Landing Reinforcement Learning onto Smart Scanning of The Internet of Things

Jian Qu[1], Xiaobo Ma[1], Wenmao Liu[2], Hongqing Sang[2], Jianfeng Li[3], Lei Xue[3], Xiapu Luo[3], Zhenhua Li[4], Li Feng[5], Xiaohong Guan[1]

[1] Xi'an Jiaotong University
[2] NSFOCUS Inc.
[3] The Hong Kong Polytechnic University
[4] Tsinghua University
[5] Wuhan Digital Engineering Institute

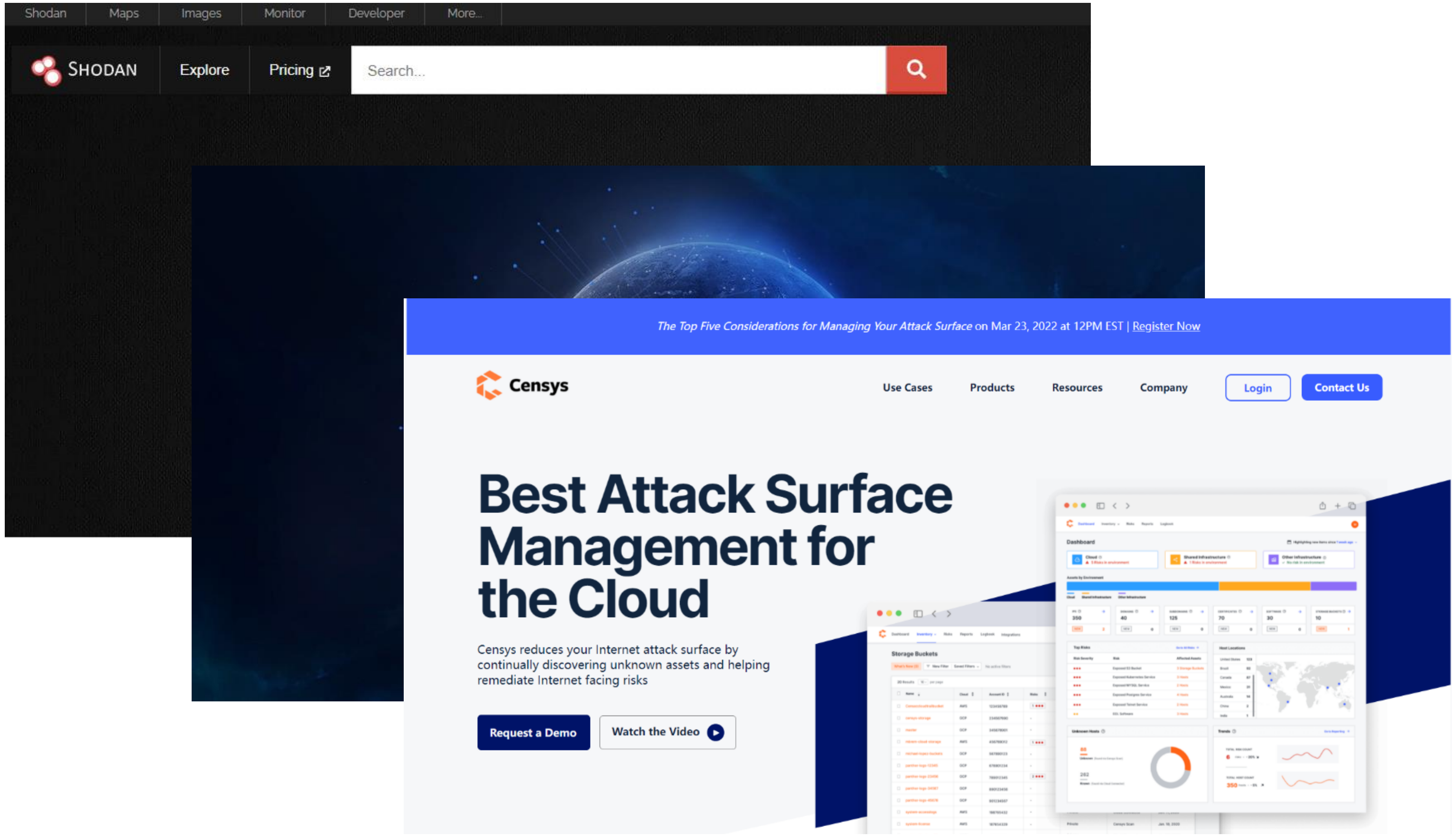
IEEE INFOCOM, May 2022

Outline

- Background and Problem Description
- Understanding IP-device Mapping Dynamics
- System Design
- Evaluation
- Discussion
- Summary

Cyber Search Engines:

Shodan
ZoomEye
Censys
Fofa
BinaryEdge
…



Cyber Search Engines actively scan IoT devices for unearthing IP-device mapping, offering publicly available search engine services.

