CS 5450

Networked and Distributed Systems

Vitaly Shmatikov

Motivation

- The Internet
  - Every system you use or build does and will depend on it – good to know how it works
  - Lessons for scalability, longevity, robustness
- Distributed systems
  - Key techniques for building robust, scalable, performant systems on top of Internet infrastructure

Course Logistics

- Lectures: Monday & Wednesday, 5:00-6:15pm
- Course webpage:

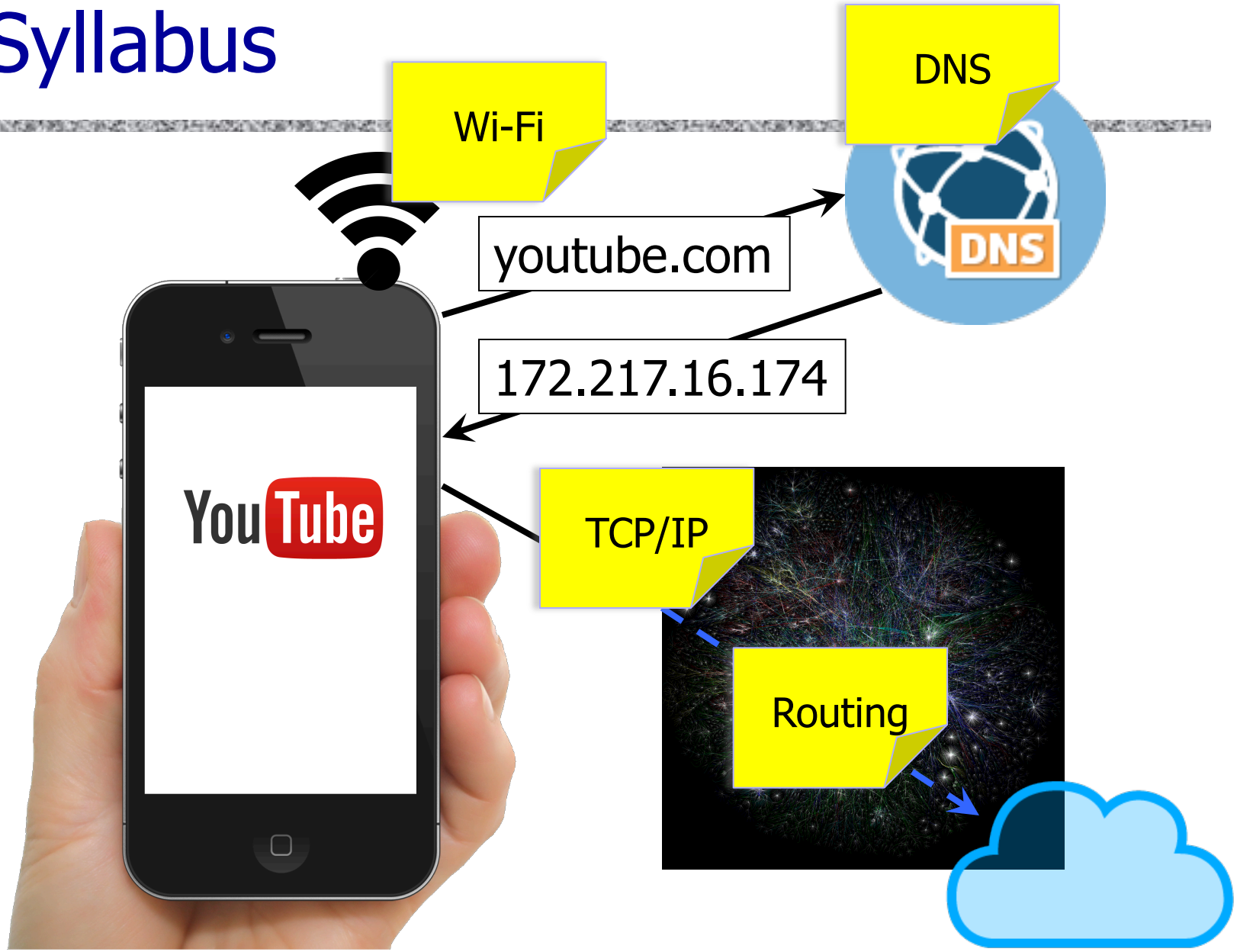
https://cs5450.github.io/

- TAs: Jialing Pei (full)
  Eugene Bagdasaryan (half)

Grading

- Weekly class attendance/participation (solo): 10%
- Four labs (in rotating pairs): 60% (15% each)
  - Design and significant implementation
  - 2 late days to use as you see fit… once used up, no late assignments accepted (applies to both team members)
- Two in-class midterms (solo): 30% (15% each)
  - Pen-and-paper exercises, 1-page reference sheet Ok

