HotNets 2015

How to destroy networks for fun (and profit)

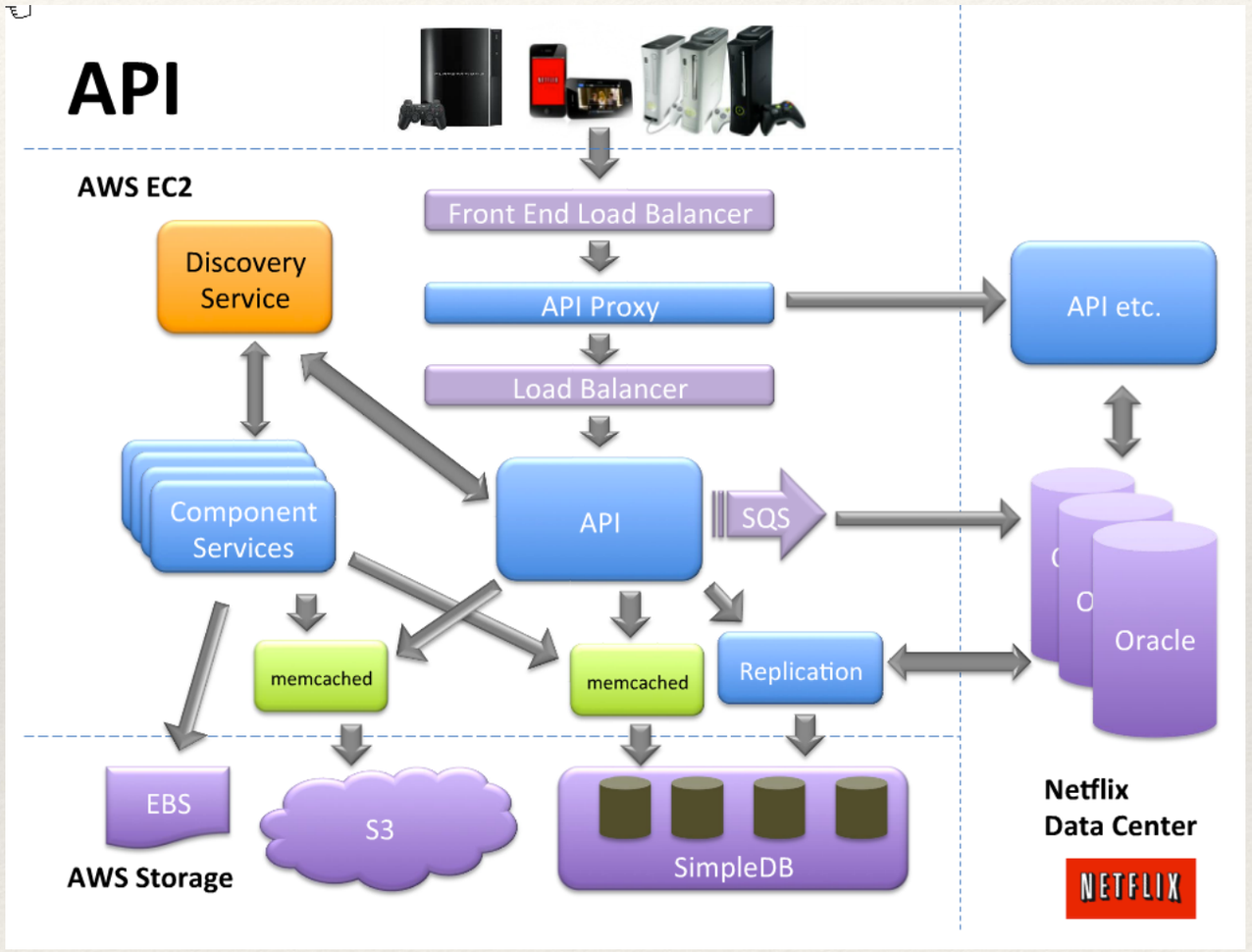
Nick Shelly

ETH zürich

Joint work with:
Brendan Tschaen, Klaus-Tycho Förster, Michael Chang,
Theophilus Benson, Laurent Vanbever

The network has become more complex

- Web services are growing in complexity and across location
- Millions of lines of code in devices
- Need to test systemic failure
- NY Stock Exchange, United, and WSJ.com all went down in one day (Jul 2015) due to technical glitches (“single router issue”)



Outline

- Netflix's Approach and black-box testing
- Design of Armageddon
- Results on real-world topologies
- Coverage scenarios and failure types

