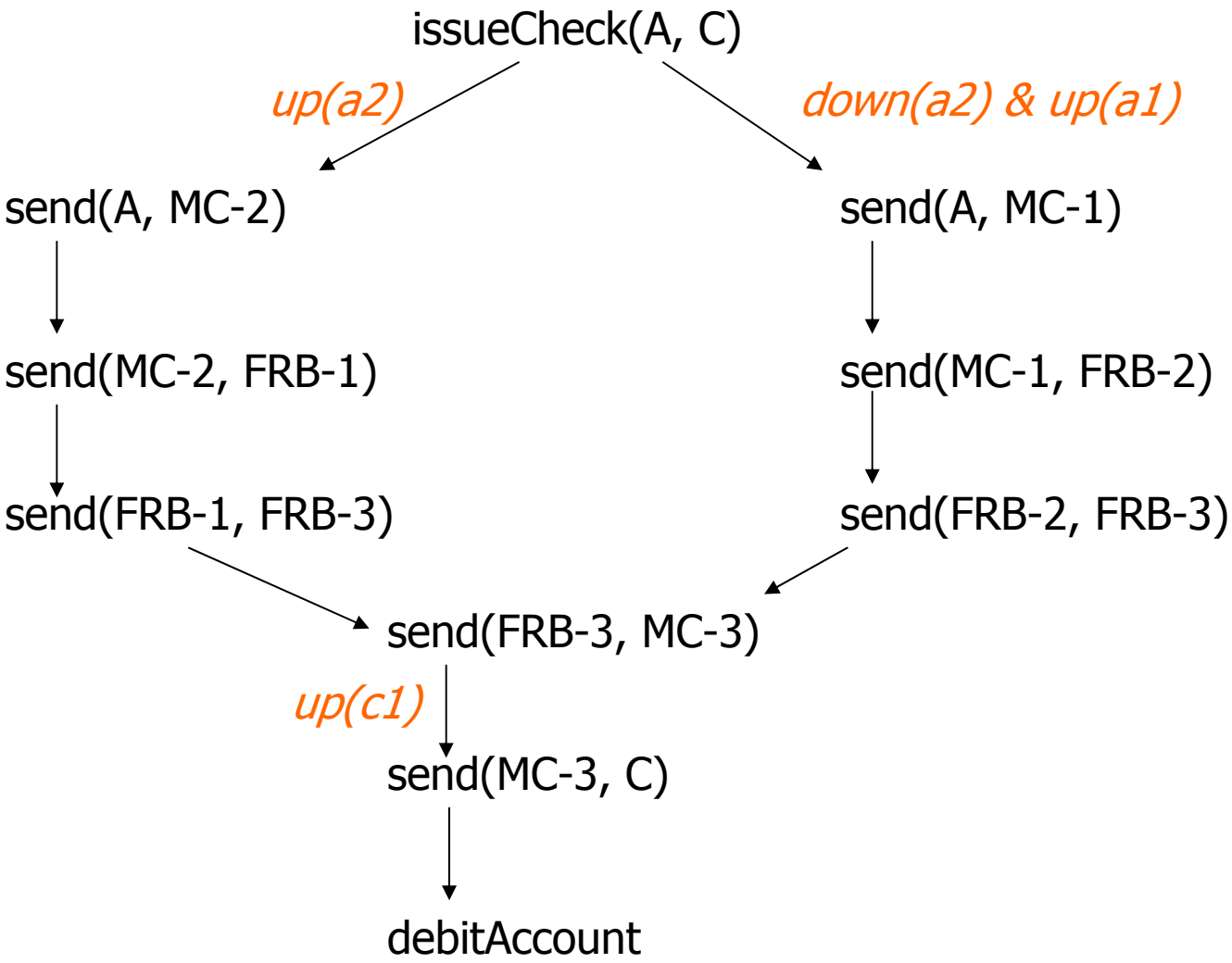
Each path is a scenario of how a *fault* can occur.