Syllabus


Wi-Fi
DNS
DNS
youtube.com
172.217.16.174
YouTube
TCP/IP
Routing

Syllabus

CDNs

BitTorrent

P2P

Tor

Anonymity networks

Blockchain

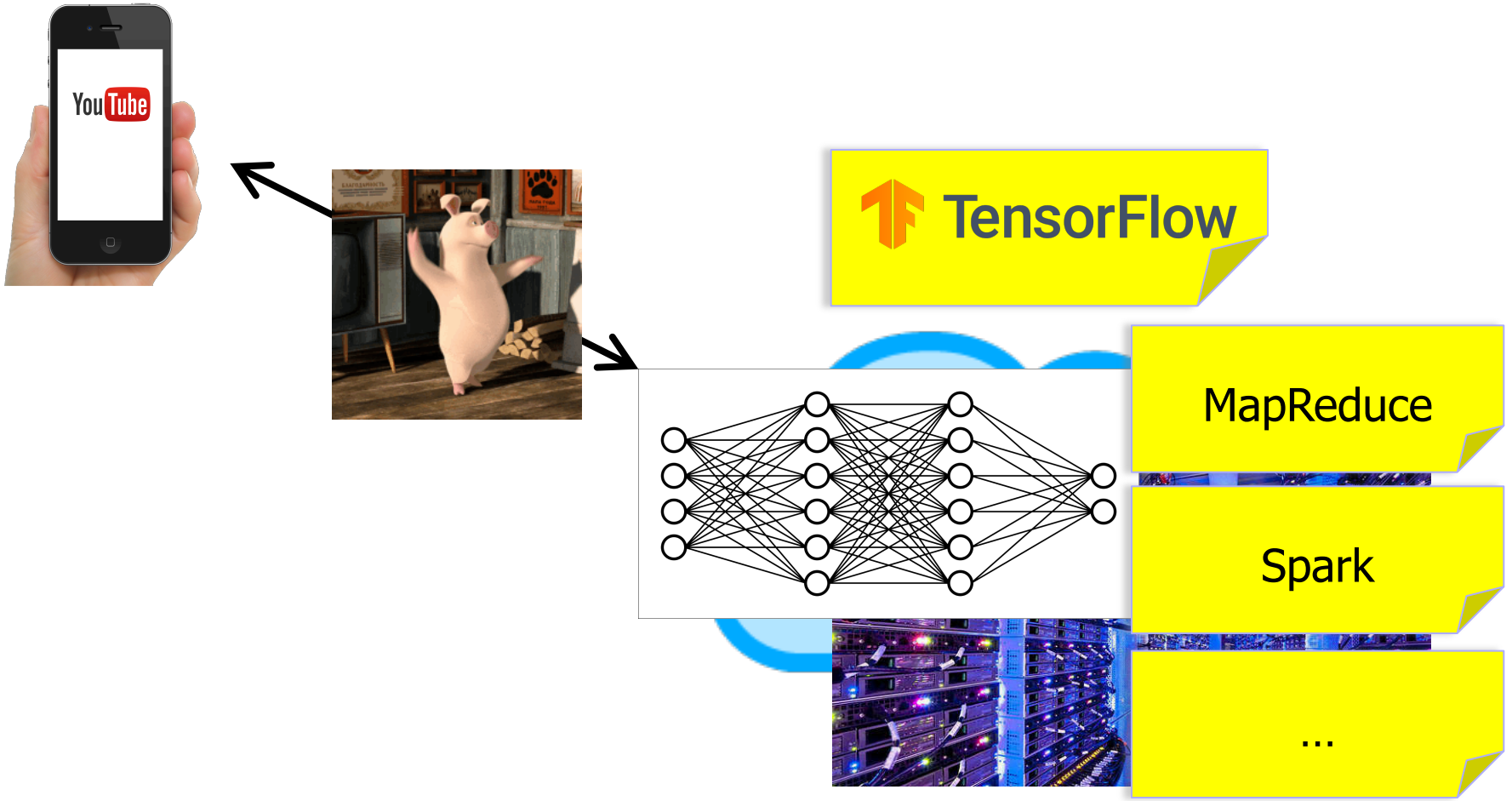
Virtualization

Dynamo

Zookeeper

...