Netflix's Approach

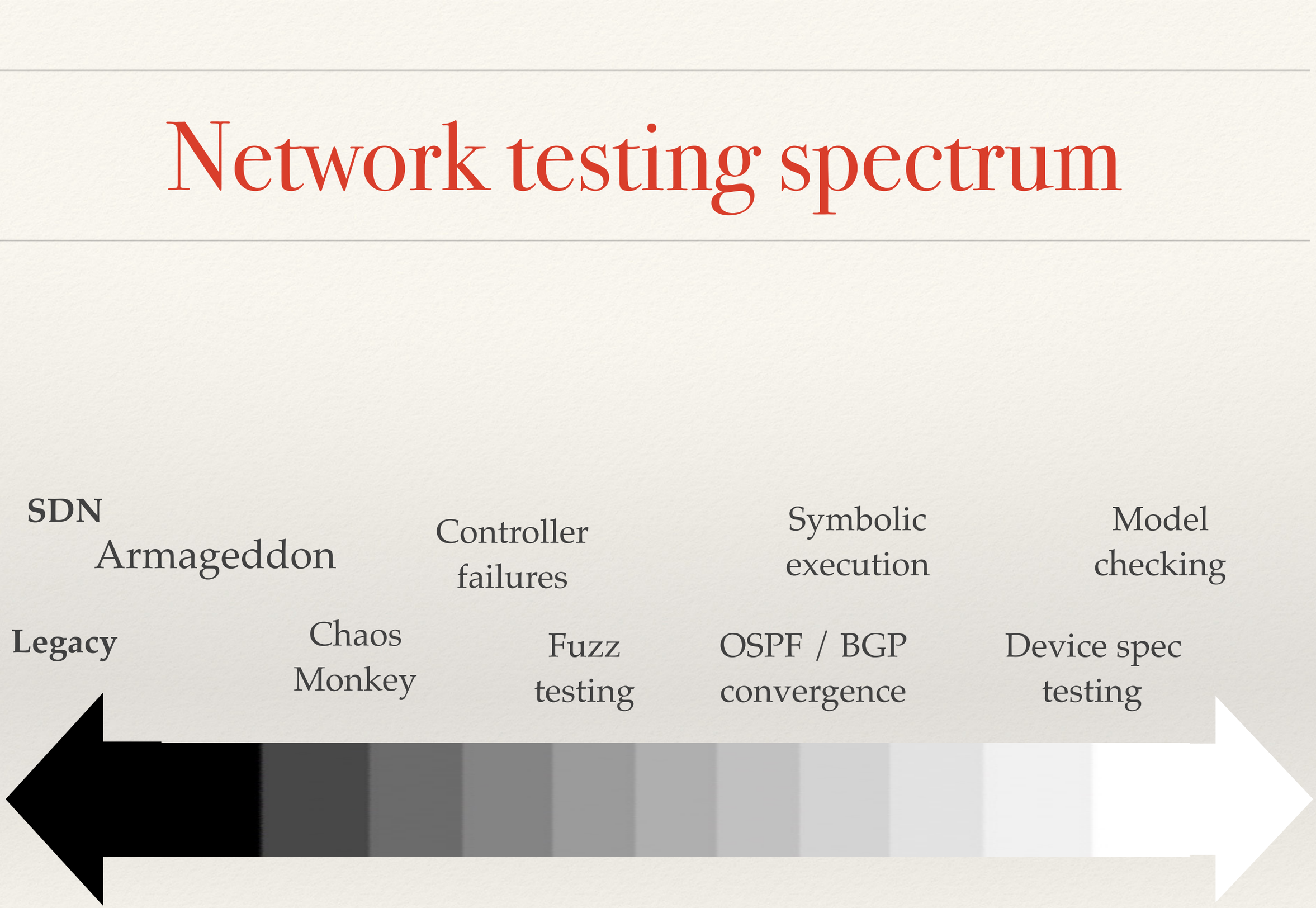
"The best way to avoid failure is to fail constantly."

Problem	Test
Disable production	Chaos Monkey
Client-server interaction	Latency Monkey
Company "best practices"	Conformity Monkey
Health of systems	Doctor Monkey
Cloud garbage collection	Janitor Monkey
AWS security, SSL certs	Security Monkey
Internationalization	10-18 Monkey
Disable entire zone	Chaos Gorilla

Random vs. pre-meditated chaos

- More overall coverage
- Planned failures should have a minimum duration (say 1 hour of outage during work hours), so cover as much as possible in as few "iterations"
- Systematic but not exhaustive
- Allow a well-programmed network to succeed

