Internet Measurement

Prof. Dr. Thomas Schmidt

http://inet.haw-hamburg.de | t.schmidt@haw-hamburg.de

Agenda

Measuring the Internet ecosystem

Examples of measurements

Principle approaches to measurement

Measurement and ethics

Internet-wide scanning

Objectives of this lecture

Better understanding of the current Internet ecosystem

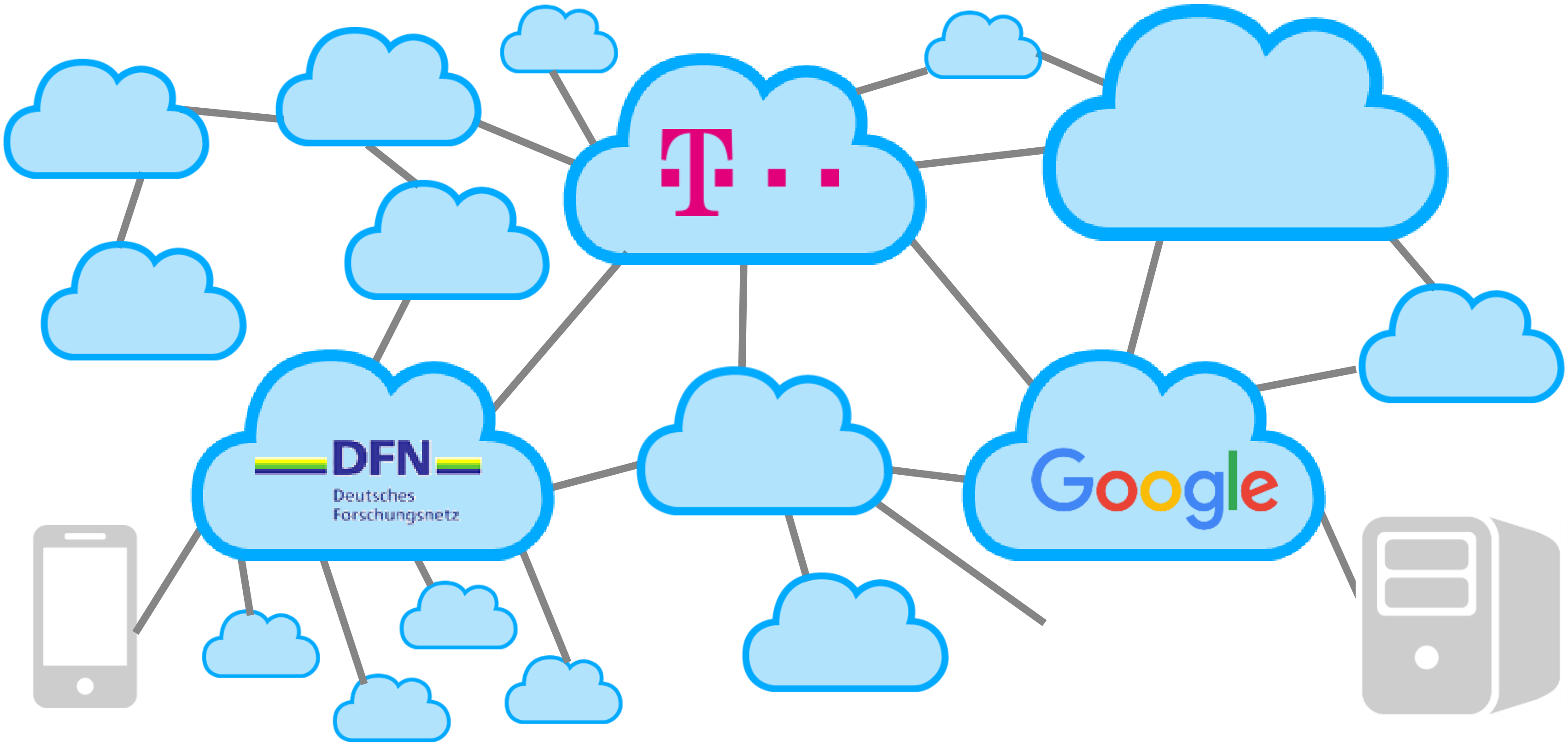
Mastering the assessment of protocol and application deployment

Understanding of potentials and limitations of Internet measurement (data)

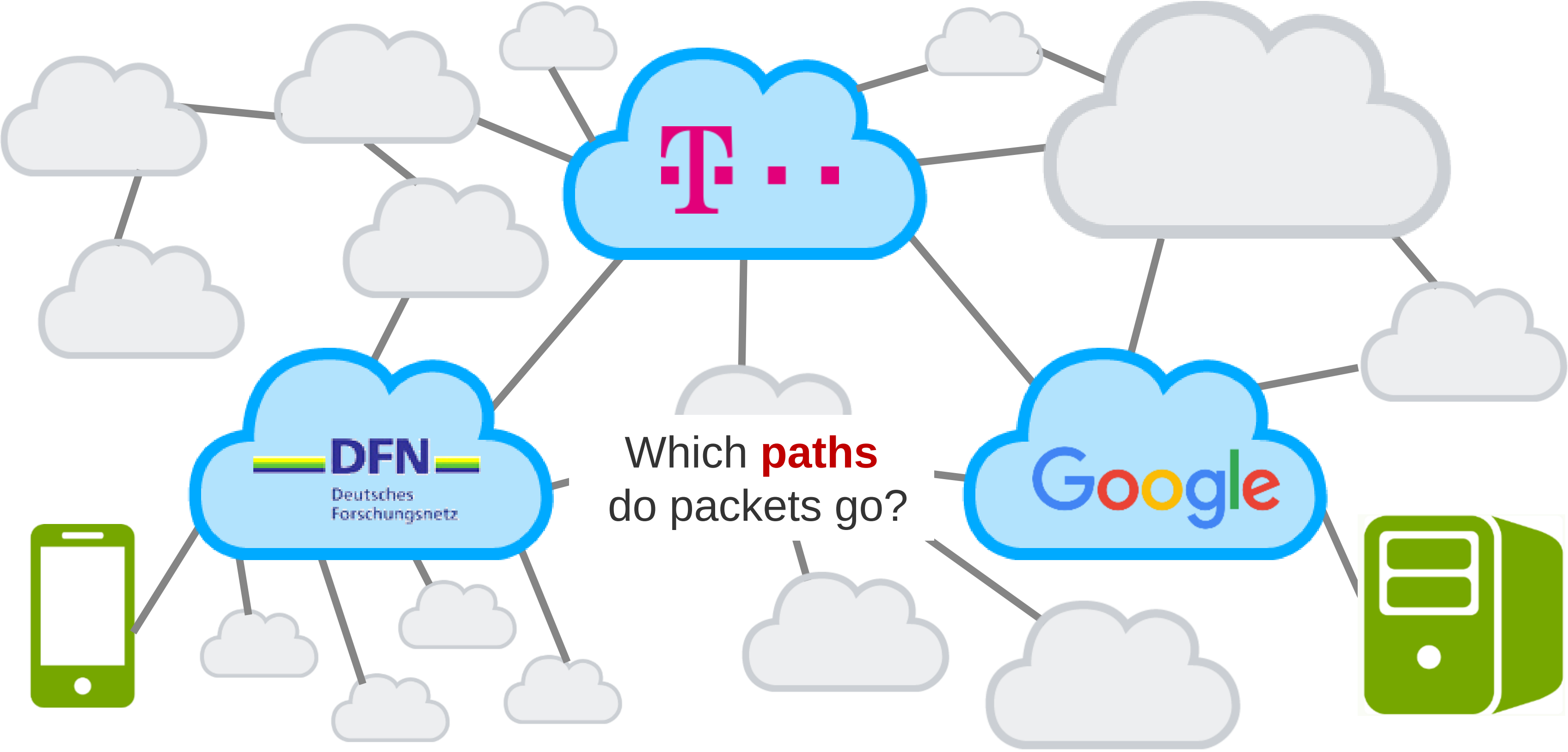
Why measuring the Internet ecosystem?