Syllabus



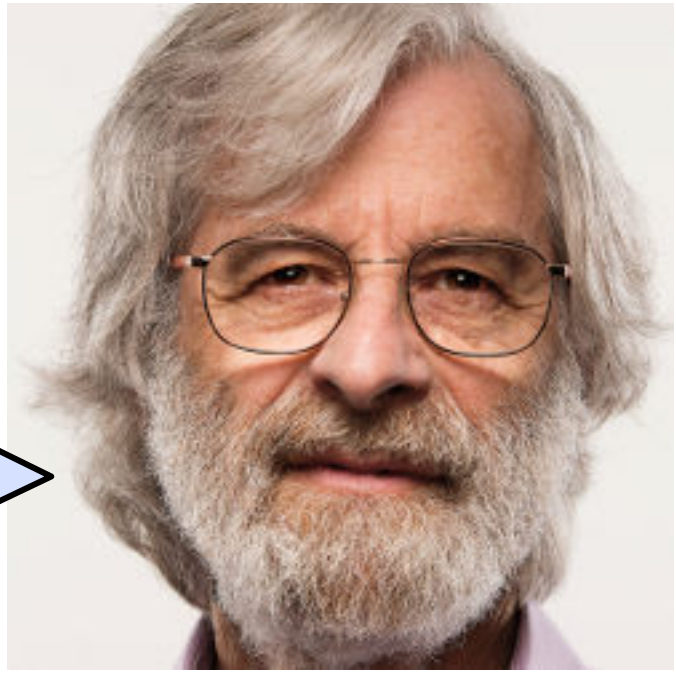
What's a Distributed System?

"a collection of independent computers that appear to the users of the system as a single computer"

"several computers doing something together. Thus, a distributed system has three primary characteristics: multiple computers, interconnections, and shared state"

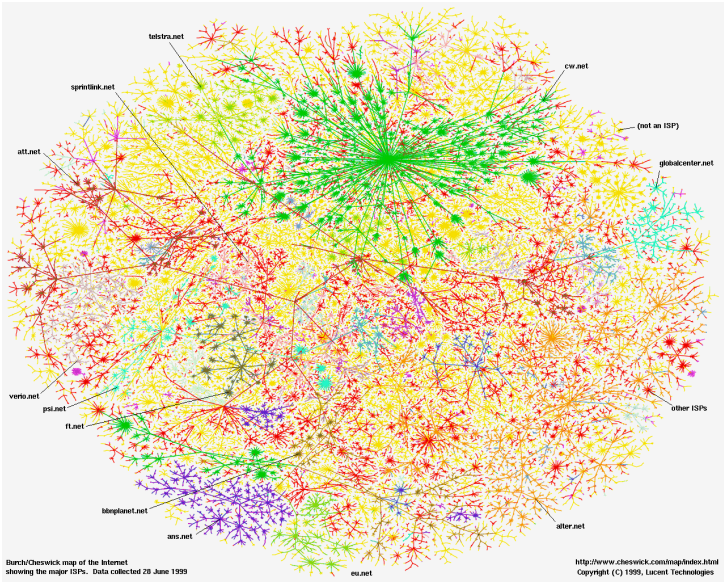
Back in 1990

"A distributed system is one where you can't get your work done because some machine you've never heard of is broken."



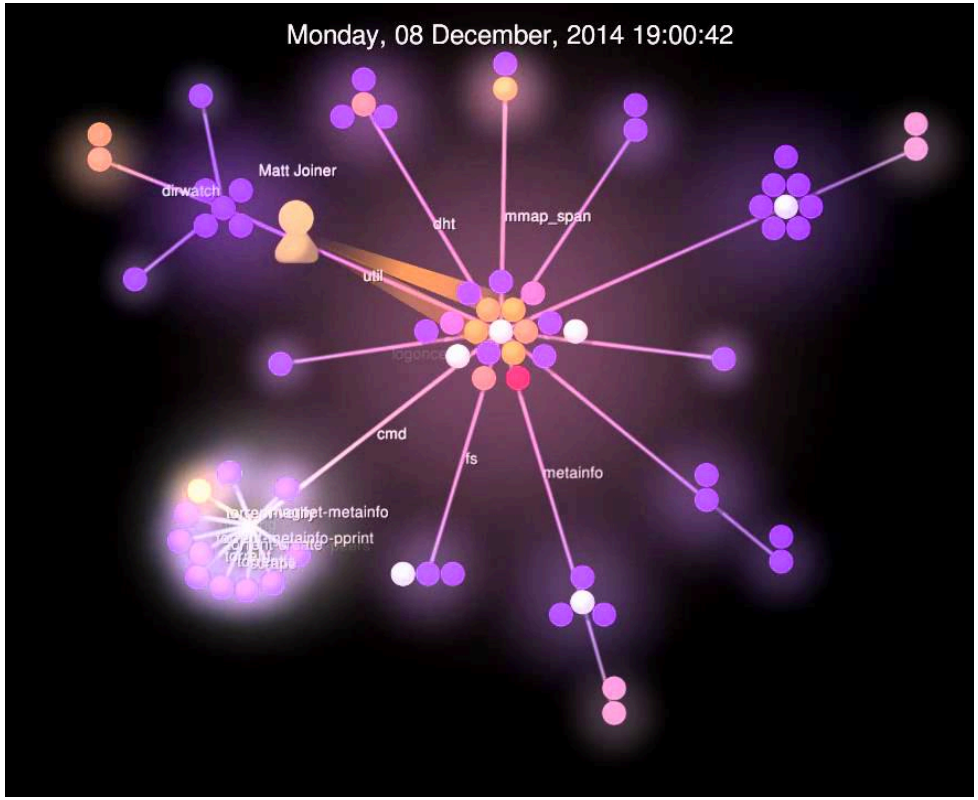
Which of These is a DS?