Research Question

1. Scan rate is limited.

    a) Scanning resources are limited.

    b) High-rate scanning may be blocked by firewalls.

2. IP-device mappings keep changing.

    a) IoT changes their IP addresses.

3. Timeliness is decided by two major aspects:

    Scan rate & Scheduling algorithm.

Research Question: In the case of limited scan rate, can we improve the timeliness performance by optimizing the scan scheduling algorithm?

Outline

- Background and Problem Description
- Understanding IP-device Mapping Dynamics
- System Design
- Evaluation
- Discussion
- Summary

Understanding IP-device Mapping Dynamics

IP-DEVICE MAPPING:

Scan an IP ADDRESS a, get an scan result DEVICE TYPE d, MAPPING $a \rightarrow d$

Three methods to configure one device’s IP address:

Dynamic Host Configuration Protocol (DHCP),
Point-to-Point Protocol (PPP)
Static IP configuration

Measuring IP-device Mapping Dynamics

We scanned the entire IPv4 space for identifying IP cameras.

1. June 5, 2021 (2,896,824 records)
2. June 15, 2021 (3,089,436 records)
3. June 26, 2021 (3,093,510 records)
4. July 6, 2021 (3,076,343 records)

Measuring IP-device Mapping Dynamics

Whether the following two attributes will affect the IP-device Mapping Dynamics?

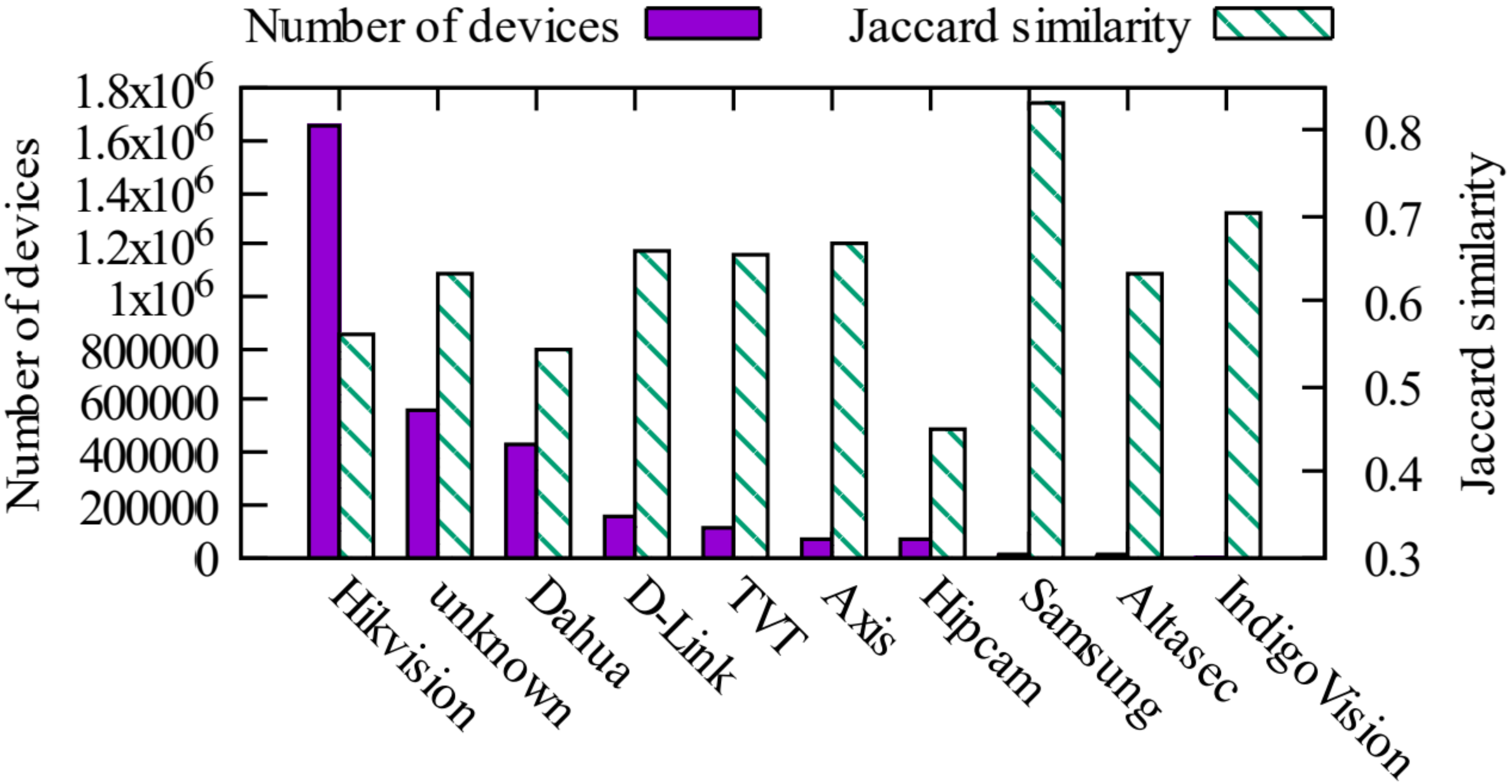
- Device Types
- IP Pools

We employ Jaccard similarity to measure the mapping dynamics between two scans:

$$J(t_1, t_2) = \frac{|S_{t_1} \cap S_{t_2}|}{|S_{t_1} \cup S_{t_2}|}.$$

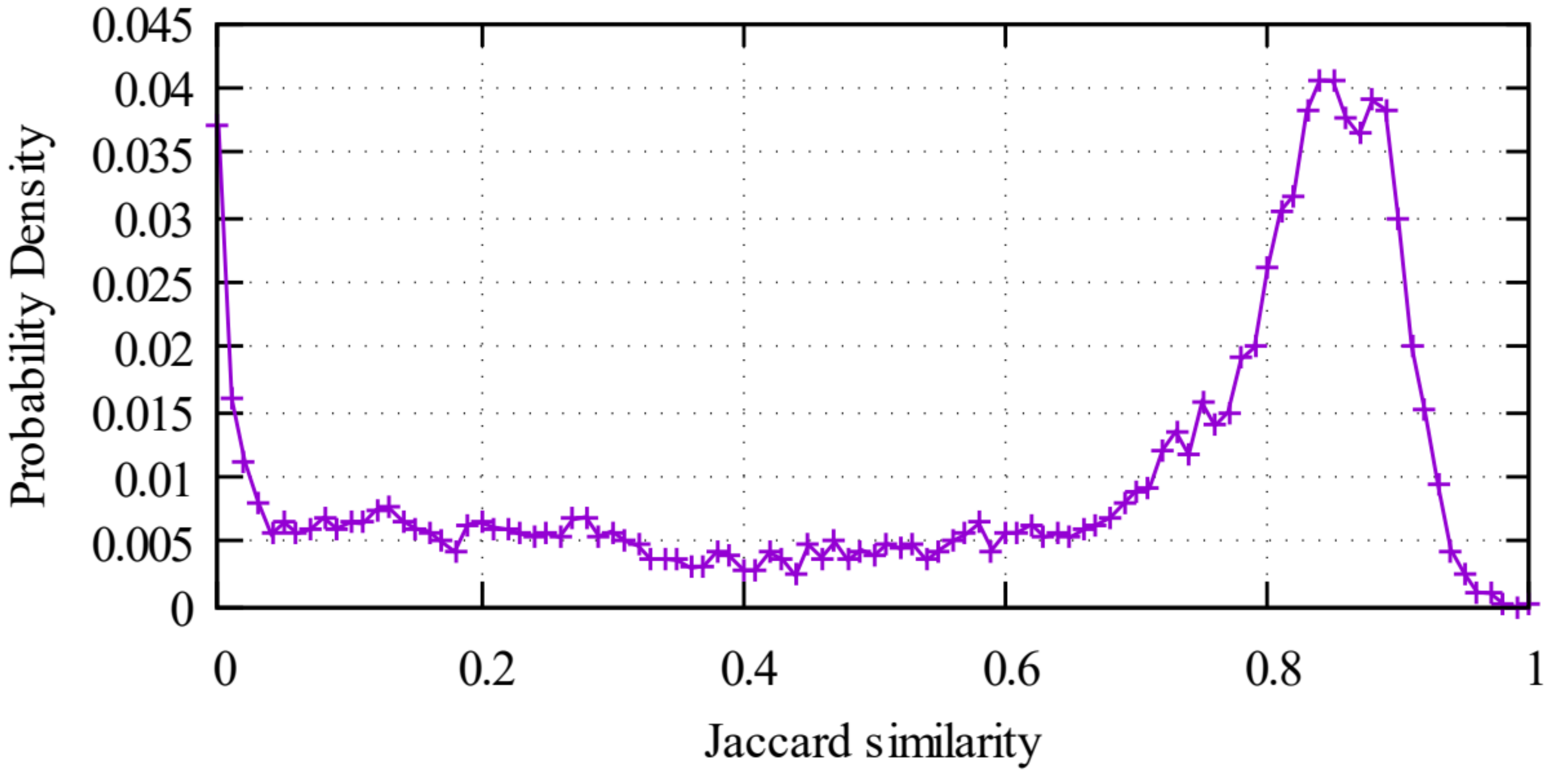
Measuring IP-device Mapping Dynamics

Device Types



Measuring IP-device Mapping Dynamics

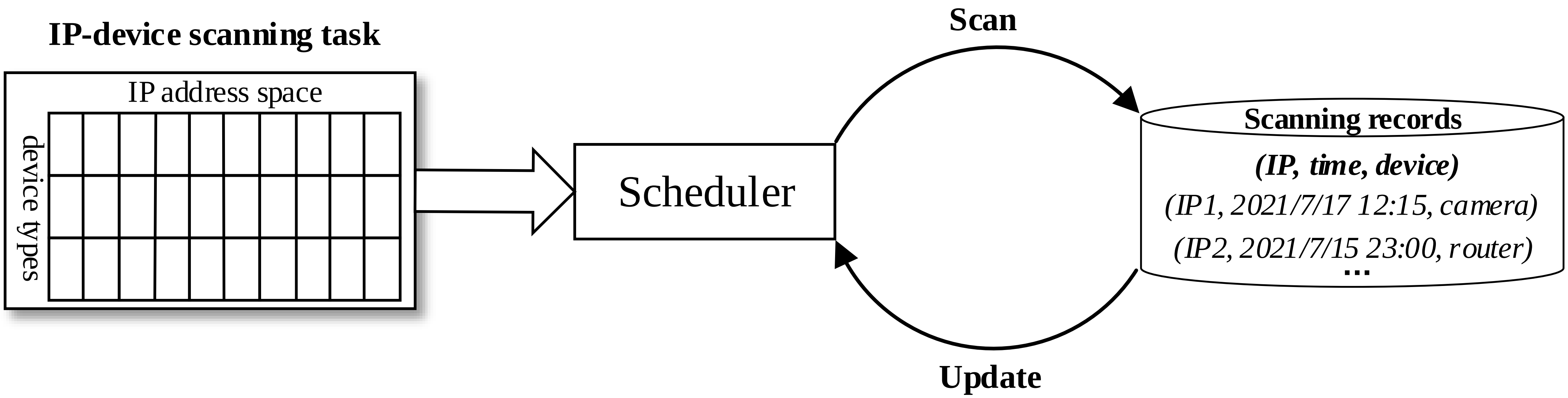
IP Pools



Outline

- Background and Problem Description
- Understanding IP-device Mapping Dynamics
- System Design
- Evaluation
- Discussion
- Summary

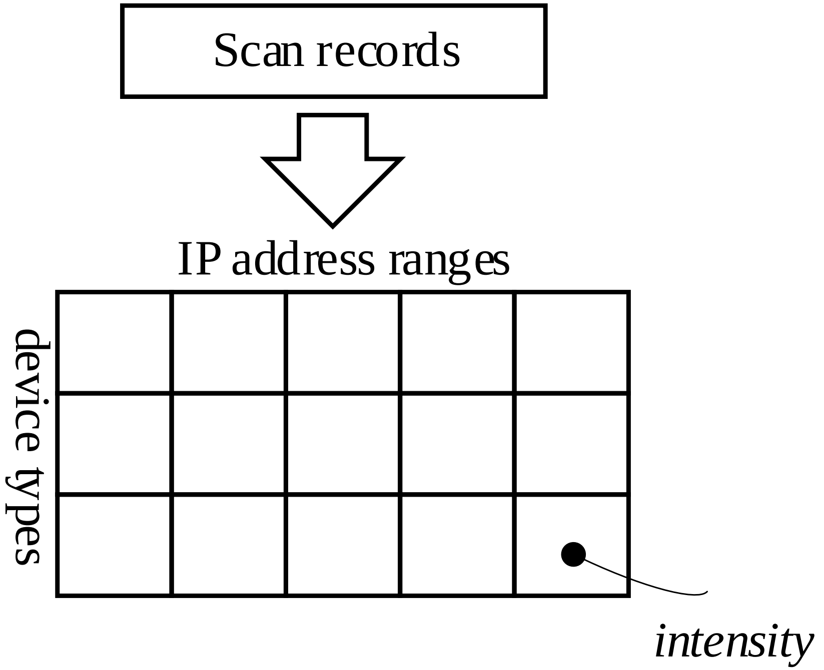
System Design



Scheduler

Basic idea:

Scan the IP addresses most likely to have IP-device mapping mutations



P(last scan time, last scan result, intensity) = 0.7223…

Scheduler

Basic idea:

Scan the IP addresses most likely to have IP-device mapping mutations

Algorithm 1: Scanning using Online Learning Strategy

Input: scanning task set $A = \{\text{IP addresses}\} \times \{\text{device types}\}$
Output: scanning records
initialization:
$\lambda(d, r, t) = y(d, r, t) = n(d, r, t) = 0$;
$S_t(a) = (t_0, d_0)$ for each a;
while *True* **do**
  $\lambda(d, r, t) = \frac{y(d,r,t)+1}{y(d,r,t)+n(d,r,t)+1}$;
  scan $\pi(S_t)$ and get a set of 2-gram scanning records E;
  for *each* $[a, < (t_1, d_1), (t_2, d_2) >]$ *in* $[\pi(S_t), E]$ **do**
    if $d_1 \neq d_2$ **then**
      $y(d_1, r_a, t)$ += $\frac{|T(\lambda(d,r_a,t)) \cap (t_1,t_2]|}{t_2 - t_1}$;
    else
      $n(d_1, r_a, t)$ += $\frac{|T(\lambda(d,r_a,t)) \cap (t_1,t_2]|}{|T(\lambda(d,r_a,t))|}$;
    end
    update $S_t(a) = (t_2, d_2)$;
  end
end