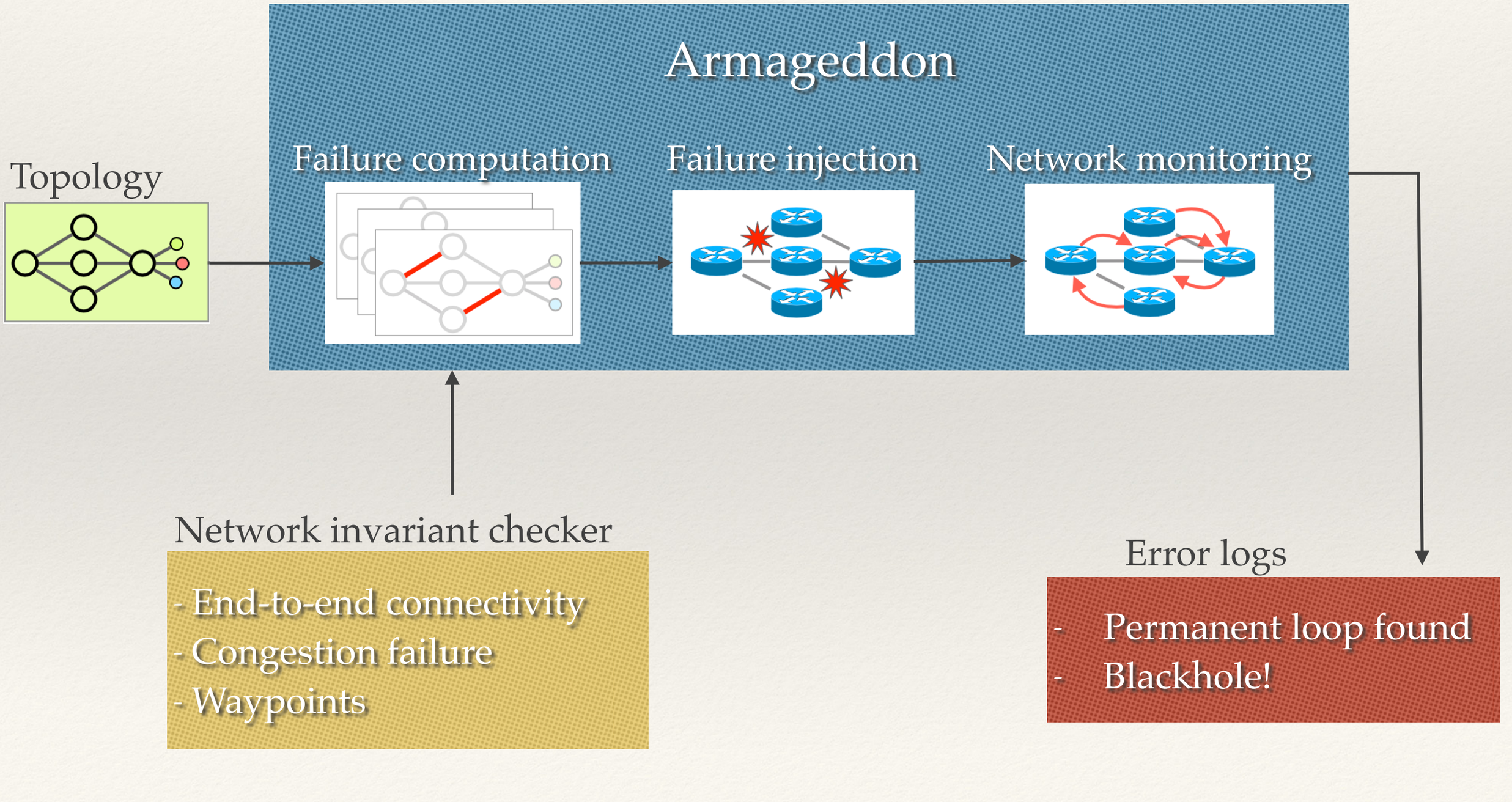
Network testing spectrum



Outline

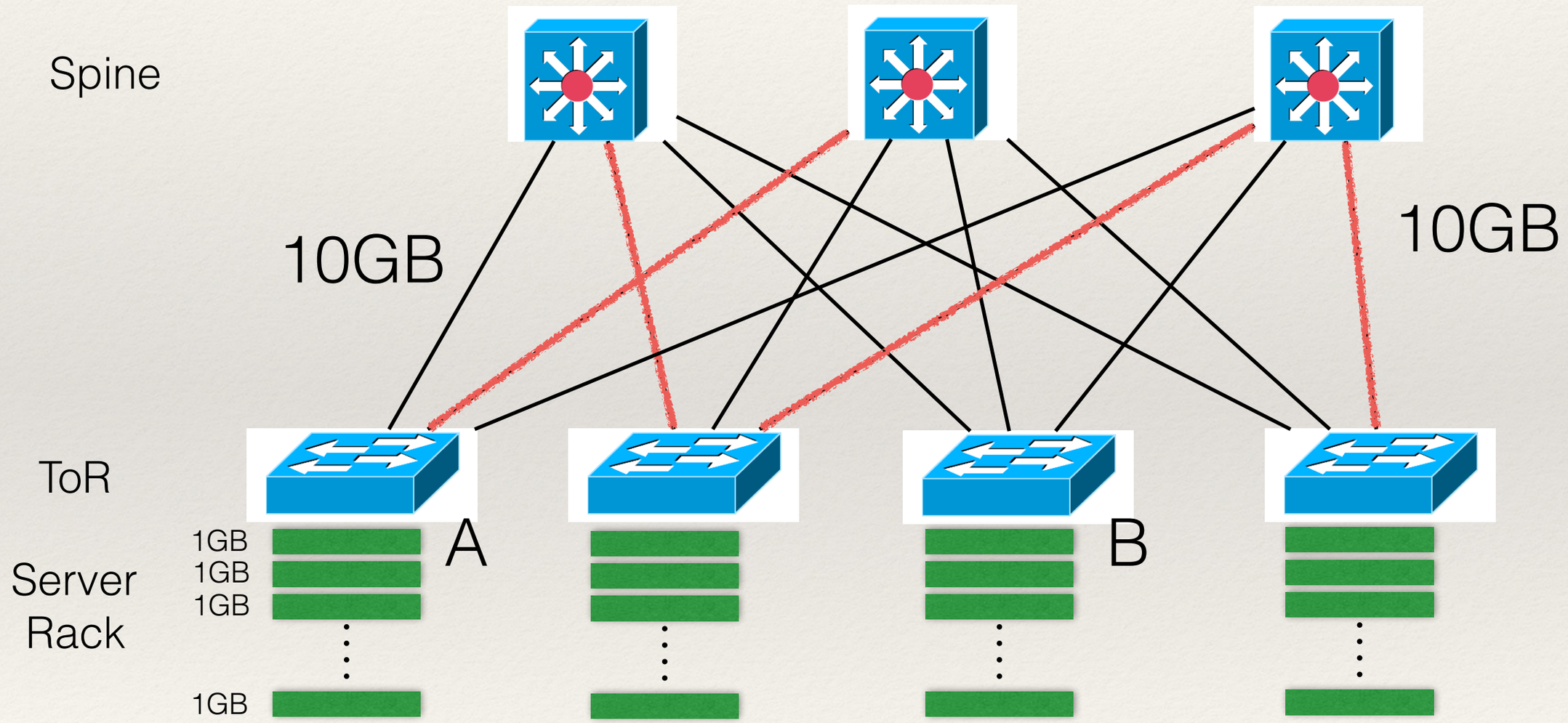
- Netflix's Approach and black-box testing
- Design of Armageddon
- Results on real-world topologies
- Coverage scenarios and failure types