- Network Debugging
- Performance
- Resilience
- Security
- Regulation and Policies
- Broader impact on society: state censorship, price and traffic discrimination, impact of social media, …

Which part of the Internet do we consider?



From control plane to data plane



From control plane to data plane

Which **service** is available?

Which **traffic** is exchanged?

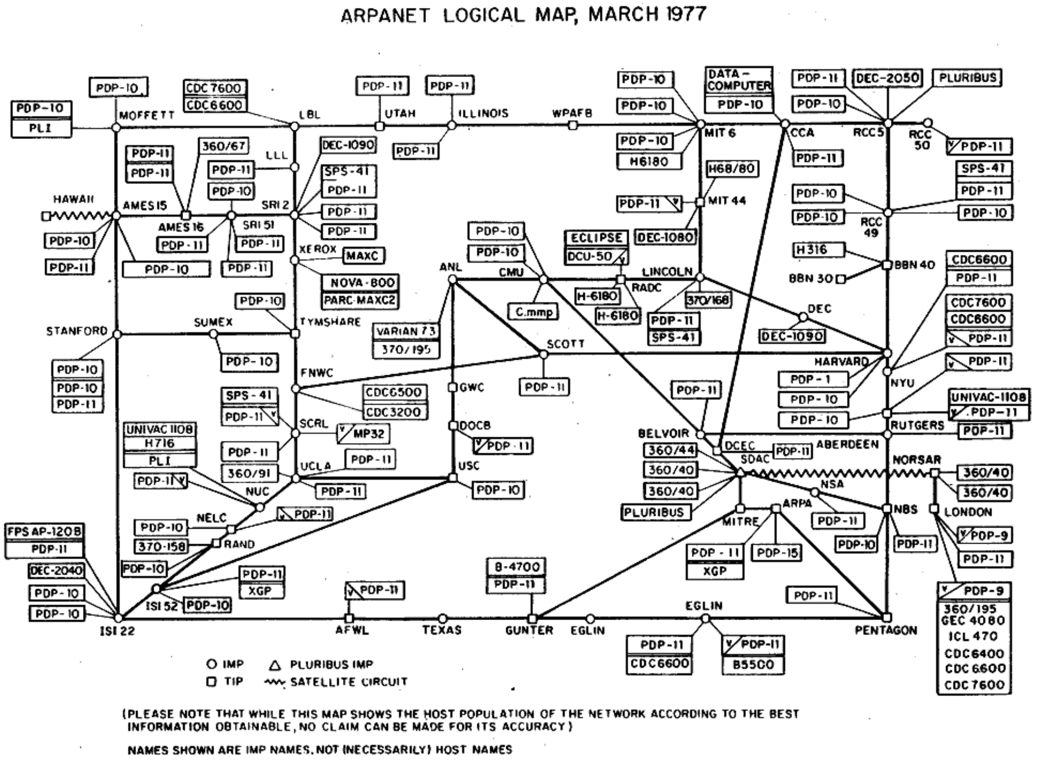
Example 1: ARPANET Routing

1802 IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-26, NO. 12, DECEMBER 1978

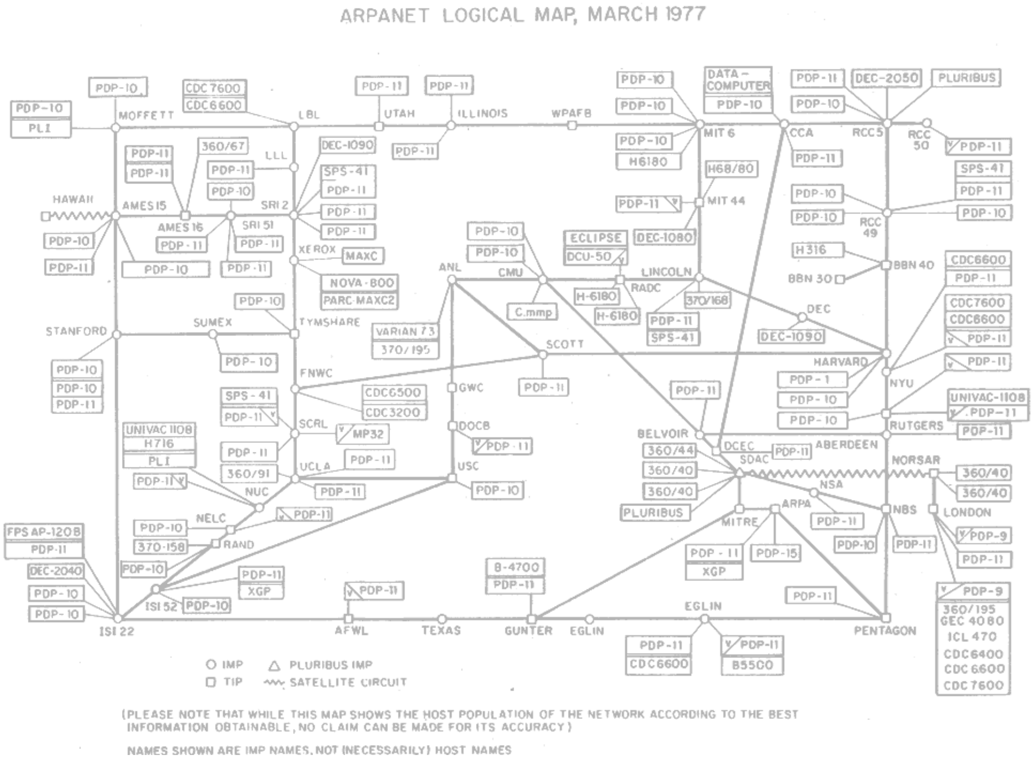
A Review of the Development and Performance of the ARPANET Routing Algorithm

JOHN M. McQUILLAN, MEMBER, IEEE, GILBERT FALK, MEMBER, IEEE, AND IRA RICHER, MEMBER, IEEE

Example 1: ARPANET Routing



Example 1: ARPANET Routing



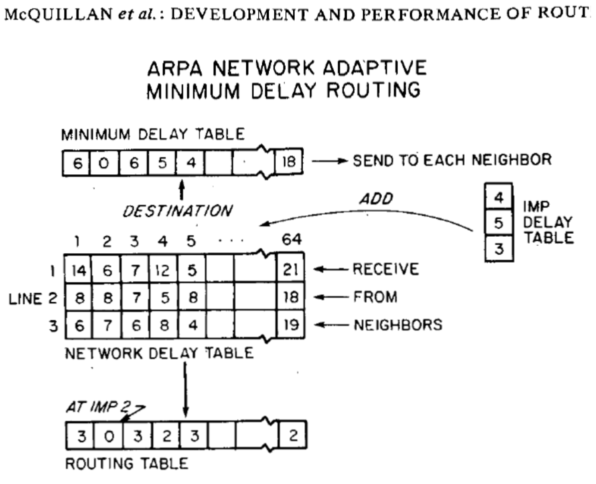


Figure 1 ARPANET Routing Algorithm Tables

Every 2/3 of a second, IMP selects the minimum delay to each destination.
Every 2/3 of a second, IMP sends minimum delay table to neighbors.