Survivability Properties

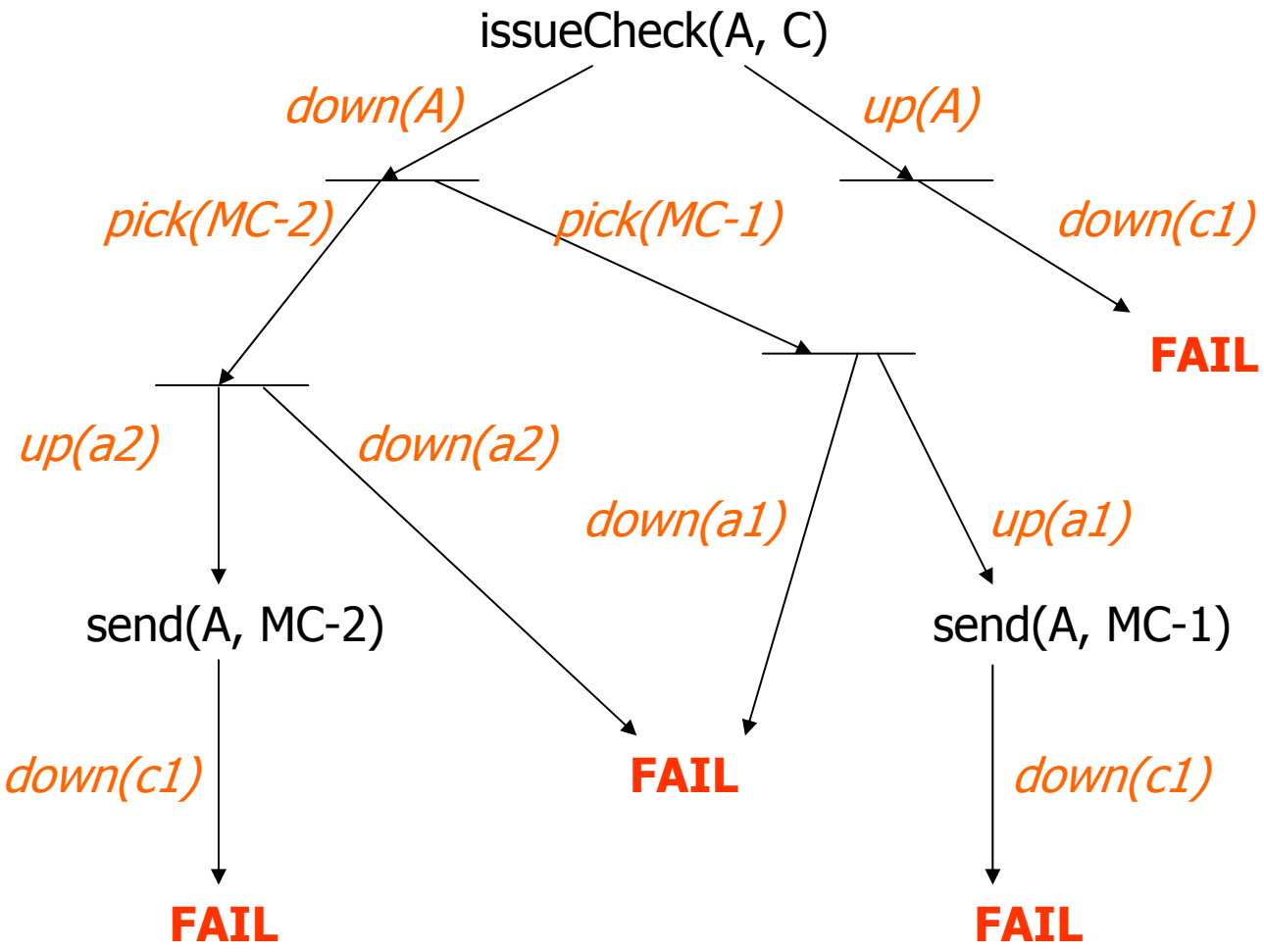
- Fault-related
  - Money never deposited into wrong account.
    - AG($\neg$error)

- Service-related
  - A check issued eventually clears.
    - AG(checkIssued $\rightarrow$ AF(checkCleared))