From topology to failures, while checking



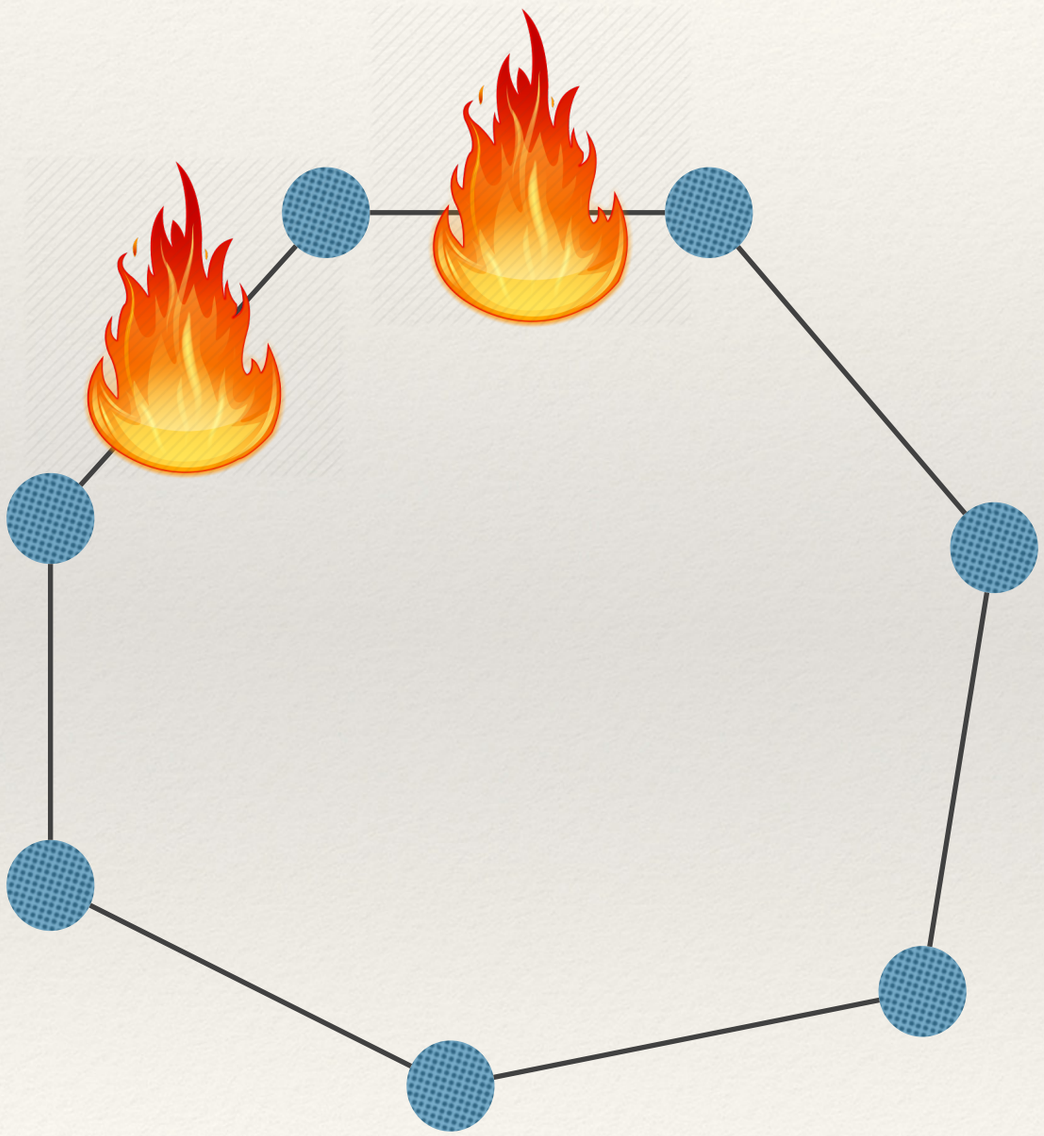
How to disrupt the network

Which links can we fail, while making sure there is connectivity between A and B?