What can we say about basic performance properties?

1. Information travels every 2/3 of a second per interface line.
   Topology changes are recognized by the whole network in few seconds.
2. Calculates path of least delay.
   Low frequency of routing updates means that the estimated traffic delay is a function of past traffic, may result in oscillations and bad line usage.
3. It’s simple. No complete network knowledge necessary.
4. Not costly in terms of network resources.
   Calculation of min delay proportional to the number of nodes and lines.

There are also drawbacks …

There are also drawbacks …

5. NEW PROBLEMS

For several years the ARPANET has been subjected to occasional disturbances stemming from a variety of external causes: faulty IMP hardware, software bugs, circuit difficulties, traffic overloads (stochastic), etc. The real problem is not any particular irritant but the vulnerability of the ARPANET to congestion caused by such irritants [5].

What is a network disturbance? We can offer three common characteristics:

1. The NCC host detects that some of the normal periodic IMP reports are missing.

2. Some IMPs declare other IMPs in the network unreachable (when in fact the IMPs are reachable).

3. Users in the network see their connections broken.

These events appeared to be closely correlated to IMPs retransmitting packets many times to adjacent IMPs. When an IMP retransmits a packet 600 times (which takes at least 75 s), it declares the line down.

Challenges when measuring …

Determining the causes of ARPANET disturbances is a complex and difficult task given the nature of the network: the IMPs have limited memory and must communicate with the NCC by means of the same circuits that are involved in a disturbance. We have developed a flexible set of measurement programs in the IMP program which allows us to take a snapshot of a given set of data (queue lengths, buffer counts, etc.) whenever a network disturbance occurs. When the disturbance has ended, a single command from the NCC causes all the IMPs to transmit their data to the NCC.

Measurement setup

We have used this measurement package to analyze a total of 36 network disturbances which occurred in the period July to September 1977. Of this total, 19 were spontaneously occurring disturbances of various magnitudes and 17 were disturbances which we provoked artificially. We used the two-hour period from 7-9 a.m. on Tuesday mornings (a time reserved for ARPANET software maintenance) to conduct experiments. We used various means (making a line appear to be up in one direction only, making an IMP artificially slow, etc.) to induce congestion in one region of the network, which then led to network disturbances. The utmost caution must be used in creating such disturbances since too severe a test can readily disrupt all network service. Thus we designed all of our experiments to minimize risk by programming the experimental module to deactivate itself after a fixed interval of time.

Major result

The basic cause of the disturbances seen over the last several years in the ARPANET is that the network has no built-in protection against traffic congestion [1]. That is, when the offered traffic in some region of the network exceeds the region's capacity to carry that traffic, then congestion builds up throughout that region and sometimes throughout the network as a whole. Eventually, the network is so full of traffic for the congested area that little or no other traffic can flow through the network. The disturbance reaches a climax when the IMPs in the affected regions determine that they have retransmitted certain packets more than the nominal limit (which had been set at 600 retransmissions). At this point the IMPs declare the circuits to be unusable. This isolates the region of congestion from the rest of the network and permits normal operations to resume, although any user with a host-to-host protocol connection in the affected region would find his connection broken.

Example 2: BGP Experiment

Background

Border Gateway Protocol (BGP) allows for different path attributes types (e.g., AS path, next hop, local preference).

One path attribute type is reserved for development.

Example 2: BGP Experiment

NANOG,

We would like to inform you of an experiment to evaluate alternatives for speeding up adoption of BGP route origin validation (research paper with details [A]).

Our plan is to announce prefix 184.164.224.0/24 with a valid standards-compliant unassigned BGP attribute from routers operated by the PEERING testbed [B, C]. The attribute will have flags 0xe0 (optional transitive [rfc4271, S4.3]), type 0xff (reserved for development), and size 0x20 (256bits).

Our collaborators recently ran an equivalent experiment with no complaints or known issues [A], and so we do not anticipate any arising. Back in 2010, an experiment using unassigned attributes by RIPE and Duke University caused disruption in Internet routing due to a bug in Cisco routers [D, CVE-2010-3035]. Since then, this and other similar bugs have been patched [e.g., CVE-2013-6051], and new BGP attributes have been assigned (BGPsec-path) and adopted (large communities). We have successfully tested propagation of the announcements on Cisco IOS-based routers running versions 12.2(33)SRA and 15.3(1)S, Quagga 0.99.23.1 and 1.1.1, as well as BIRD 1.4.5 and 1.6.3.

We plan to announce 184.164.224.0/24 from 8 PEERING locations for a predefined period of 15 minutes starting 14:30 GMT, from Monday to Thursday, between the 7th and 22nd of January, 2019 (full schedule and locations [E]). We will stop the experiment immediately in case any issues arise.

Although we do not expect the experiment to cause disruption, we welcome feedback on its safety and especially on how to make it safer. We can be reached at disco-experiment at googlegroups.com.

First wave of issues

NANOG,

We've performed the first announcement in this experiment yesterday, and, despite the announcement being compliant with BGP standards, FRR routers reset their sessions upon receiving it. Upon notice of the problem, we halted the experiments. The FRR developers confirmed that this issue is specific to an unintended consequence of how FRR handles the attribute 0xFF (reserved for development) we used. The FRR devs already merged a fix and notified users.

We plan to resume the experiments January 16th (next Wednesday), and have updated the experiment schedule [A] accordingly. As always, we welcome your feedback.

Second round

NANOG,

This is a reminder that this experiment will resume tomorrow (Wednesday, Jan. 23rd). We will announce 184.164.224.0/24 carrying a BGP attribute of type 0xff (reserved for development) between 14:00 and 14:15 GMT.

Can you stop this?

You caused again a massive prefix spike/flap, and as the internet is not centered around NA (shock horror!) a number of operators in Asia and Australia go effected by your "expirment" and had no idea what was happening or why.