Facebook social network graph



Internet ISP map

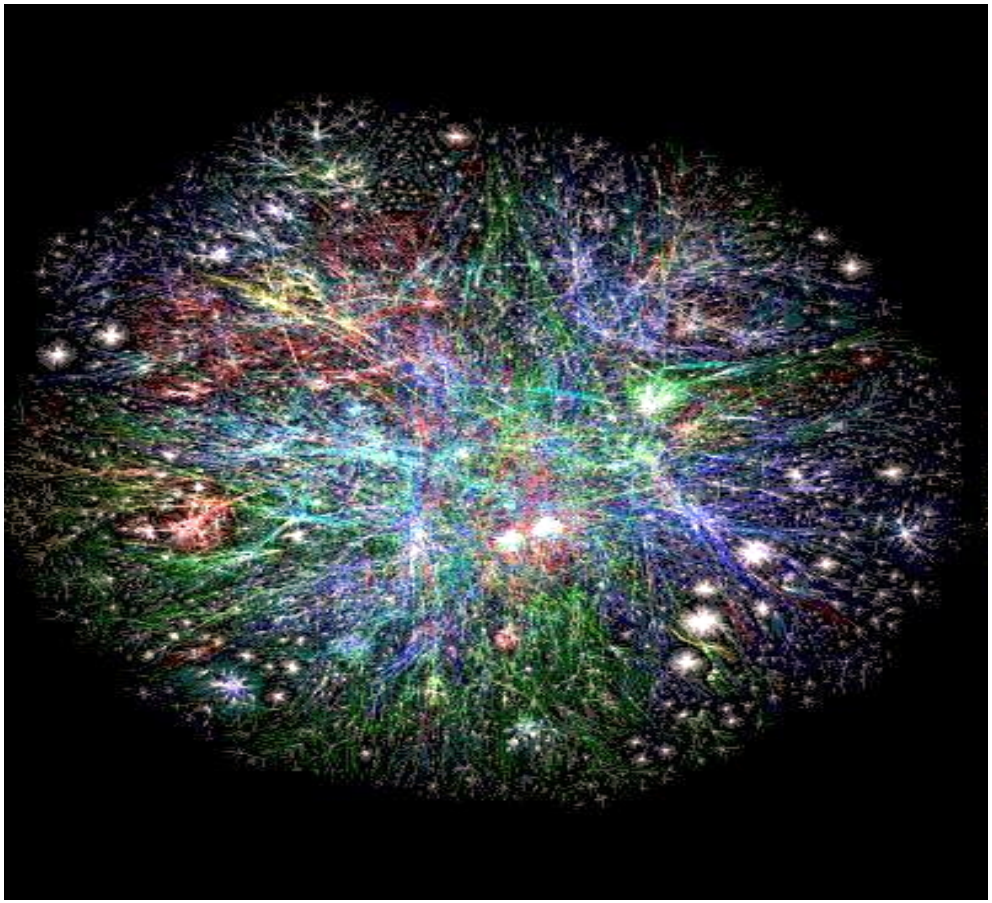
BitTorrent



Nodes?

Communication links?