Initial idea - fail links while keeping connectivity

How to test all links, while making sure there is always connectivity between every two nodes?



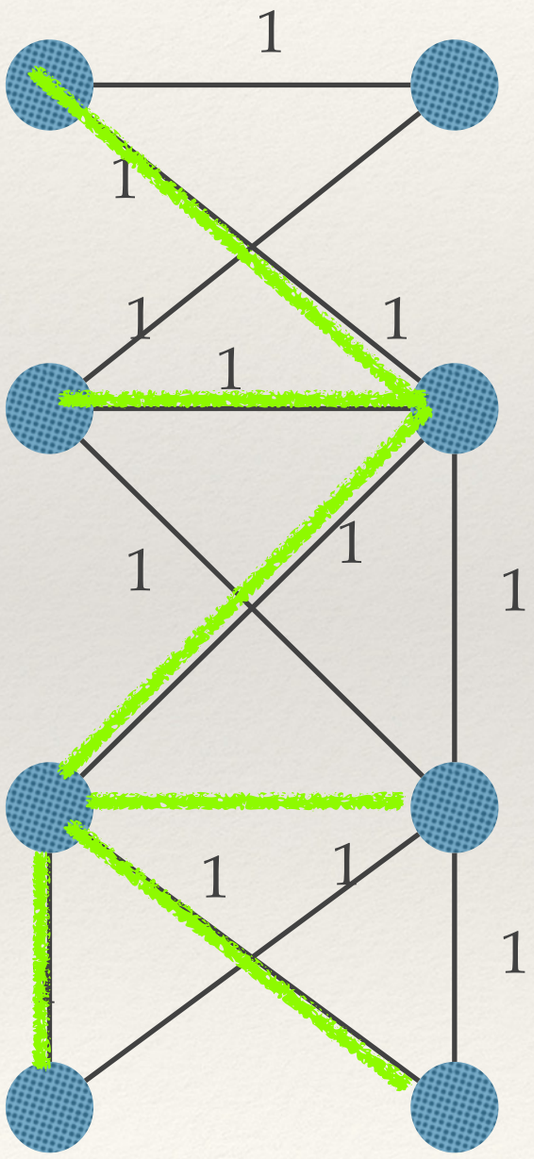
Fail only one link per iteration

For n nodes, takes n iterations to fail each node in network

Ring network is upper bound on # of iterations

General case - graph "Jenga"

How to test all links, while making sure there is always connectivity between every two nodes?

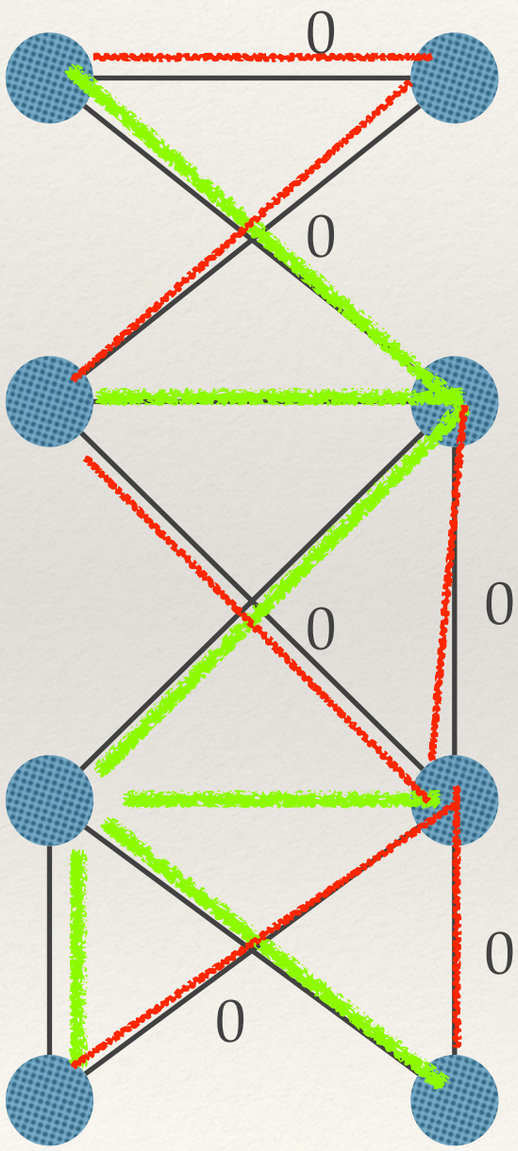


Greedy Killer Algorithm:

1. Set all links to *weight=1*
2. Find Minimum Spanning Tree — this will be left over after failing everything else

General case - graph "Jenga"

How to test all links, while making sure there is always connectivity between every two nodes?