Get a sandbox like every other researcher, as of now we have black holed and filtered your whole ASN, and have reccomended others do the same.

```
Ben, NANOG,

We have canceled this experiment permanently.
```

- BGP Experiment *valdis.kletnieks at vt.edu*
- BGP Experiment *Tom Beecher*
- BGP Experiment *Randy*
- BGP Experiment *Mark Tees*
- BGP Experiment *Mark Tees*
- BGP Experiment *Randy Bush*
- BGP Experiment *Owen DeLong*
- BGP Experiment *valdis.kletnieks at vt.edu*
- BGP Experiment *Owen DeLong*
- BGP Experiment *Randy Bush*
- BGP Experiment *Eric Kuhnke*
- BGP Experiment *Randy Bush*
- BGP Experiment *William Allen Simpson*
- [2019/01/27] Re: BGP Experiment *Hansen, Christoffer*
- BGP Experiment *Randy Bush*
- BGP Experiment *Nick Hilliard*
- BGP Experiment *Brian Kantor*
- BGP Experiment *Nick Hilliard*

- BGP Experiment *Italo Cunha*
  - BGP Experiment *Job Snijders*
  - BGP Experiment *Eric Kuhnke*
  - BGP Experiment *Naslund, Steve*
  - BGP Experiment *Aled Morris*
  - BGP Experiment *Töma Gavrichenkov*
  - BGP Experiment *Naslund, Steve*
  - BGP Experiment *Naslund, Steve*
  - BGP Experiment *Töma Gavrichenkov*
  - BGP Experiment *Nick Hilliard*
  - BGP Experiment *Filip Hruska*
  - BGP Experiment *Naslund, Steve*

Example 3: Caching & DNS

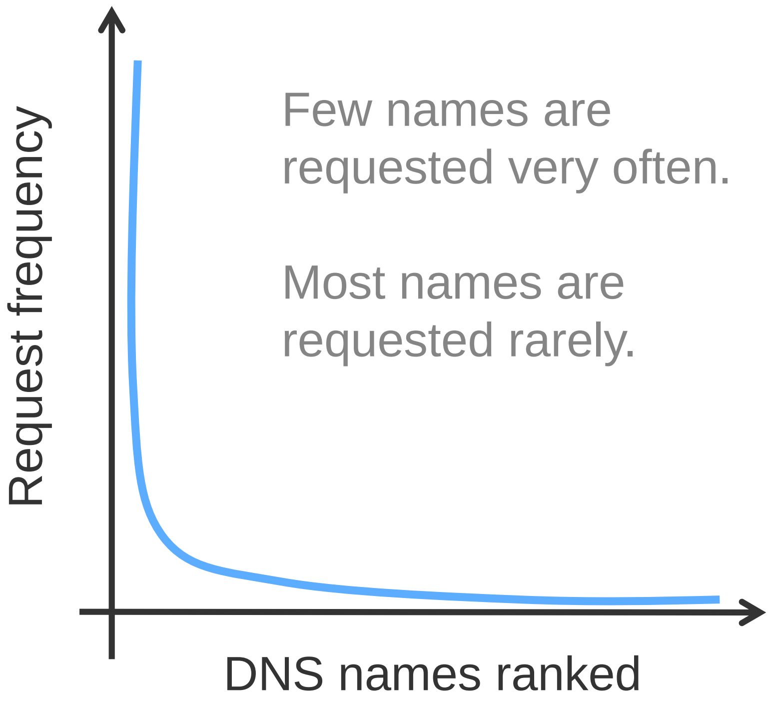
Is caching a reasonable design option in DNS?

Example 3: Caching & DNS

Is caching a reasonable design option in DNS?

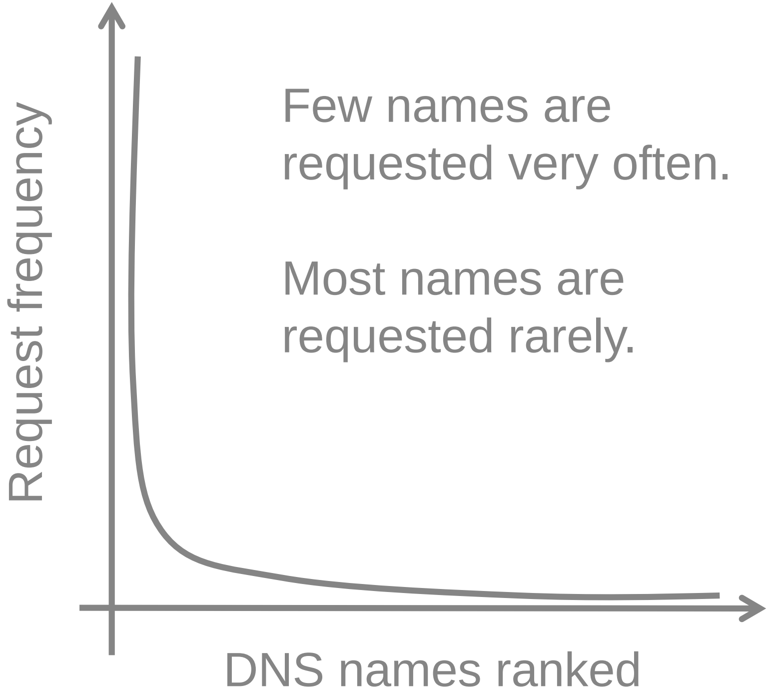
Depends how often the same name is requested by resolvers.

Example 3: Caching & DNS



Example 3: Caching & DNS

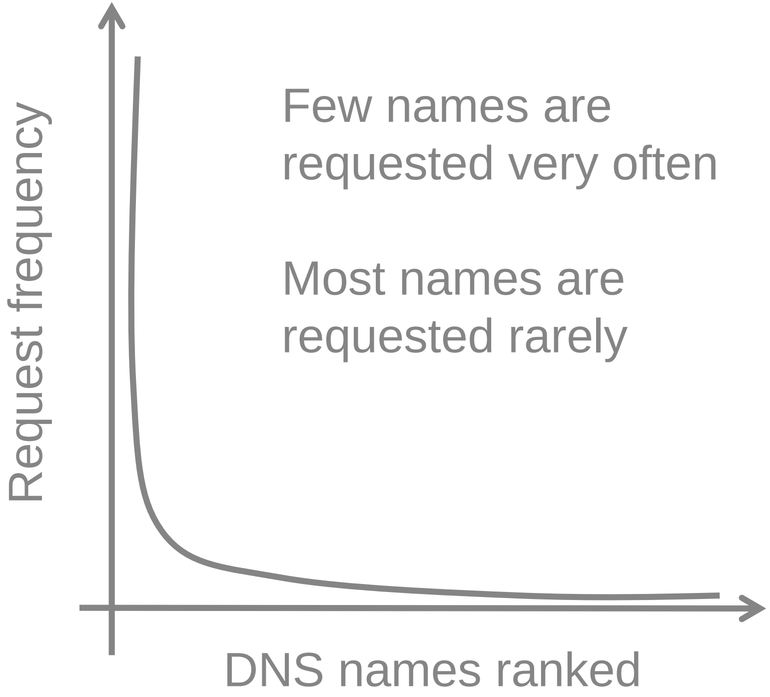
Why should you not trust the results?



Example 3: Caching & DNS

Why should you not trust the results?

You don't know anything about the measurement setup!