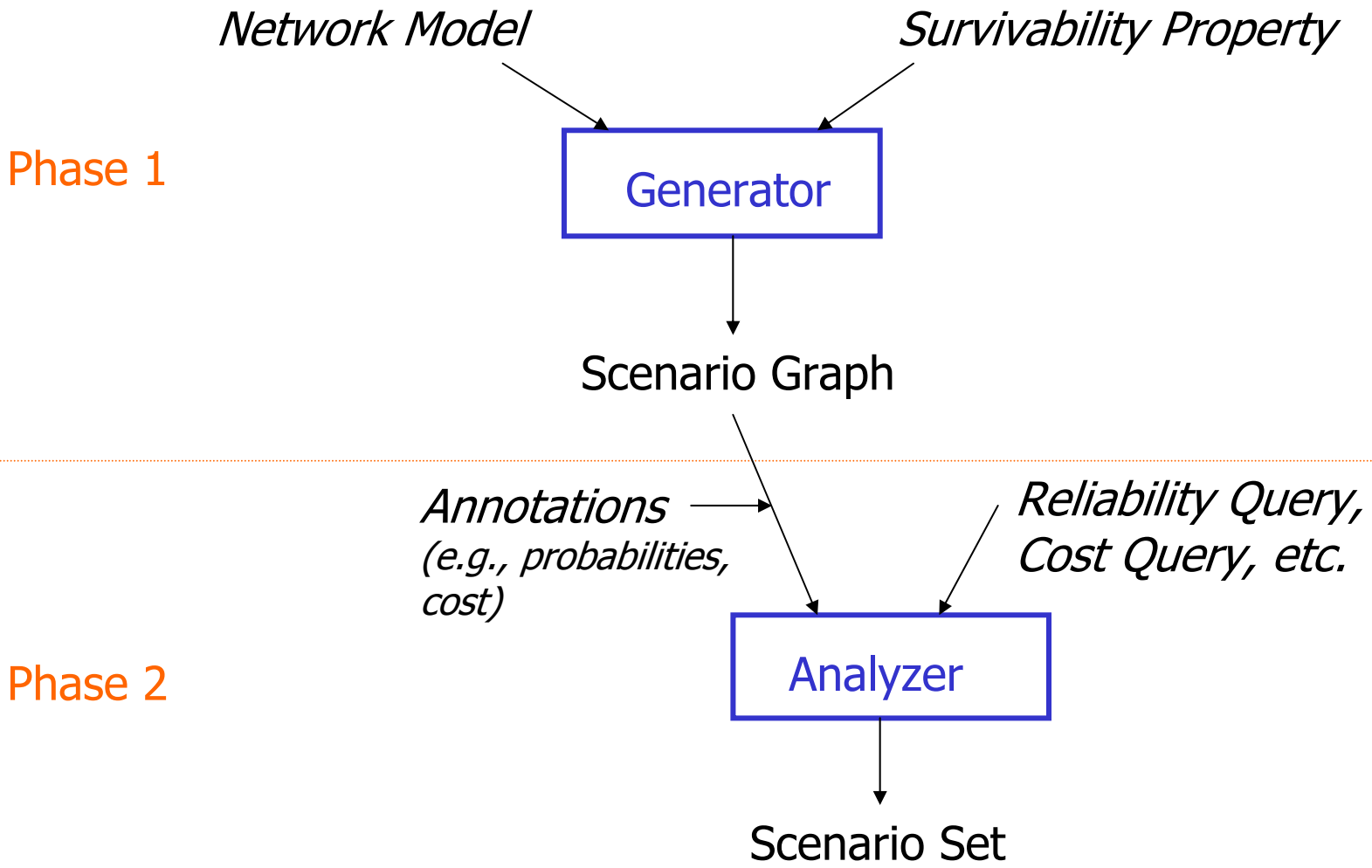
A *Service Success* Scenario Graph



A *Service Fail* Scenario Graph



Overview of Method



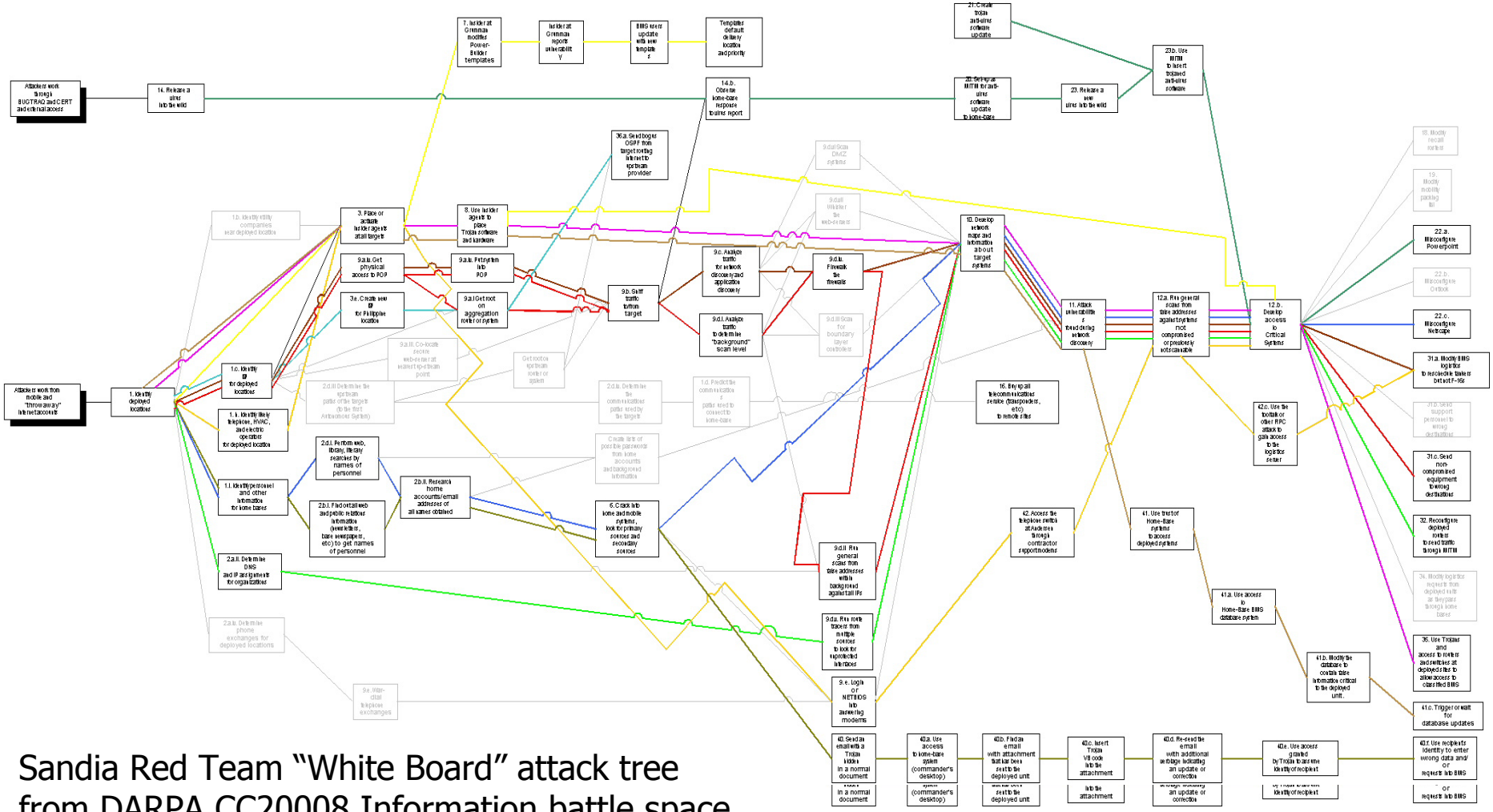
Phase 2: Reliability Analysis (in a Nutshell)

- Annotations = Probabilities
  - Use Bayesian Networks to model dependence of events.
- Symbolic
  - Use symbolic probabilities
    - high, medium, low
  - Use NDFA theory to compute scenario set.
- Continuous
  - Use numeric probabilities
    - [0.0, 1.0]
  - Use Markov Decision Processes to model both nondeterministic and probabilistic transitions.