Greedy Killer Algorithm:

1. Set all links to *weight=1*
2. Find Minimum Spanning Tree — this will be left over after failing everything else
3. Terminate all links not in MST
4. Mark those terminated as *weight=0*

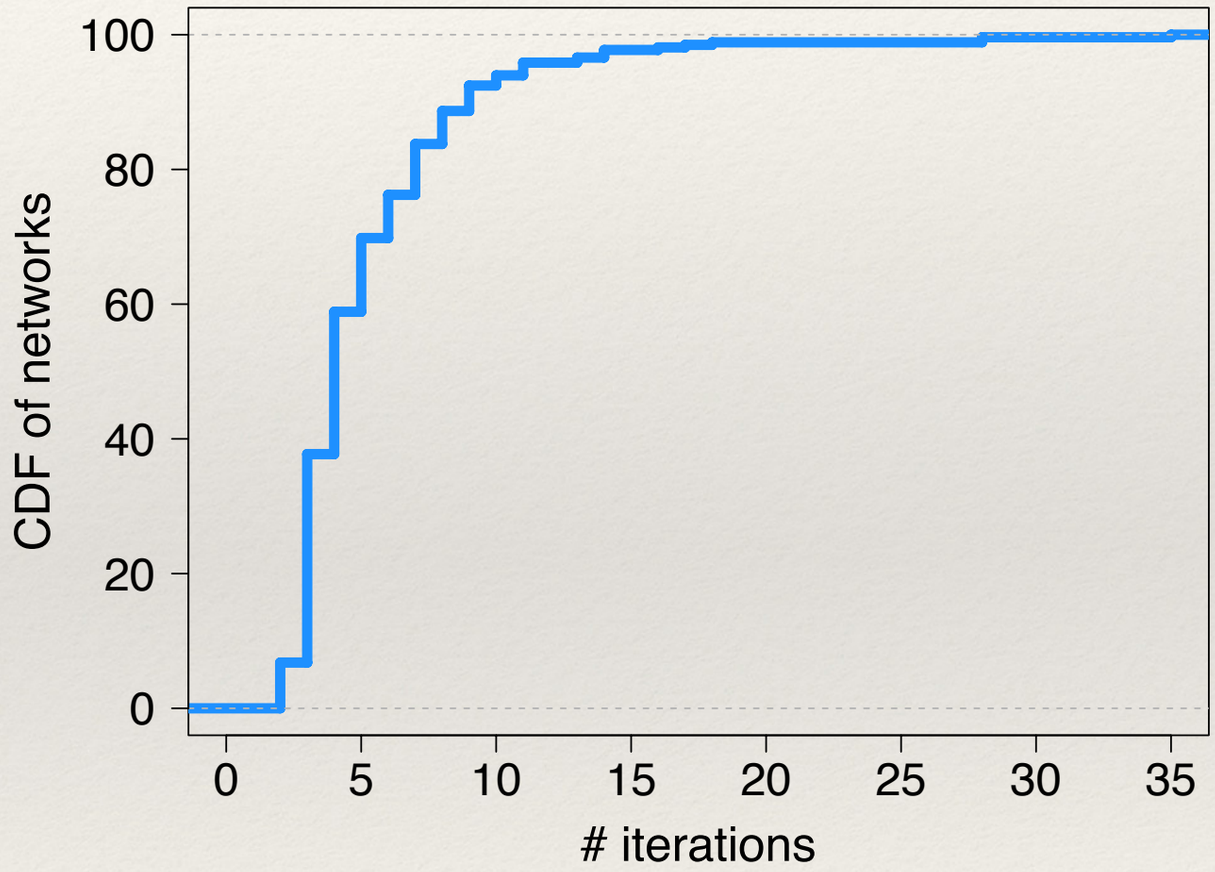
Repeat Steps 2 - 4, until all links are tested at least once.

Outline

- Netflix's Approach and black-box testing
- Design of Armageddon
- Results on real-world topologies
- Coverage scenarios and failure types