Scheduler

Improvement:

Stage1: collect useful information

Stage2: scanning

Algorithm 2: Scanning using Batch Learning Strategy

Input: scanning task set $A = \{\text{IP addresses}\} \times \{\text{device types}\}$
Output: scanning records
initialization:
$\lambda(d, r, t) = y(d, r, t) = n(d, r, t) = 0$;
$S_t(a) = (t_0, d_0)$ for each a;
—**Stage 1**: batch learning
while *True* **do**
  perform sequential scanning and get $< (t_1, d_1), (t_2, d_2) >$;
  if $d_1 \neq d_2$ **then**
    $y(d_1, r_a, t) \mathrel{+}= \frac{|T(\lambda(d, r_a, t)) \cap (t_1, t_2]|}{t_2 - t_1}$;
  else
    $n(d_1, r_a, t) \mathrel{+}= \frac{|T(\lambda(d, r_a, t)) \cap (t_1, t_2]|}{|T(\lambda(d, r_a, t))|}$;
  end
  update $S_t(a) = (t_2, d_2)$;
end
$\lambda(d, r, t) = \frac{y(d,r,t)+1}{y(d,r,t)+n(d,r,t)+1}$;
—**Stage 2**: delayed scanning
while *True* **do**
  calculate $P(a|s)$ for each address a using (4);
  scan address a' that maximize $P(a'|s)$;
end

Outline

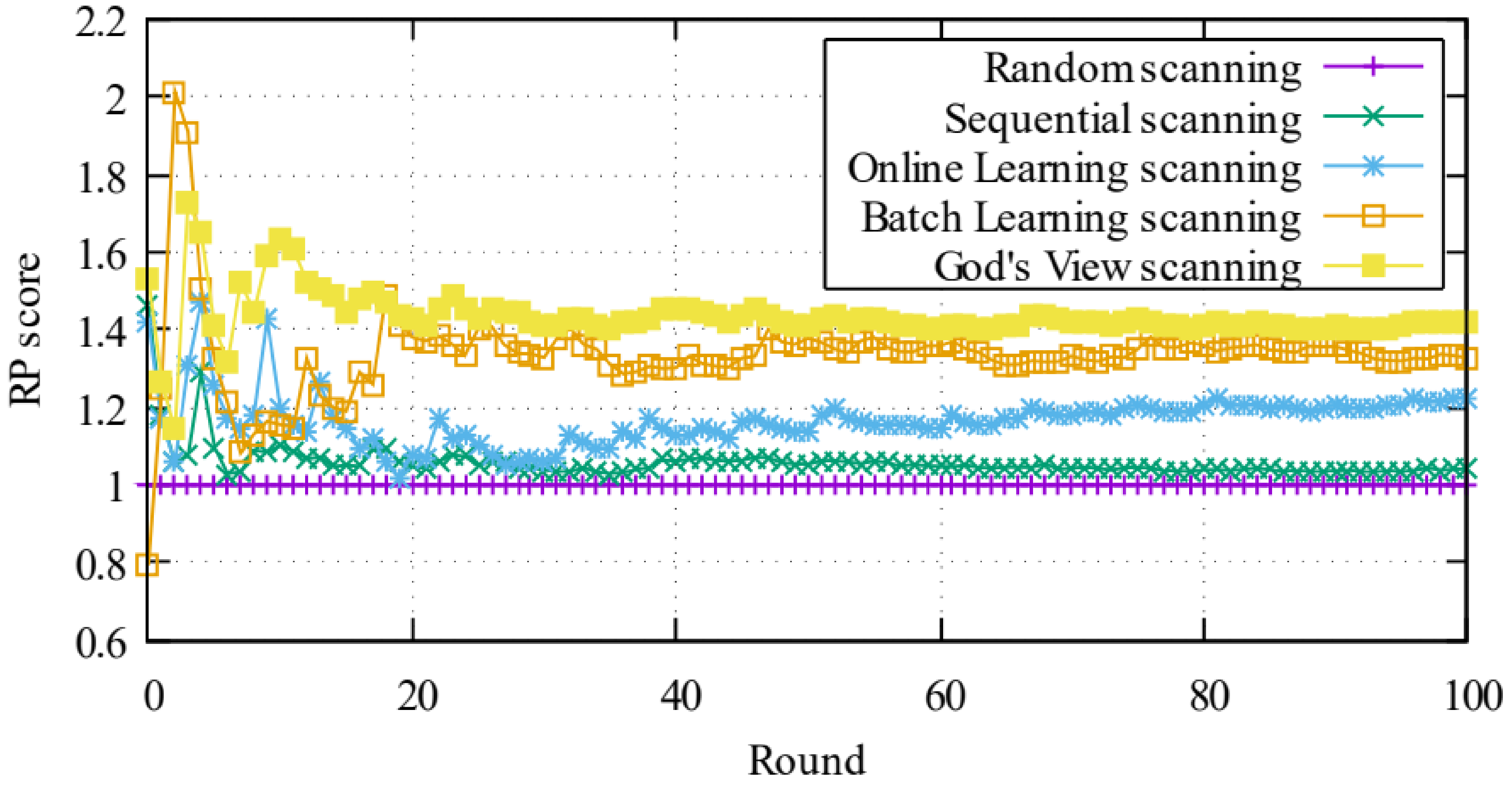
- Background and Problem Description
- Understanding IP-device Mapping Dynamics
- System Design
- Evaluation
- Discussion
- Summary

Parameter Settings

DEFAULT PARAMETER SETTINGS.

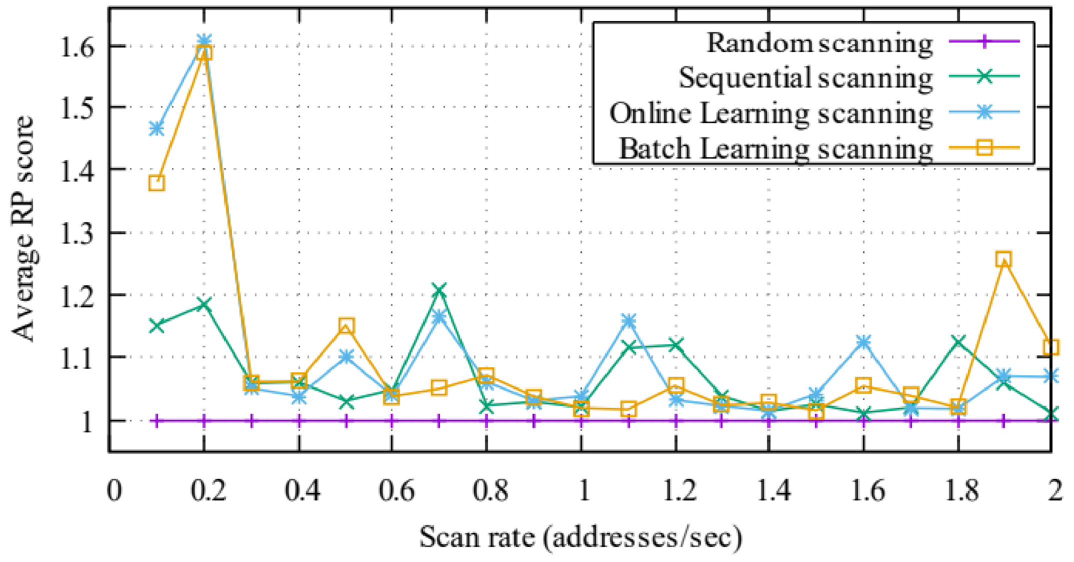
Parameters Name	Value
Total number of IP addresses	8,192 (32 class C networks)
Total number of IP pools	10
Scan rate	0.5 IP addresses per second
The proportion of devices to addresses	0.8
Number of device types	20
Number of scanning rounds	100
λ in Scenario A	$1/\lambda \sim$ U(0h,24h)
Address change time in Scenario B	$t \sim$ U(0h,24h)

Scanning Performance using Different Strategies

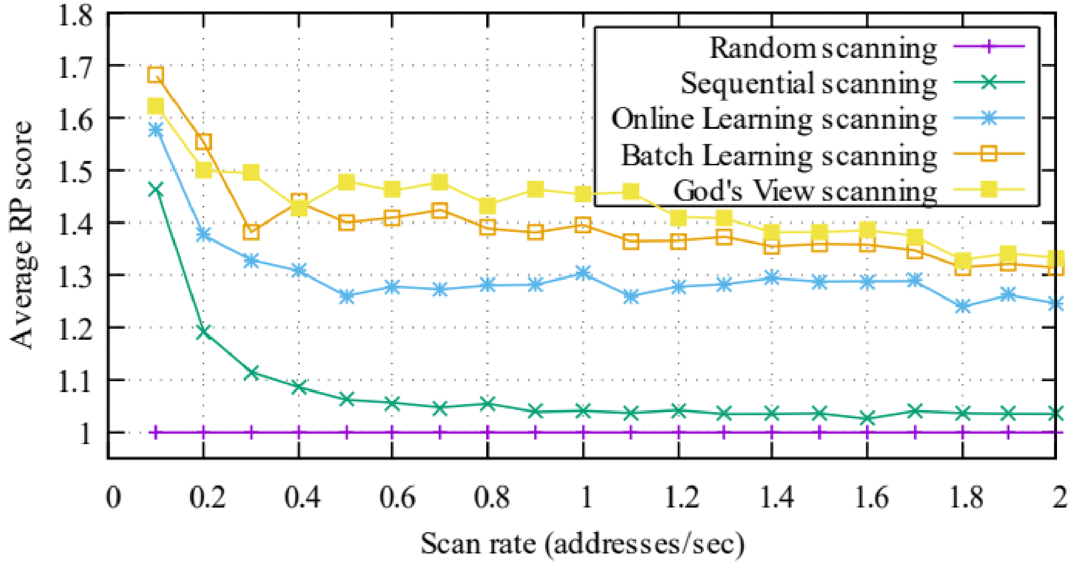


Performance Sensitivity

Scan rate



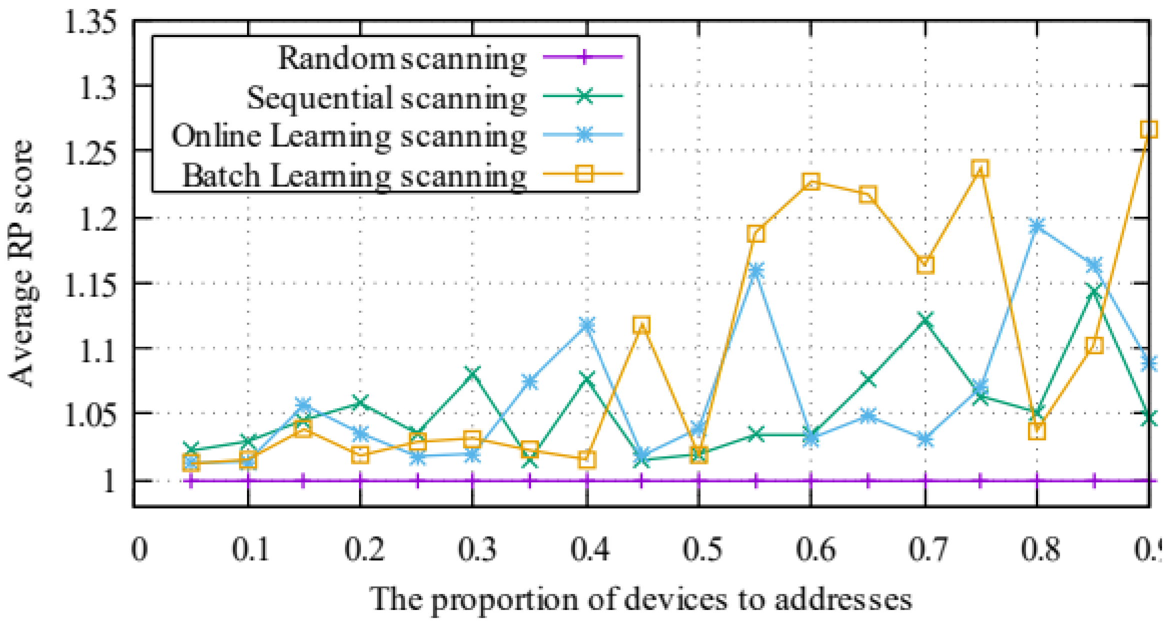
(a) Scenario A



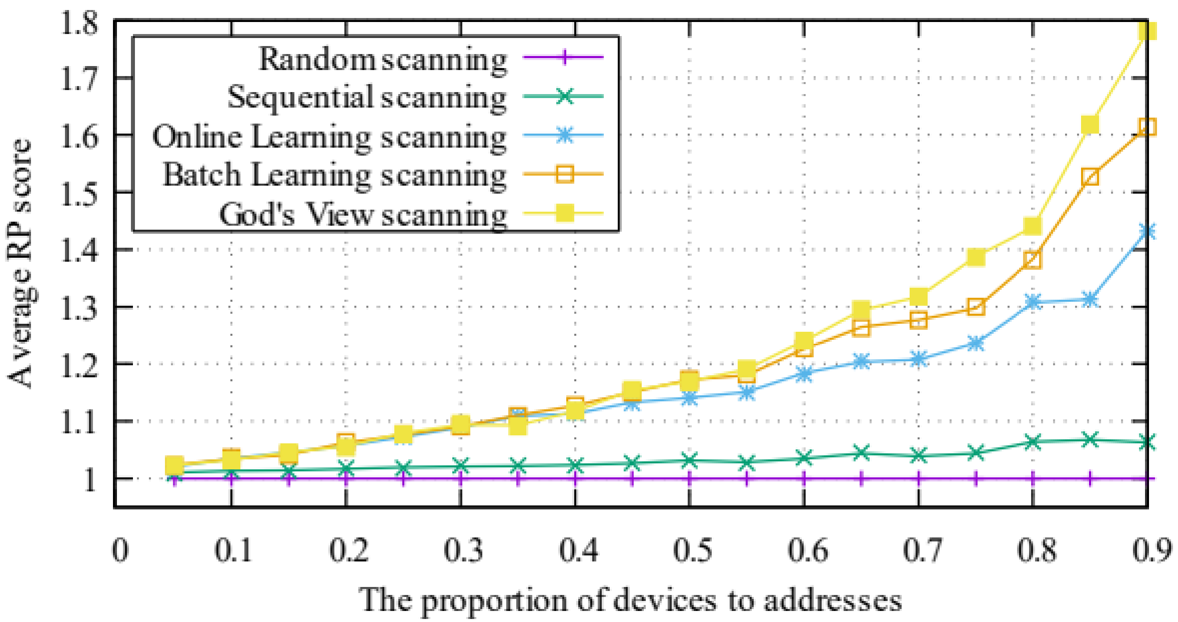
(b) Scenario B

Performance Sensitivity

The Proportion of Devices to IP Addresses

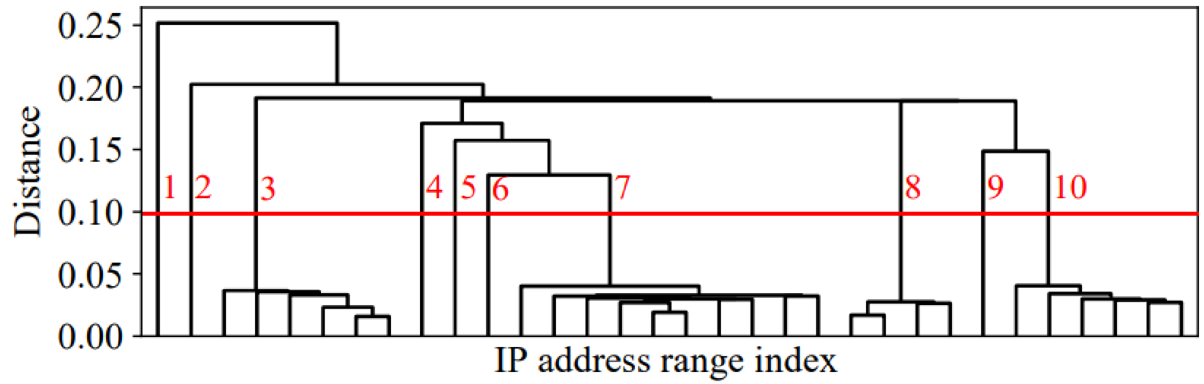


(a) Scenario A

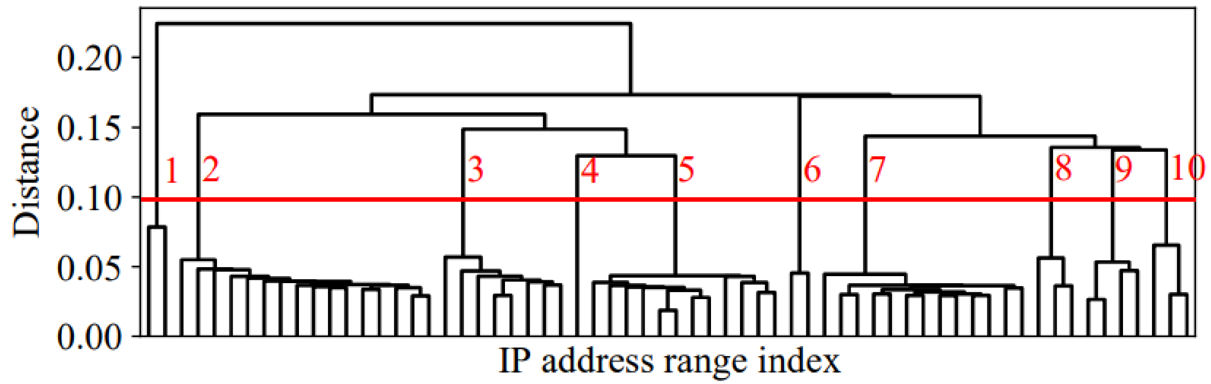


(b) Scenario B

IP Pool Estimation



(a) Setting 1: the number of IP address ranges=32, scan rate=0.5



(b) Setting 2: the number of IP address ranges=64, scan rate=0.25

Outline

- Background and Problem Description
- Understanding IP-device Mapping Dynamics
- System Design
- Evaluation
- Discussion
- Summary

Discussion

- One IP addresses with multiple devices
- Simulation vs. Real-world
- Calculation trade-off

Outline

- Background and Problem Description
- Understanding IP-device Mapping Dynamics
- System Design
- Evaluation
- Discussion
- Summary

Summary

- We perform measurements based on large-scale real-world IoT scanning records by scanning the entire IPv4 space for about 40 days, and quantify the IP-device mapping dynamics. The results reveal that both the IoT device types and IP address pools affect the dynamics.

- We land reinforcement learning onto a system capable of smartly scanning IoT devices. The system can encourage scans to networks with more dynamic IP-device mapping while impeding scans to those with less dynamic mapping.

- Through extensive experiments, we demonstrate that our system could generally capture more IP-device mapping mutations than random and sequential scanning.