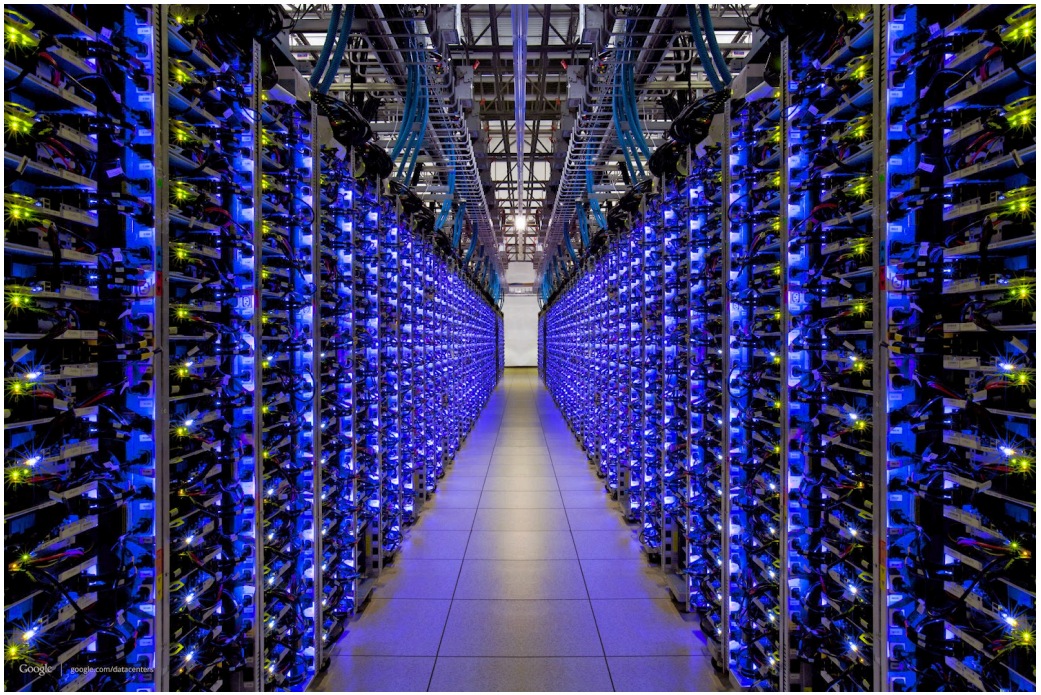
Web Domains



Nodes?

Communication links?

Data Center



Nodes?

Communication links?

Goals

- Overcome geographic separation
  - Think Google, Facebook …
- Build reliable systems out of unreliable components
  - How many computers in a modern data center? How many disks? How often do they fail?
- Aggregate systems for higher capacity, customize for specific tasks
  - Example: Web server

What We Want from a DS

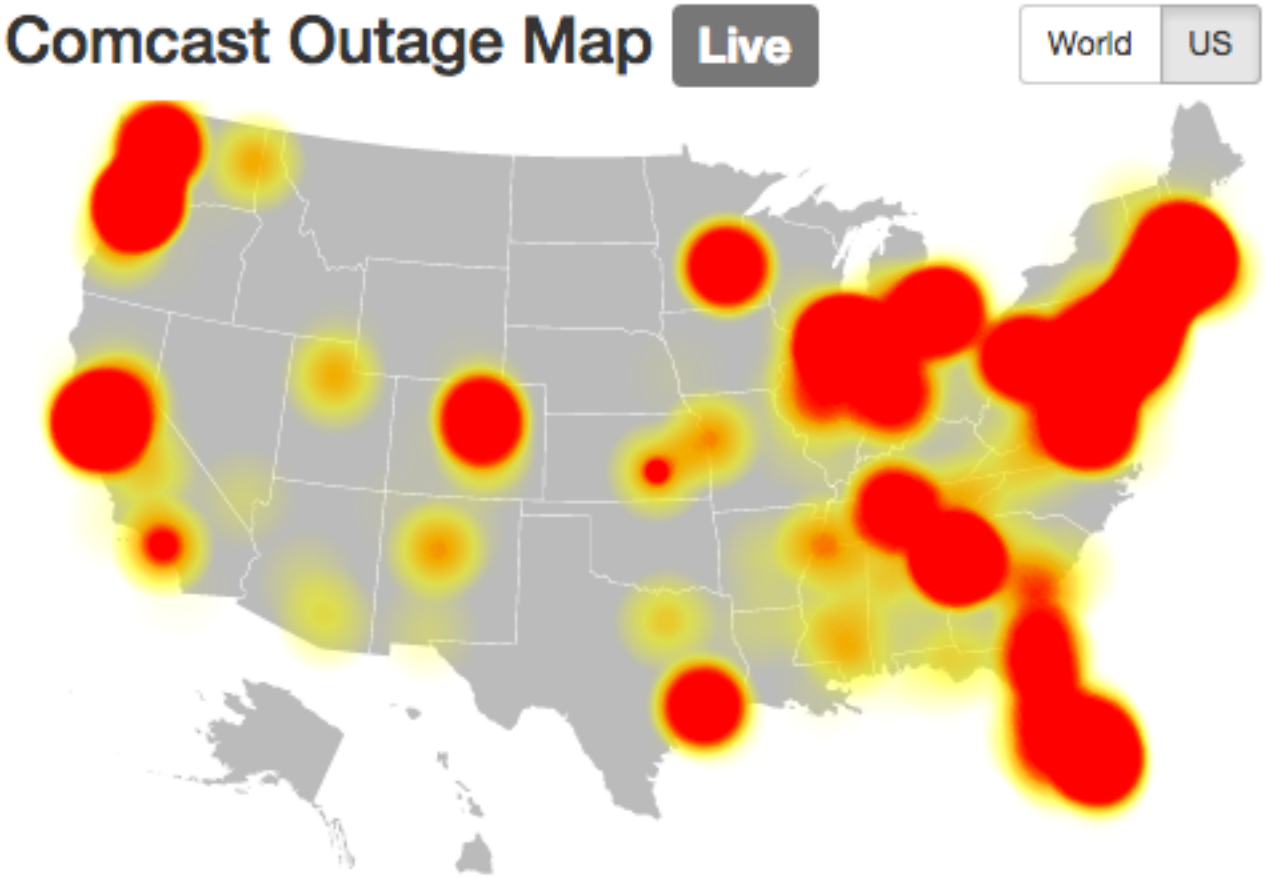
- Fault tolerant
- Highly available
- Recoverable
- Consistent
- Scalable
- Transparent
- Predictable performance
- Secure