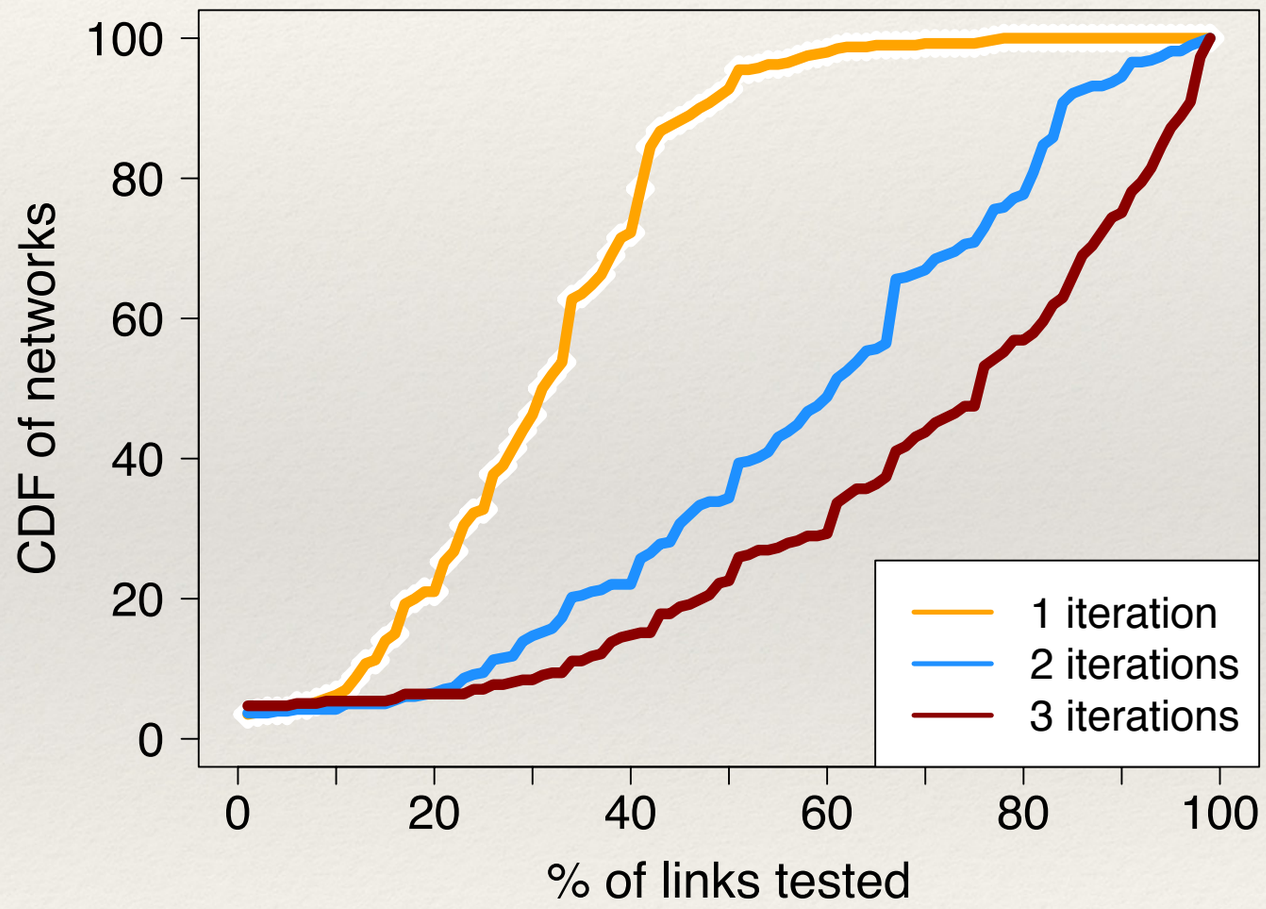
Iterations to test network completely

- 261 network topologies from Internet Topology Zoo and 7 RocketFuel graphs on ISPs
- Some links cannot be failed: remove “un-failable” links and treat as sub-networks
- Ring networks take a long time to fail
- 78% of the networks can be failed entirely in 6 iterations, 91% in 8 iterations



Can "stress test" most of network quickly

- In one iteration we can fail about 30% of the links
- In half the networks we can test 80% of the links in 3 iterations
- Optimal > 50% of the time



Outline

- Netflix's Approach and black-box testing
- Design of Armageddon
- Results on real-world topologies
- Coverage scenarios and failure types