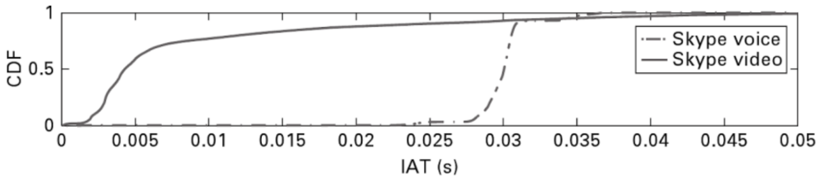
Discuss two measurement setups that lead to completely different results.

Example 4: Classification of multimedia flows

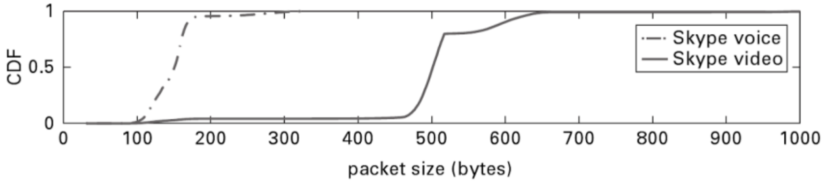
Motivation	ISPs want to understand what happens in their network for business, QoS, and security reasons
Consideration	Voice and video flows
Background	Voice sender uses fix inter-packet delay Voice packets are similar and small
	Video frames vary in size and complexity Video smoothes out transmission intervals

Example 4: Classification of multimedia flows

Skype



(a)



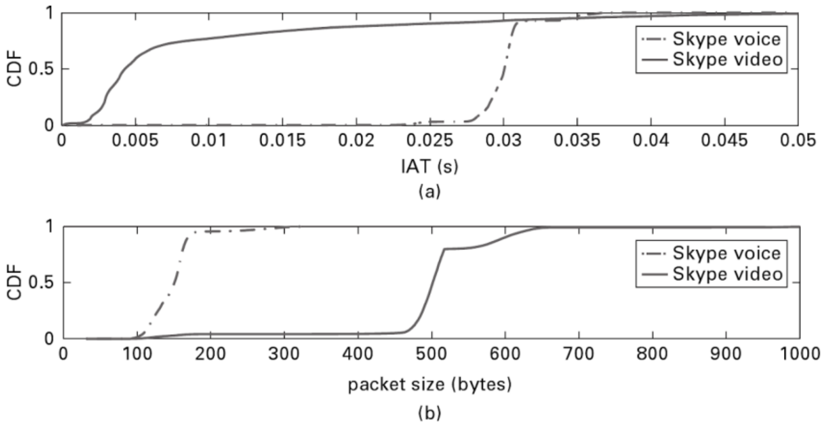
(b)

IAT: Inter-arrival time
CDF: Cumulative distribution function, F_X(x)=P(X<=x)

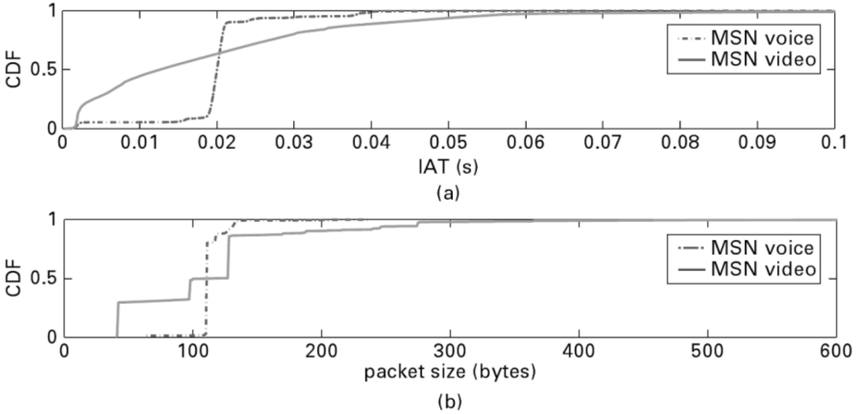
[A. Nucci and K. Papagiannaki, “Design, Measurement and Management of Large-Scale IP Network,” Cambridge University Press, 2009.]

Example 4: Classification of multimedia flows

Skype



MSN



IAT: Inter-arrival time
CDF: Cumulative distribution function, F_X(x)=P(X<=x)

[A. Nucci and K. Papagiannaki, “Design, Measurement and Management of Large-Scale IP Network,” Cambridge University Press, 2009.]

What did we learn from the examples?

There are different measurement techniques

Clear descriptions of the experiments and measurement data are crucial

Be careful when your experiment runs in the real Internet

Different implementation of the same service may lead to different patterns

Internet measurements: Classic topics

Transport layer	e.g., performance of transport protocols, congestion control
Network layer	e.g., routing failures, Internet topology, performance

[Slide from Philipp Richter, 2018]

Internet measurements: Broadening field

Layer	Example
“Layer 8” User/political layer	e.g., (fake) news propagation in social networks
Application layer	e.g., cloud services, specific applications
Transport layer	e.g., performance of transport protocols, congestion control
Network layer	e.g., routing failures, Internet topology, performance
Physical layer	e.g., infrastructure properties, location

[Slide from Philipp Richter, 2018]

Internet measurements: Cross-layer measurements

“Layer 8”
User/political layer

Application layer

Transport layer

Network layer

Physical layer

e.g., censorship measurements and impact

e.g., measurement of end-to-end video quality

[Slide from Philipp Richter, 2018]

Internet measurement: A creative field

Demystifying Porn 2.0: A Look into a Major Adult Video Streaming Website

Gareth Tyson
Queen Mary, University of London, UK
gareth.tyson@qmul.ac.uk

Yehia Elkhatib
Lancaster University, UK
yehia@comp.lancs.ac.uk

Nishanth Sastry
King's College London, UK
nishanth.sastry@kcl.ac.uk

Steve Uhlig
Queen Mary, University of London, UK
steve@eecs.qmul.ac.uk

Email Typosquatting

Janos Szurdi
Carnegie Mellon University
jszurdi@andrew.cmu.edu

Nicolas Christin
Carnegie Mellon University
nicolasc@andrew.cmu.edu

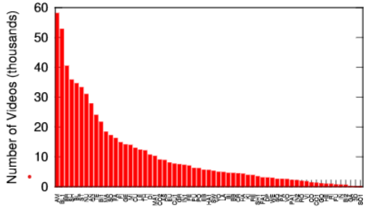


Figure 10: Number of videos per category (ordered by number of videos in the category).

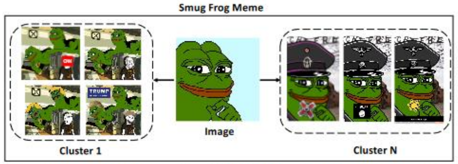


Figure 1: An example of a meme (Smug Frog) that provides an intuition of what an image, a cluster, and a meme is.

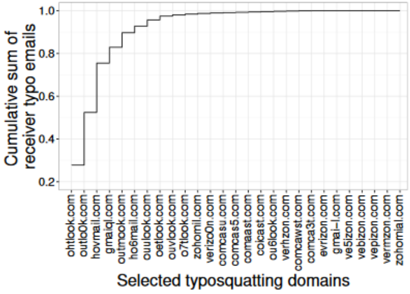


Figure 5: Cumulative sum of emails received by our typosquatting domains.