Failure is what distinguishes distributed from local programming

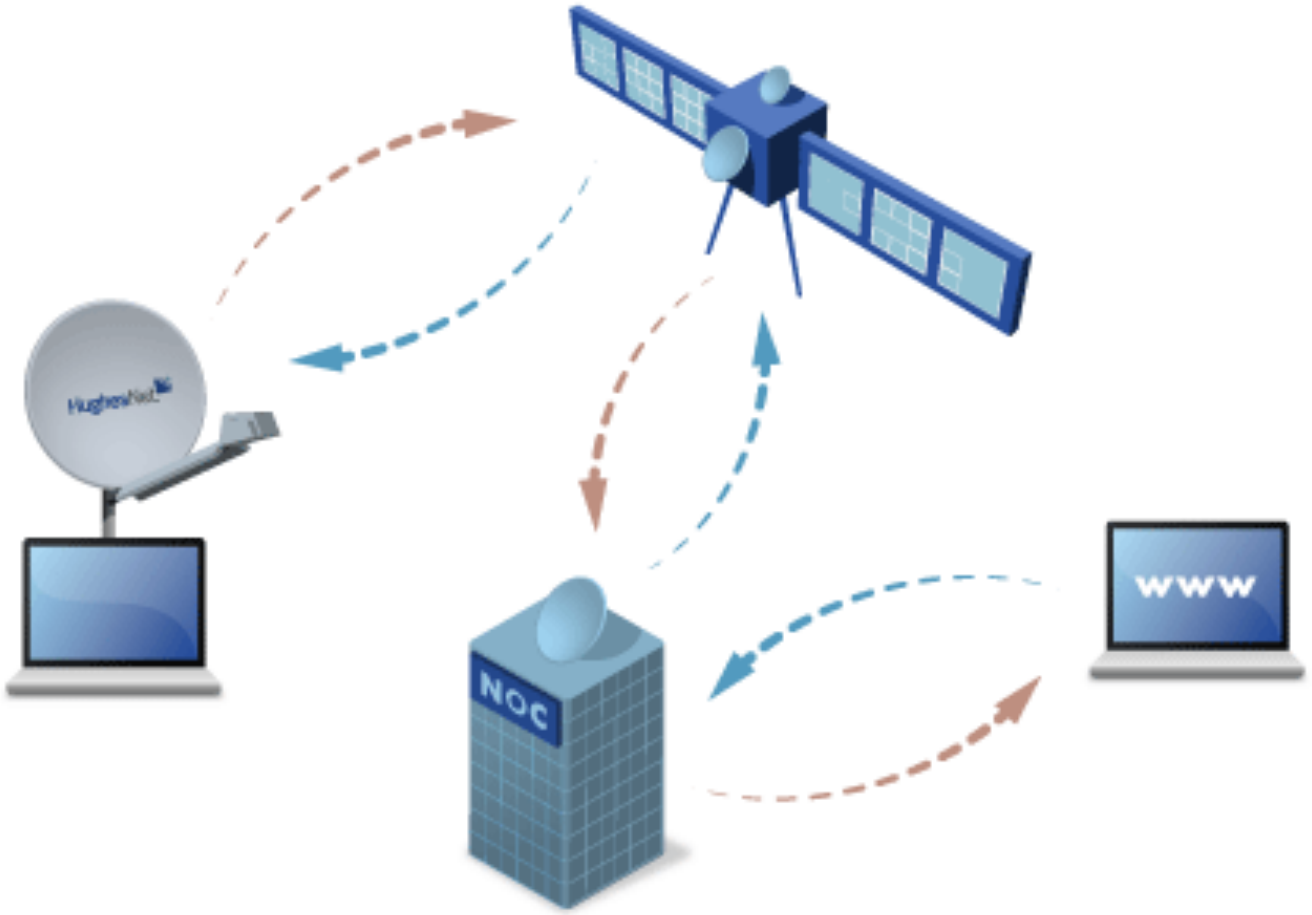
The 8 Fallacies

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn’t change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

“The Network is Reliable”