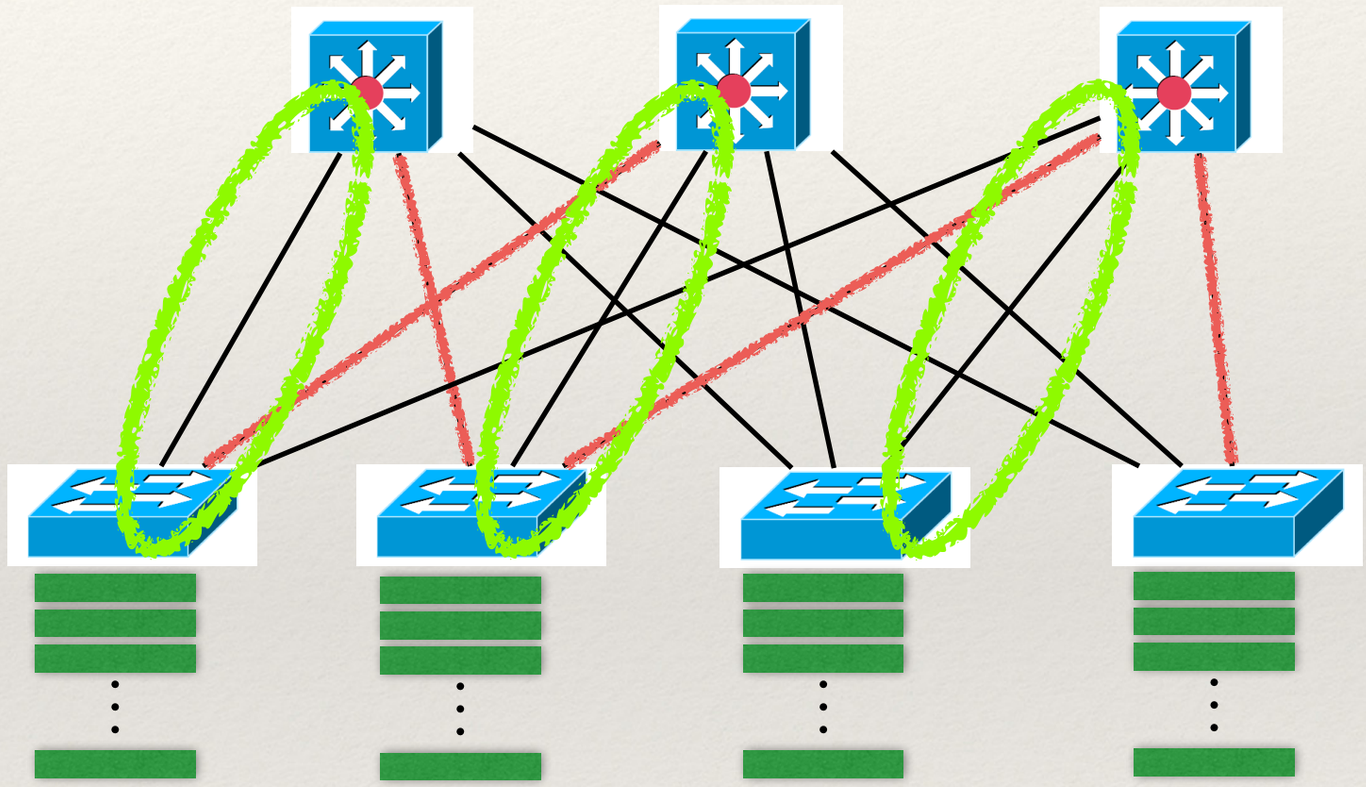
Future work - better failure scenarios

We currently only fail to force connectivity over single links (no more than *n* failure iterations)

Can we test multiple links at a time (~m^2 scenarios test all sets of 2 link failures with *m* links)

Generalize the network via clustering (isomorphism)

Bandwidth guarantees on failures



Future work - different failure types

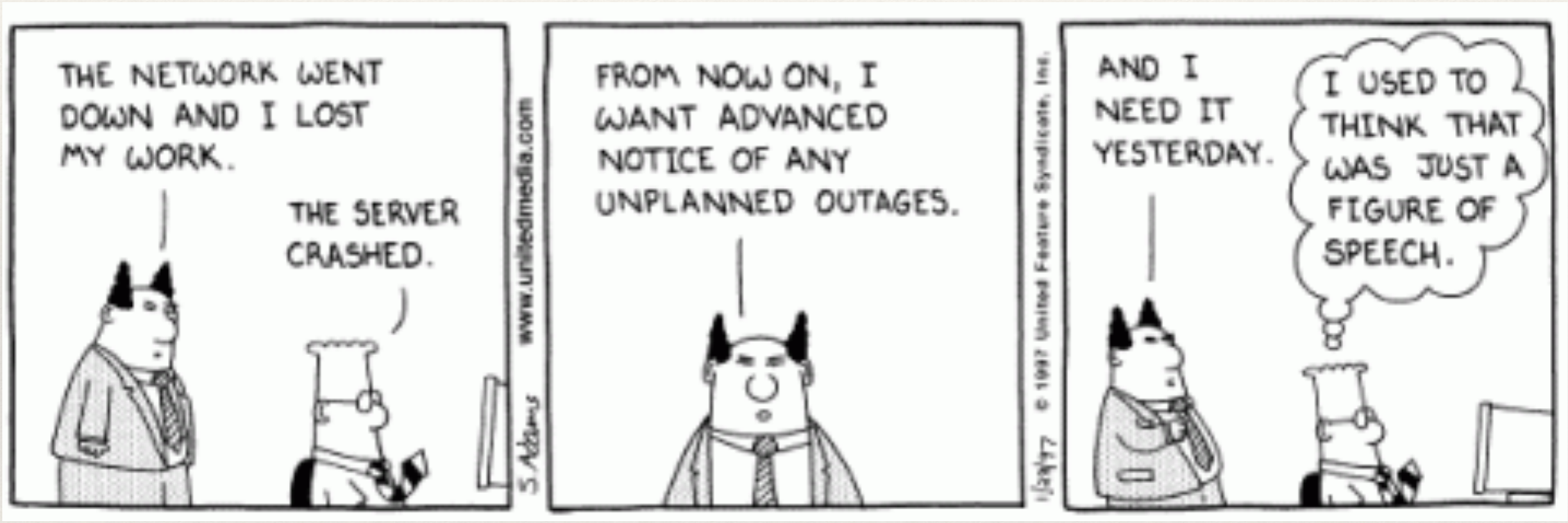
Control Plane:

- Failing erroneous replica
- Send random drop or erroneous command
- Force usage of waypoint

Data-plane:

- Fail device or ports
- Add link delays or drops
- Resource exhaustion
- Traffic congestion (over use resource)

Questions?


THE NETWORK WENT DOWN AND I LOST MY WORK.
THE SERVER CRASHED.
FROM NOW ON, I WANT ADVANCED NOTICE OF ANY UNPLANNED OUTAGES.
AND I NEED IT YESTERDAY.
I USED TO THINK THAT WAS JUST A FIGURE OF SPEECH.
www.unitedmedia.com
S. Adams
United Feature Syndicate, Inc.