On the Origins of Memes by Means of Fringe Web Communities

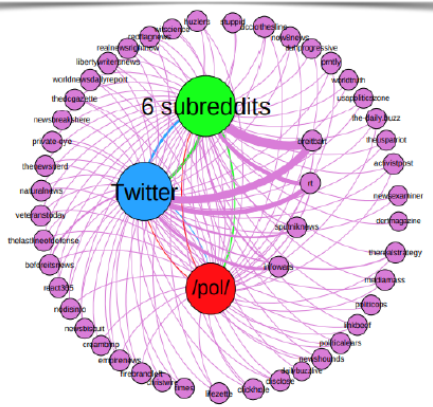
Savvas Zannettou★, Tristan Caulfield‡, Jeremy Blackburn†, Emiliano De Cristofaro‡,
Michael Sirivianos★, Gianluca Stringhini⋄, and Guillermo Suarez-Tangil+

[Inspired by Philipp Richter, 2018]

Internet measurement: Broader societal impact

The Web Centipede: Understanding How Web Communities Influence Each Other Through the Lens of Mainstream and Alternative News Sources

Savvas Zannettou*, Tristan Caulfield†, Emiliano De Cristofaro†, Nicolas Kourtellis‡, Ilias Leontiadis‡, Michael Sirivianos*, Gianluca Stringhini†, and Jeremy Blackburn+



Your State is Not Mine: A Closer Look at Evading Stateful Internet Censorship

Zhongjie Wang, zwang048@ucr.edu, University of California, Riverside
Yue Cao, ycao009@ucr.edu, University of California, Riverside
Zhiyun Qian, zhiyunq@cs.ucr.edu, University of California, Riverside
Chengyu Song, csong@ucr.edu, University of California, Riverside
Srikanth V. Krishnamurthy, krish@cs.ucr.edu, University of California, Riverside

Examining How the Great Firewall Discovers Hidden Circumvention Servers

Roya Ensafi, Princeton University
David Fifield, UC Berkeley
Philipp Winter, Karlstad & Princeton University
Nick Feamster, Princeton University
Nicholas Weaver, UC Berkeley & ICSI
Vern Paxson, UC Berkeley & ICSI

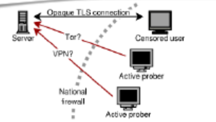


Figure 1: The firewall cannot determine, by mere inspection, whether the encrypted connection carries a prohibited circumvention protocol. Therefore it issues its own probes and observes how the server responds.

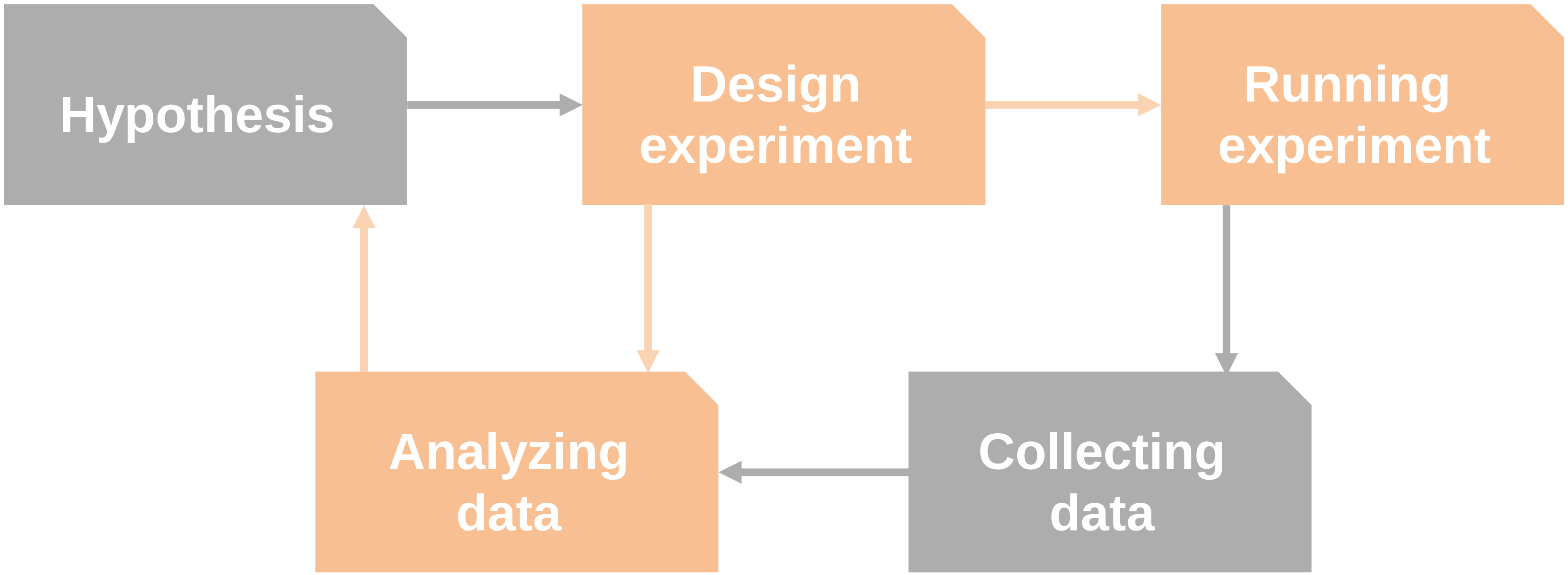
Censorship in the Wild: Analyzing Internet Filtering in Syria

Abdelberi Chaabane, INRIA Rhône-Alpes, Montbonnot, France
Terence Chen, NICTA, Sydney, Australia
Mathieu Cunche, University of Lyon & INRIA, Lyon, France
Emiliano De Cristofaro, University College London, London, United Kingdom
Arik Friedman, NICTA, Sydney, Australia
Mohamed Ali Kaafar, NICTA & INRIA Rhône-Alpes, Sydney, Australia