Intrusion Detection System Case Study

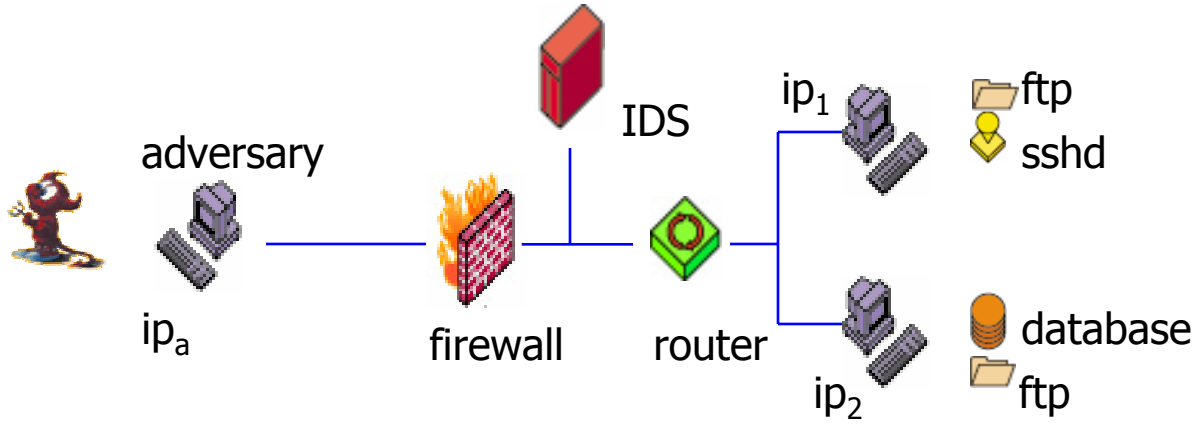
- Done by Oleg Sheyner in consultation with Lincoln Labs.
- Motivated by hand drawn poster of attack scenarios.
- So far, only a simplistic analysis for second part of method.

Example of Attack Tree Developed by a Professional Red Team



Sandia Red Team "White Board" attack tree from DARPA CC20008 Information battle space preparation experiment

Multistage Network Penetration



Goal: Gain root access to host ip_2.

Attack Arsenal	Always Detected
0 sshd buffer overflow: remotely get root	✗
1 ftp .rhosts file: establish trust between hosts	✗
2 remote login: exploit trust between hosts	✓
3 local buffer overflow: locally get root	✗

Scenario-Generating Properties

- Don’t care about detection
  - AG (adversary.privilege[2] < root)
- Want stealth
  - AG ((adversary.privilege[2] < root) or (IDS.detected))

NuSMV Encoding

- Network
  - 1 attack host, 2 target hosts with services
  - 3x3 connectivity matrix
    - ♦ existence of routing path
    - ♦ ability to connect to ftp and ssh services
  - 3x3 trust matrix
- Adversary
  - Privilege levels for each host
- Attacks
  - 4 attacks
  - some have multiple flavors