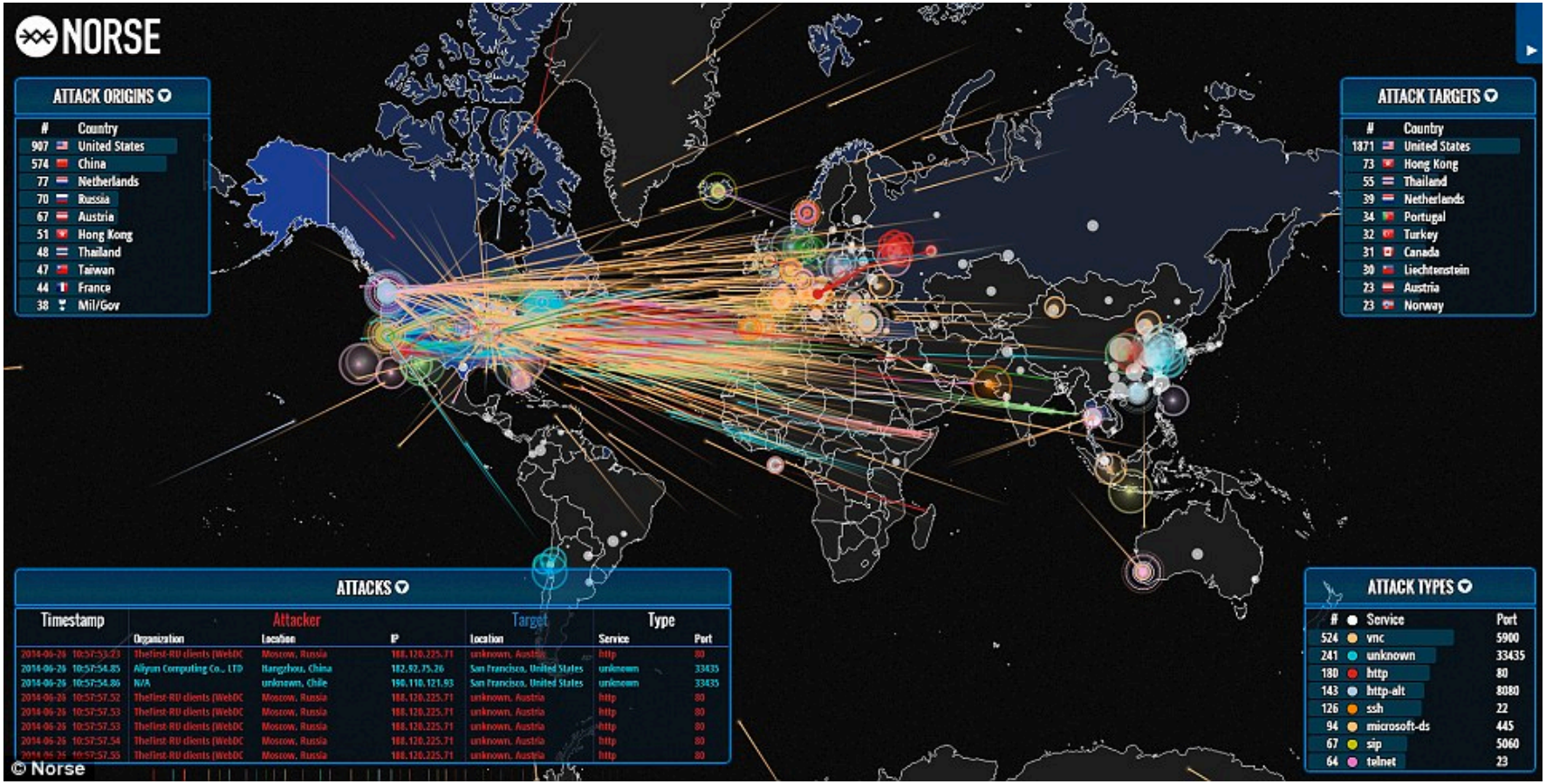


“Latency Is Zero”



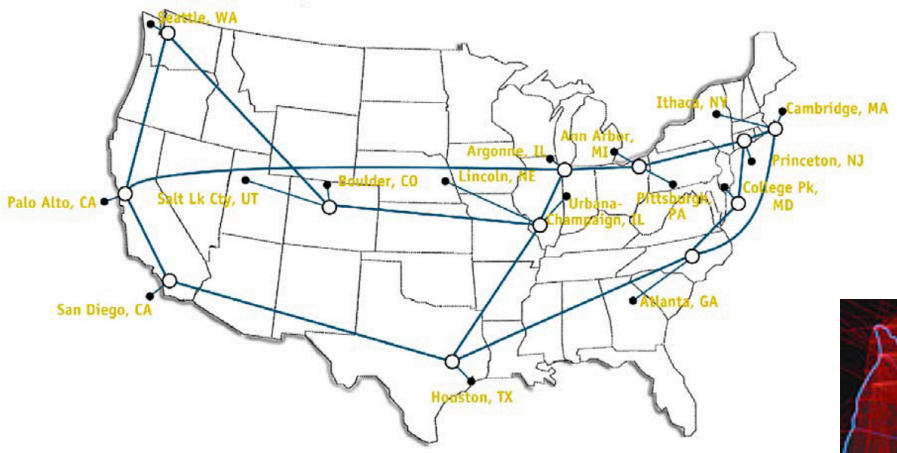
“Bandwidth Is Infinite”