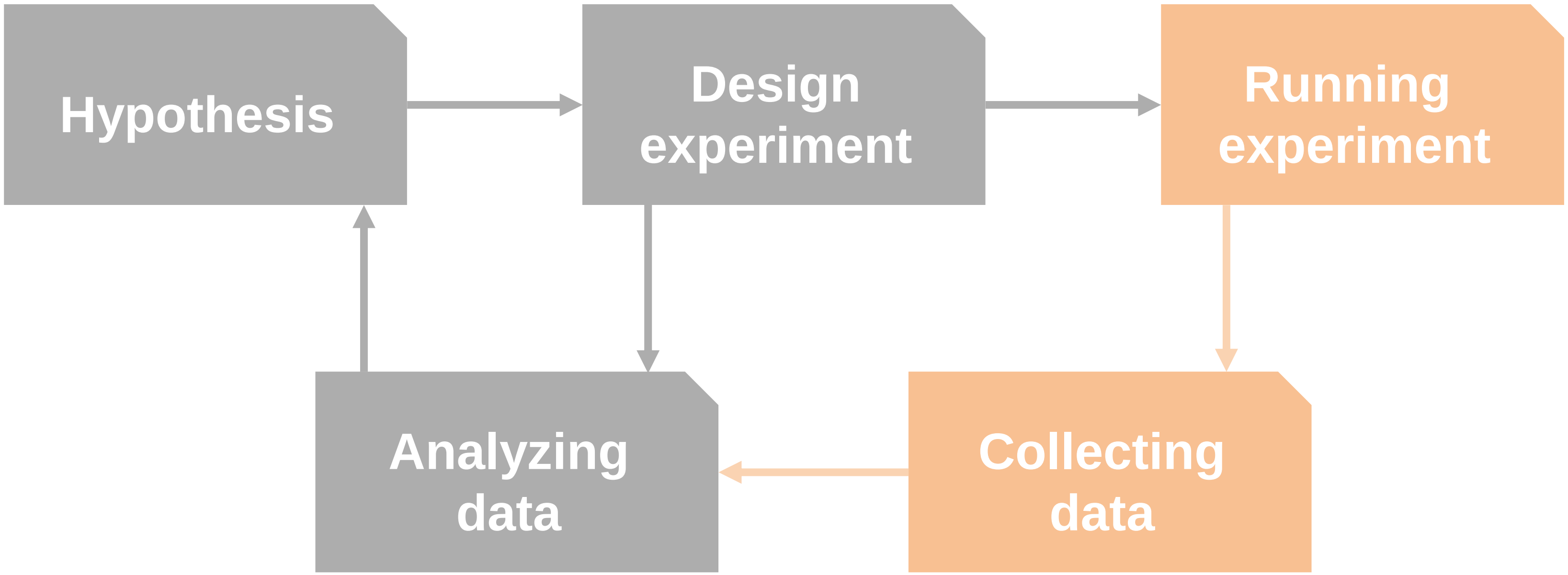
[Slide from Philipp Richter, 2018]

Typical measurement life cycle

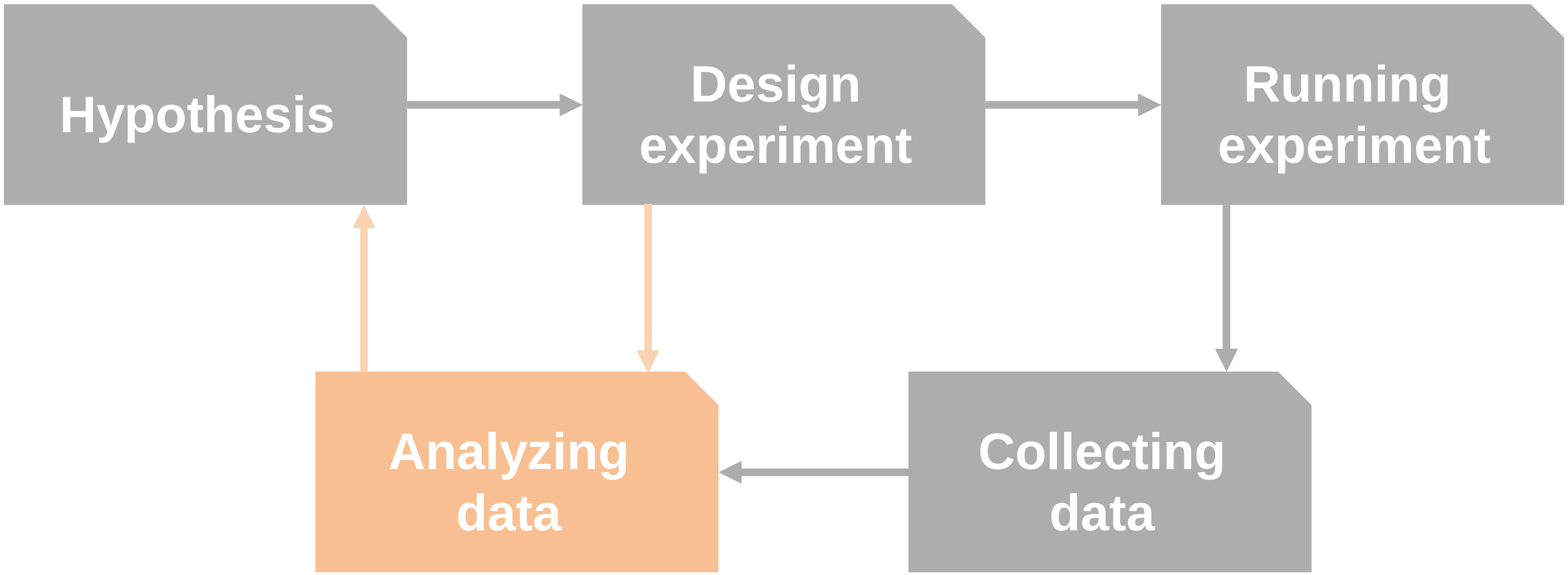
Typical measurement life cycle



Typical measurement life cycle



Typical measurement life cycle



There are two types of experiments

How to test a hypothesis

Uncontrolled experiments

Factor of interest varies outside the control of the researcher and independent of the research question.

Controlled experiments

You vary one factor of interest, then you measure the outcome.

There are two types of measurements

How data are collected

Passive measurements

You observe data that is collected independently of your experiment.

Active measurements

You inject probe traffic in the network. More intrusive.

Classification of controlled versus uncontrolled describes **experiments** (**how to test a hypothesis**) is orthogonal to the classification of passive versus active **measurements** (**how data are collected**), and passive versus active measurements are orthogonal to control plane versus data plane measurements (**what data are collected**).

Example: Distribution of IP path lengths

Passive measurement	Each node dumps forwarding table periodically
Active measurement	External node performs traceroutes
Uncontrolled experiment	Analysis of external (traceroute/FIB) dumps
Controlled experiment	You select the nodes that dump information, or the destinations
Data plane	Forwarding information base or traceroute replies
Control plane	BGP dumps

Human subject experiments

Likely require approval by an institutional review board (IRB) or ethics panel

You should document key considerations for protecting human subjects that anybody replicating your study should be aware of

See “The Menlo Report: Ethical Principles Guiding Information and Communication Technology,” 2012, and “Applying Ethical Principles to Information and Communication Technology Research: A Companion to the Menlo Report,” 2013

Good example: Spamaltyics [CCS'08]

Study	Analyze the conversion rate of spam campaigns
Approach	Infiltrate a botnet of spam campaigns, manipulate spam messages being relayed through systems under control of researchers
Justification	Neutral actions that strictly reduce harm

Bad examples: Password discovery and Internet Census 2012

Study	(1) Show vulnerability based on default or non-existent passwords (2) Find active IP addresses
Approach	(1) Brute force scanning and dictionary attack (2) Create a scanning botnet
Justification	(1) Not showing how to hack, rather how easy. (2) No justification.

Internet Scanning: Measurement objectives

Which IP address is online?