- NuSMV Statistics

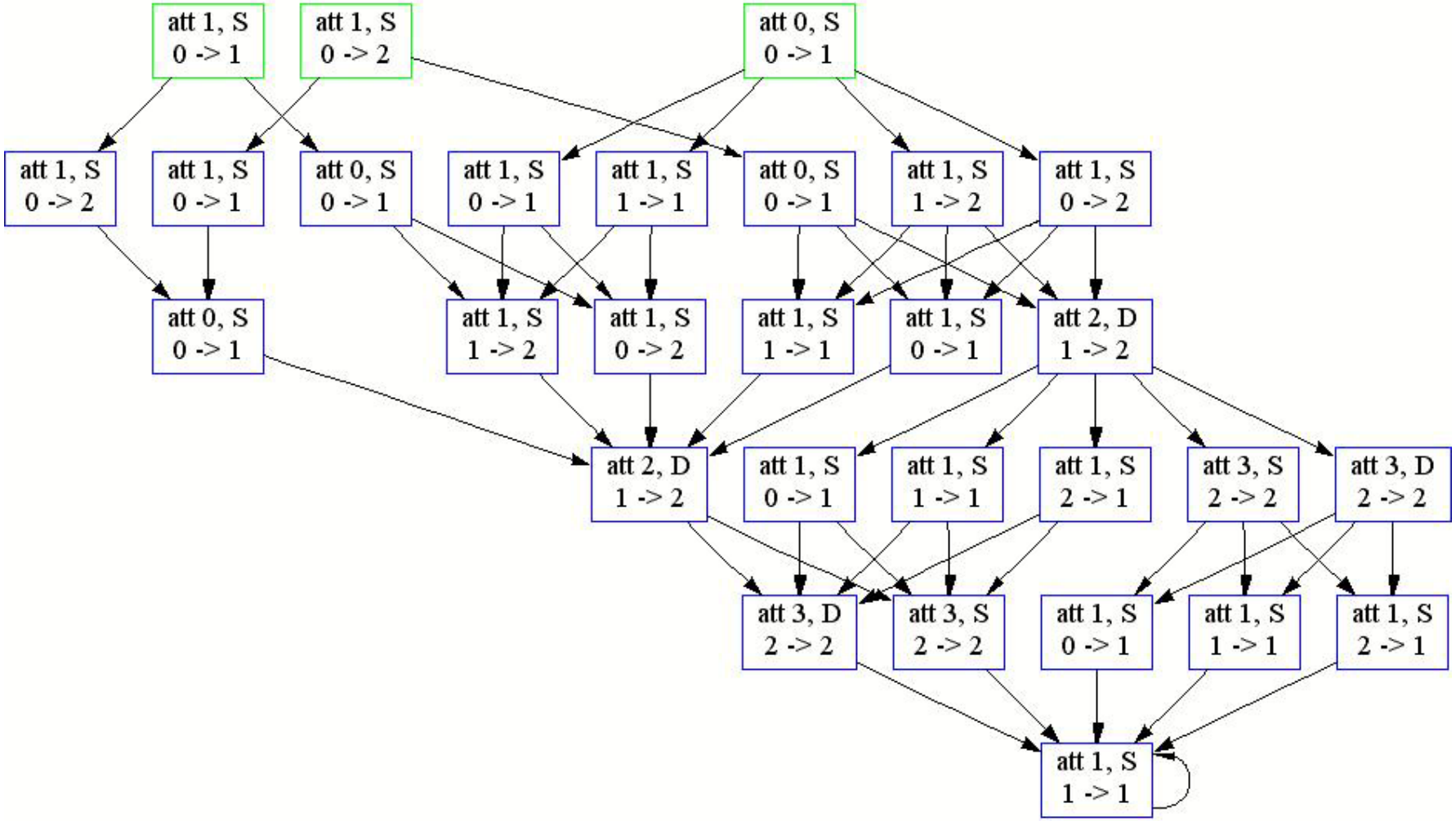
  82 bits of state (2^{82} states)

  < 40K representation nodes

  ~7000 reachable states

- 2 sec runtime on 1GHz Pentium III
- 8MB of memory used

Goal: Get Root, Avoiding Detection



Reliability Analysis Under Uncertainty

Each attack "exists" with probability P_e

- Add a boolean constant for each attack to the model indicating whether the attack exists
- Splits scenario graph into a "forest" of graphs.

Scenario Graph: Adding Uncertainty

Green – Initial States
Blue – Attacker Undetected
Red – Attacker Detected

No sshd attack

All attacks

Questions:

What is the probability that the attacker will succeed?
What is the probability that the attacker will be detected?

Scalability

- Expanded case study
  - ✓ 5 hosts
  - ✓ 4 new attacks
  - ✓ legitimate users
  - ✓ background traffic
    - ❖ high priority
    - ❖ low priority
  - ✓ multiple firewall configurations

- NuSMV runtime: 4.5 hours
- ~ 6000 nodes in scenario graph

Two Other Case Studies (by Somesh Jha)

- Trading floor model of major investment bank (being "sanitized")
  - 10K lines of NuSMV
  - half-million nodes in scenario graph
  - 50 threat scenarios
  - 45 found by system
  - 5 new threat scenarios found
  - With independence assumption, too many misses.
- B2B e-commerce NYC start-up
  - 50K lines of Statecharts
  - 2 million lines of NuSMV beyond capability of tool

Open Research Questions

- Understanding Survivability
  - What is an appropriate logic for describing survivability properties?
  - How can you design a system for survivability?

- Analysis Technique
  - Scalability: What new data structures, abstraction techniques, compositional reasoning will let us handle larger state spaces?
  - Tools: What combination of tools can further automate the analysis?
    - Linear programming packages, theorem provers, …
  - Applicability: How applicable is the CMDP model for other application domains?
    - Can they be applied to embedded and autonomous systems?