“The Network Is Secure”

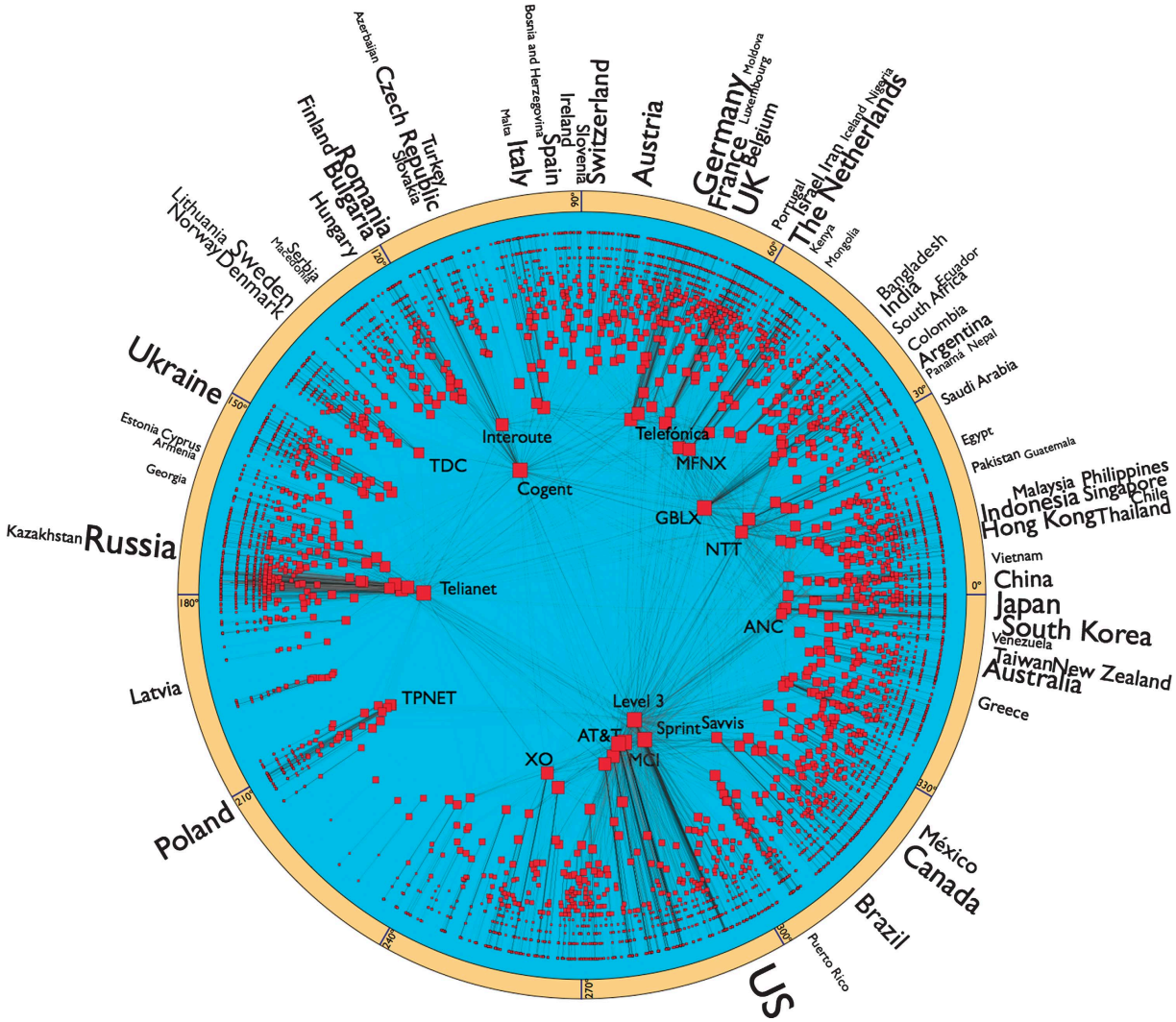


"Topology Doesn't Change"

NSFNET T3 Network 1992



“There Is One Administrator”



“Transport Cost Is Zero”

“The Network Is Homogeneous”

How Things Goes Wrong

- Halting failure (stop silently)
- Fail-stop (notify other components)
- Omission failure (silently fail to send message)
- Network failure
- Network partition failure
- Timing failure (unsynchronized clocks, long delays, etc.)
- Byzantine failure
  - Data corruption or loss, malicious attack, etc.

Too Many Abstract Concepts?

- Yes!
- In the rest of the course, will illustrate them using concrete systems
- But concepts are important, too...

  Which is the more important invention?
  - "Car"
  - "Wheel"
  - "Bicycle"