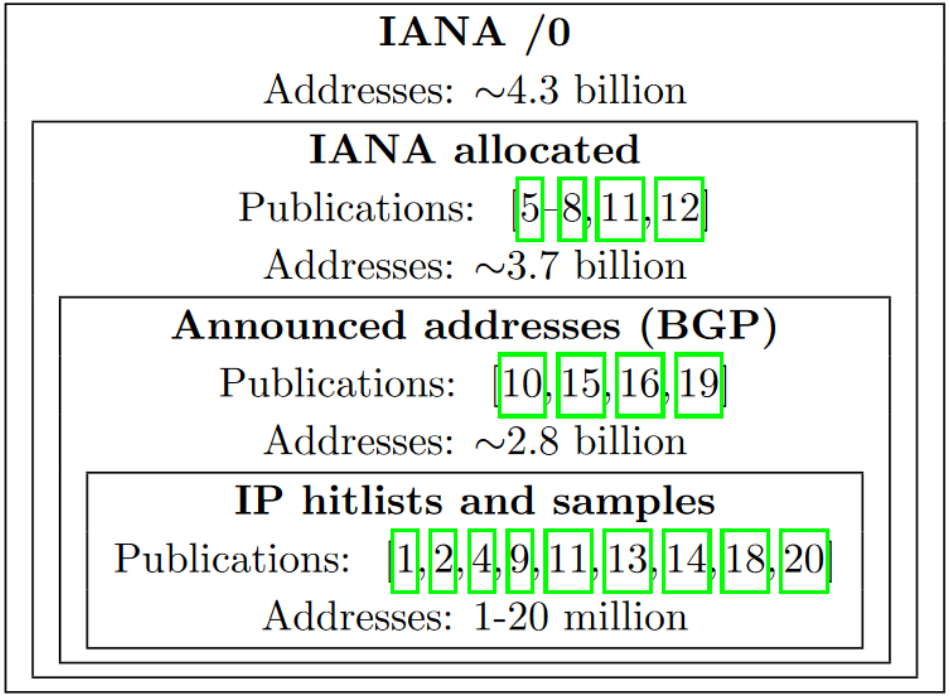
Which IP address runs which service?

You don’t have access to flow data.

You want to answer these questions for (almost) all IP addresses.

Common scanning strategies

IP hitlists are list of IP addresses that most likely offer the scanned services.



Challenges

Target probing	Packet transmission	Packet reception
How to avoid overload of target networks?	How to send packets as fast as possible?	How to identify valid responses?

Challenges

Target probing

How to avoid overload of target networks?

Packet transmission

How to send packets as fast as possible?

Packet reception

How to identify valid responses?

We discuss how ZMap overcomes these challenges compared to common approaches such as nmap.

Target probing

Sending probes to targets in numerical order may easily overload destination networks

Sending probes in random order prevents this problem

How do you know which addresses you already contacted?

Target probing: An inexpensive approach

How do we randomly scan addresses without excessive states?

Core idea

1. Scan hosts according to random permutation
2. Iterate over multiplicative group of integers modulo p

Brief math excursion: Multiplicative cyclic groups

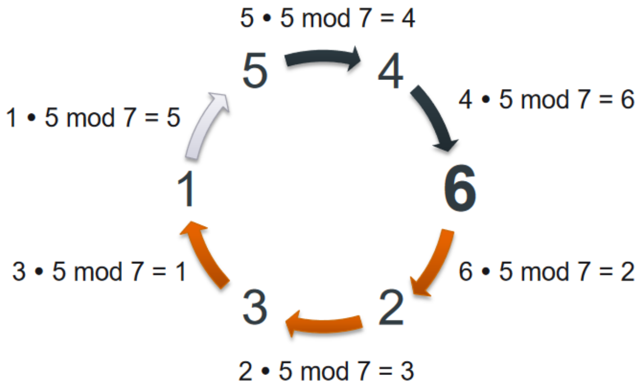
If this is a primitive root, we can iterate over all elements subsequently.

a * r mod p

Group is cyclic if p is prime.
For IPv4: 2^32+15 is the smallest prime larger 2^32.

Target probing: An inexpensive approach, details

a * r mod p



Simplified example [USENIX Security 2013]

Details to generate a fresh random permutation for each scan

1. Generate a primitive
2. Choose a random starting address

Negligible state overhead to store

1. Primitive root
2. Current address
3. Starting address

Common packet transmissions

Sending packets via common socket interface introduces overhead

- Routing table lookup
- ARP cache lookup
- Potential network filters check packets
- TCP handshakes

Fast packet transmissions

Scan packets are different from typical application layer packets.

Send packets directly at the Ethernet layer and enable

- Caching of Ethernet header
  (except checksum header is constant)

- Reduced TCP state management

Validating responses

Problems

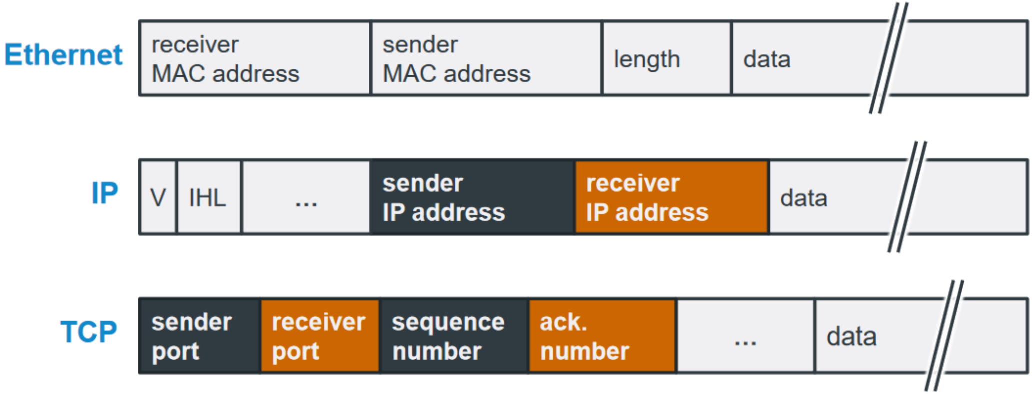
Measurement probe may see unsolicited data (other scan background traffic …)

Per-target states are expensive

Solution

Encode secrets into mutable fields of probe packets that will have recognizable effect on responses

Validating responses



Solution

Encode secrets into mutable fields of probe packets that will have recognizable effect on responses

These ideas have been implemented in ZMap

Simple network scanners

Reduce state by scanning in batches

- Time lost due to blocking
- Results lost due to timeouts

Track individual hosts and retransmit

- Most hosts will not respond

Avoid flooding through timing

- Time lost waiting

Utilize existing OS network stack

- Not optimized for immense number of connections

ZMap

Eliminate local per-connection state

- Fully asynchronous components
- No blocking except for network

Shotgun Scanning Approach

- Always send n probes per host

Scan widely dispersed targets

- Send as fast as network allows

Probe-optimized Network Stack

- Bypass inefficiencies by generating Ethernet frame

Performance of ZMap

Complete scan of v4 address space takes 44 minutes with a gigabit Ethernet connection

Experiment hardware: Xeon E3-1230 3.2 GHz, 4GB RAM