Hestia: Simple Least Privilege Network Access Policies for Smart Homes

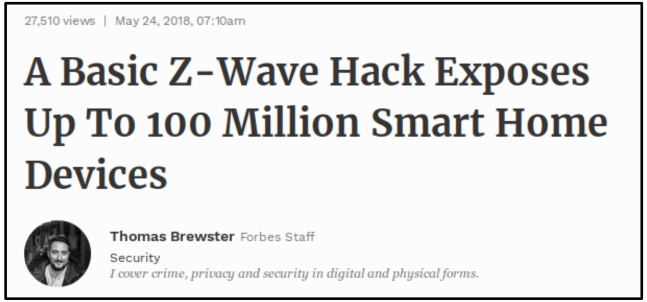
*In the Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (**WiSec**), May 2019*

Sanket Goutam

Advised by: Dr. William Enck & Dr. Brad Reaves

Managing a smart home is hard!

- ❑ Smart home devices are not secure by construction
  - ❑ Off-the-shelf IoT devices are often found to be insecure, and are difficult to patch
- ❑ Heterogeneity of devices, both hardware and software, makes standardization infeasible



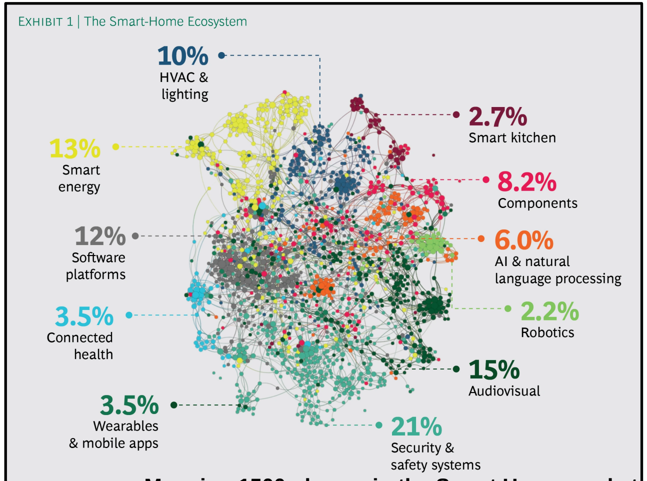
27,510 views | May 24, 2018, 07:10am

A Basic Z-Wave Hack Exposes Up To 100 Million Smart Home Devices

Thomas Brewster Forbes Staff

Security

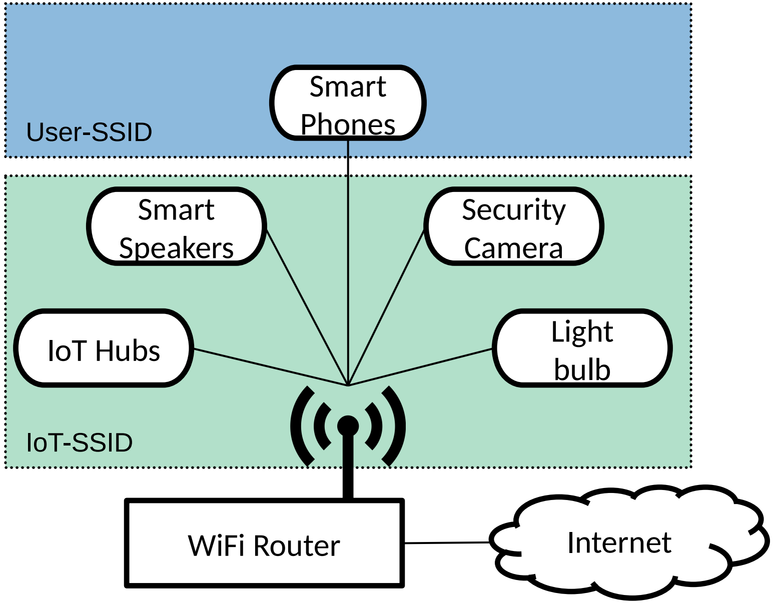
I cover crime, privacy and security in digital and physical forms.



Sources: Quid; BCG.

Mapping 1500 players in the Smart Home market

Typical smart home setup



- Average users simply deploy all devices to the same network

Best Practices, Security Trends/Attacks

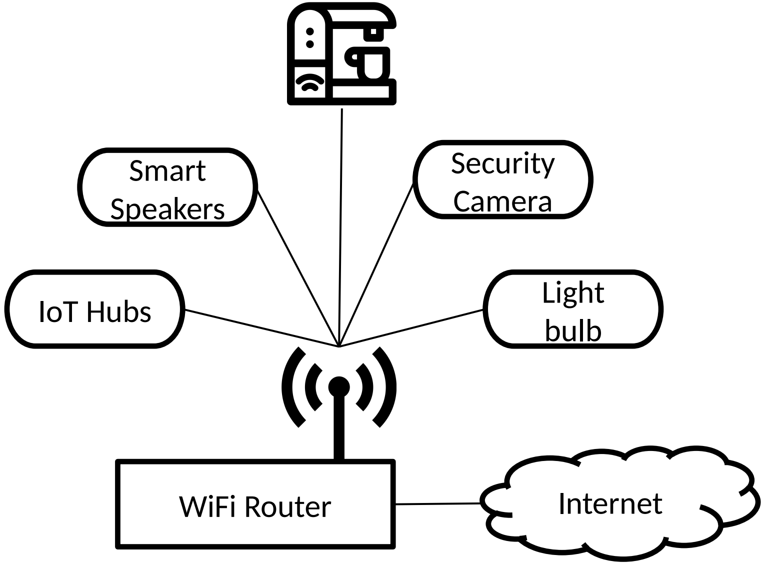
Network Segmentation: A Key Measure for IoT Security

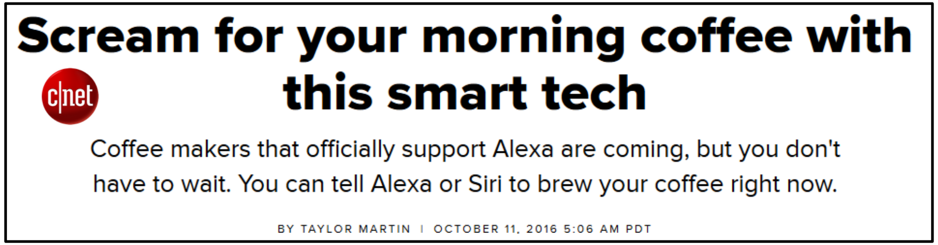
Arctic Wolf Networks

- