# Partitioning Network Testbed Experiments

Wei-Min Yao, Sonia Fahmy

Purdue University

June 21, 2011

This work is funded in part by Northrop Grumman Information Systems, and by NSF grant CNS-0831353.

# Where to run our large-scale experiments?

- Where can researchers and operators conduct large-scale network experiments?
  - Building models for large systems in the current Internet is challenging.
  - Unlikely to conduct experiments on the production network.
- The necessity of large-scale and high fidelity experimental environment.
  - Many large-scale attacks have second-order effects, e.g., worm or DoS causes excessive ARP traffic or BGP session resets

- **Today's testbeds have 100~1000 nodes but we need to conduct accurate Internet-scale experiments**

# Experimentation Methods

| Method | Scalability | Fidelity | Configuration | Limitations |
|---|---|---|---|---|
| Simulation | Medium - Large | Problematic | Easy for existing models | How to test new protocols or boxes? |
| Emulation | Small - Large | Good, but emulated parts can be problematic | Requires expertise | Expensive to build and maintain. |
| Global-scale testbeds | Medium+ | Higher | Requires expertise | Results are not reproducible.<br>Not an isolated testbed. |
| Small testbeds | Small | Higher | Easy for the target scenario, but hard to modify | Full implementation can be expensive. |

# The Question of Scale…

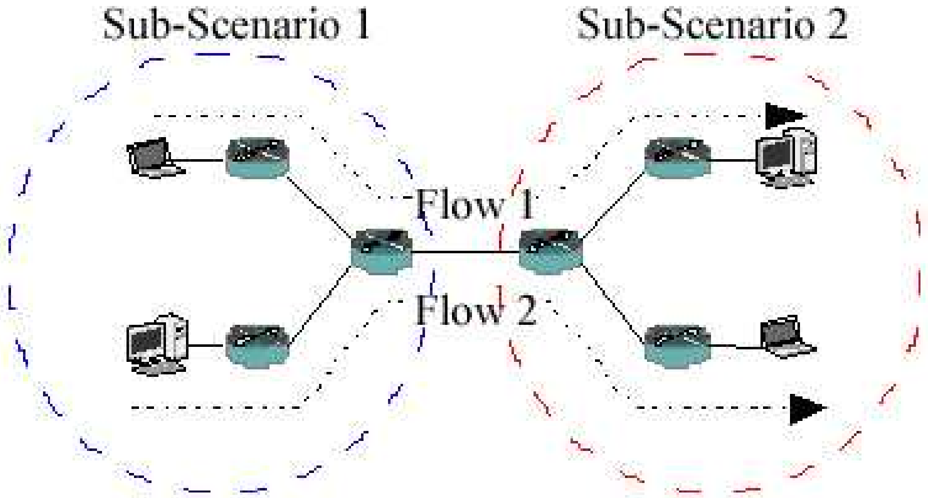
- Approaches to scale network experiments
  - Network Simulation
    - Parallelization
      - □ PDNS [Riley *et al*, TOMACS'04], SSF [Ogielski *et al*, www.ssfnet.org]
    - Reduce simulation events
      - □ SHRiNK [Pan *et al*, TON'05], TranSim [Kim *et al*, INFOCOM'06]
  - Network Emulation
    - Intelligent resource allocation
      - □ Virtualization on Emulab [Hibler *et al*, USENIX'08]
    - Emulation with time virtualization
      - □ DieCast [Gupta *et al*, NSDI'08], SliceTime [Weingärtner *et al*, NSDI'11]
    - Simplify input topology
      - □ Path emulation [Sanaga *et al*, NSDI'09]
- There is no complete solution yet
  - No single approach is capable of scaling to Internet-scale.

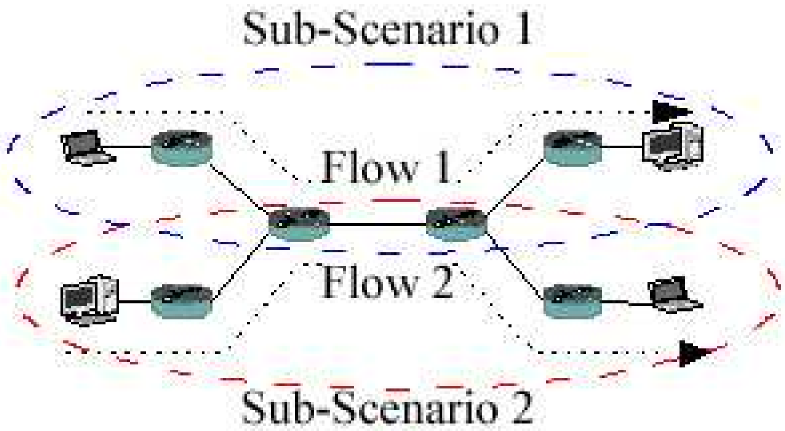
# Partitioning network experiments?

- If an experiment is too large for a testbed, can we run a sequence of experiments on a testbed?
  - Not all flows in an experiment are directly related.
    - Identify "unrelated" flows and study them *independently*.
  - Fine-grained metrics are not always required for all the flows
    - Some loss of fidelity is acceptable, especially for background flows.

# Flow-based Scenario Partitioning (FSP)

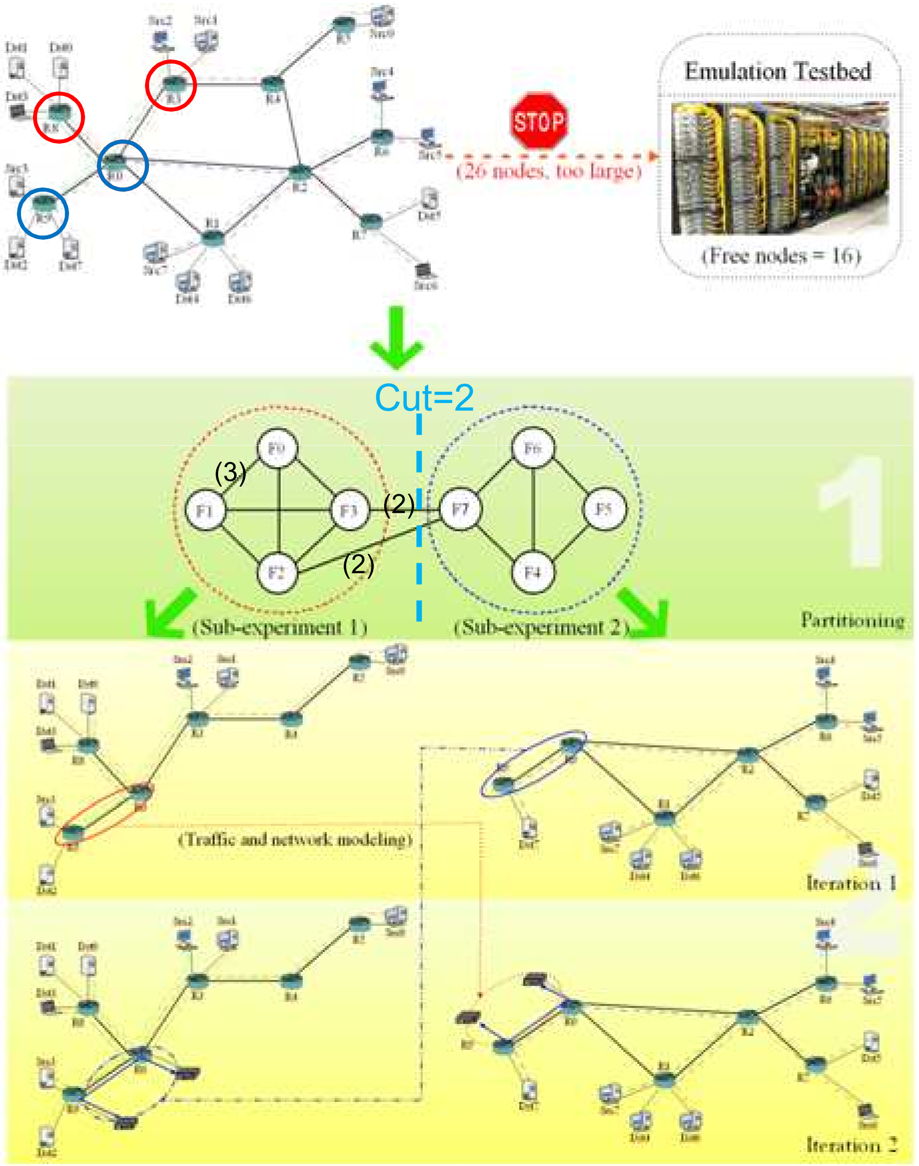
- Partition network scenario into sub-scenarios based on flows
  - Partition a scenario (topology, flows) into *sub-scenarios*, given a constraint *(maxNodes)* on the number of machines in the testbed





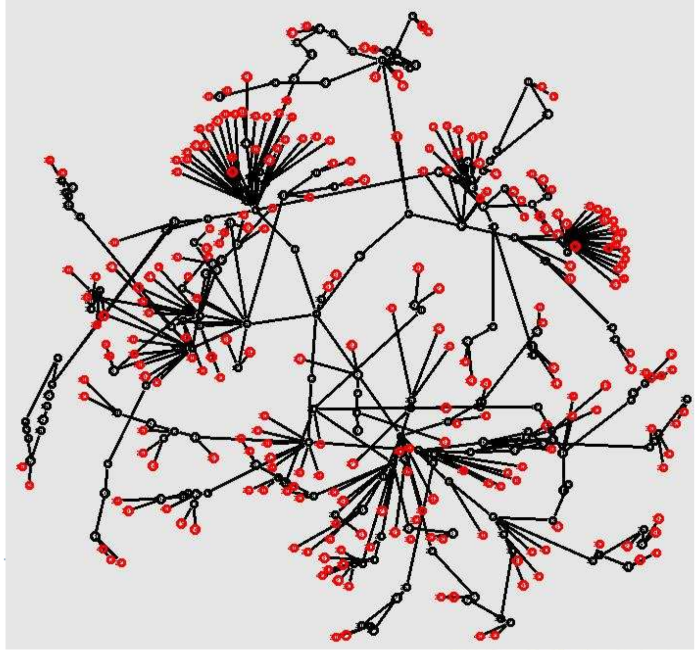
# Overview of FSP

- Phase 1:
  - Construct a Flow Dependency Graph (FDG)
- Phase 2:
  - Conduct sub-scenario experiments independently and iteratively
  - Collect traces for dependent flows, if any
  - Extract from these traces: application traffic models and network conditions on non-shared links
  - Conduct experiments



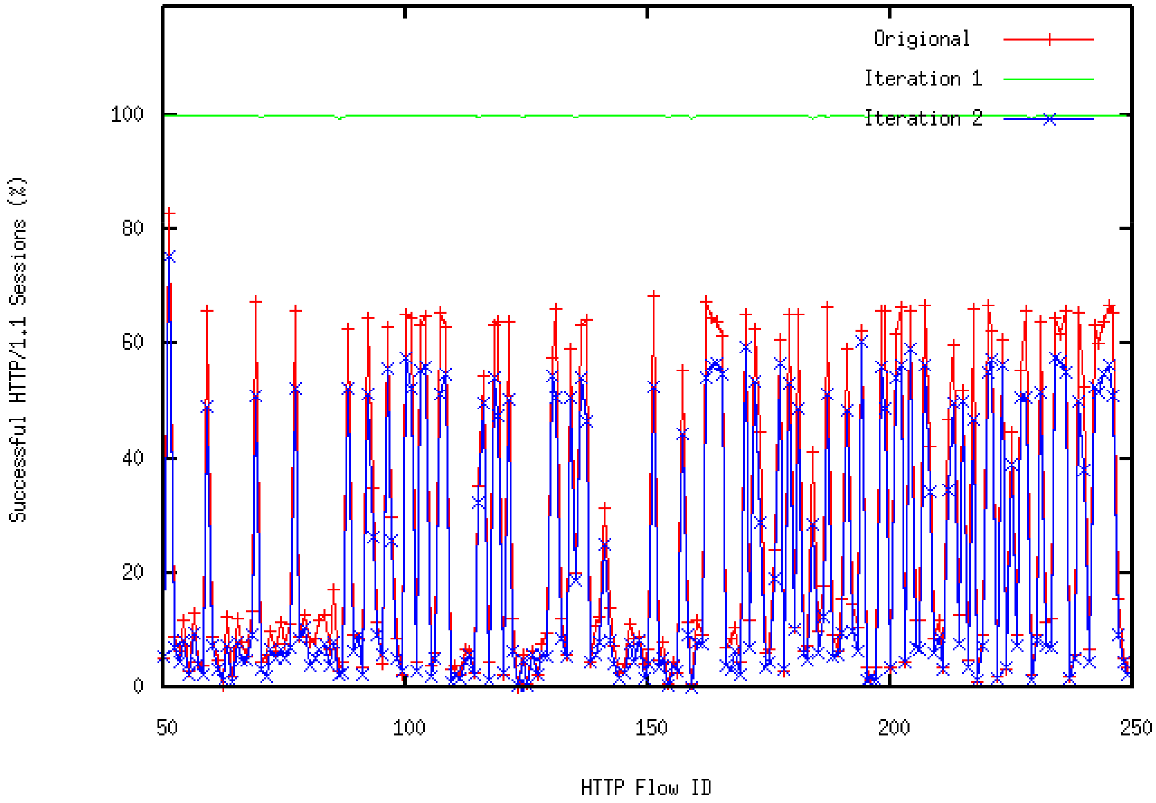
# Botnet Experiment

- Experiment Setup
  - Legitimate users from 200 subnets (/24)
    - Requests generated according to logs in a production web server
    - Cover 70% of the service providers of all visitors in 2009
  - Attackers from 50 subnets (/24)
    - Selected from public black list (Dshield.org)
    - UDP flood attack
  - Use traceroute to generate the topology
    - 438 nodes required (1232 routers initially)



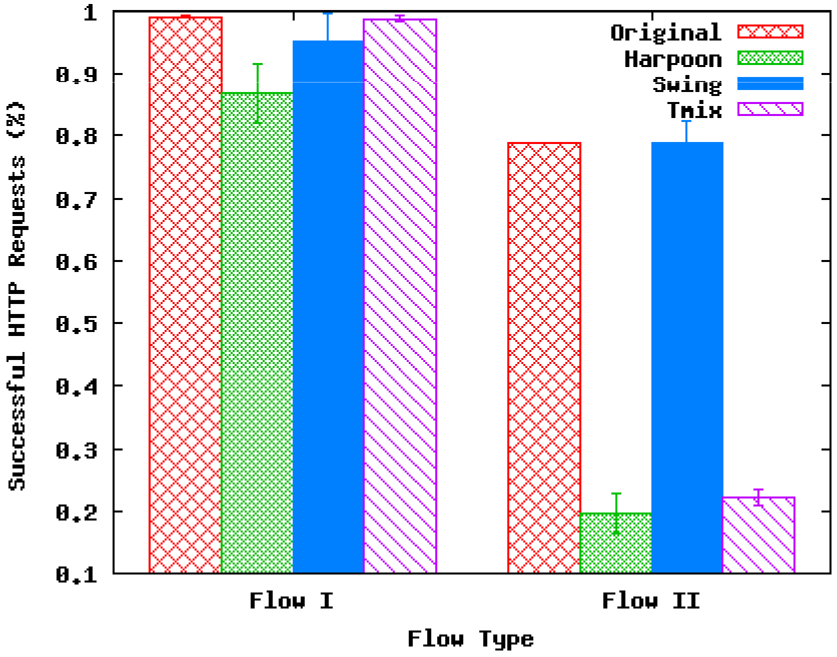
# Botnet Experiment

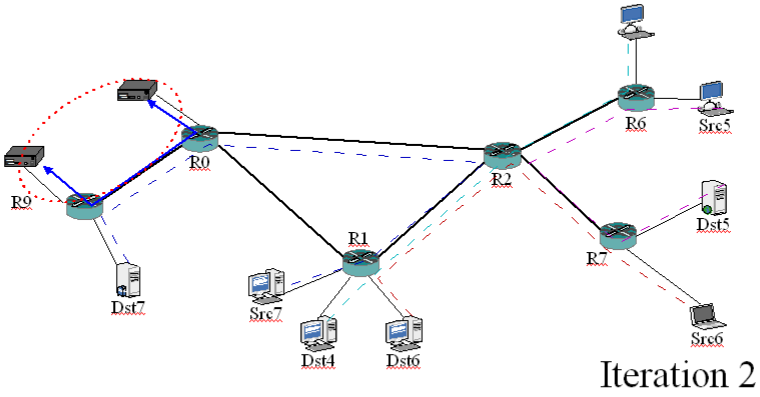
- Results from the 2nd iteration can be used to predict the original scenario (the success rate of HTTP 1.1 flows)



# Traffic modeling tools

- Tradeoff between fidelity and resources
  - 0.8 M Packets (12 K connections)
    - Trace Size: 74 MB (pcap) vs 1.2 Mb (NetFlow)
    - Processing Time: 1 sec / 40 sec / 35 sec  (Harpoon/Swing/Tmix)





# Conclusions

- FSP is a platform-independent mechanism to partition a large network experiment into smaller experiments.
  - Smaller experiments can be executed sequentially on a limited number of testbed machines.
  - No modification required on the testbed.
  - Can be integrated with existing virtualization and parallelization techniques.
  - Provides good prediction of coarse-grained metrics.
- We validated FSP in ns-2 and DETER testbed experiments
  - Evaluate the selection of weights when partitioning a FDG
  - Comparison between FSP and the TranSim downscaling technique
  - Comparison among different modeling tools in phase 2 (Tmix, Harpoon and Swing)

# Thank You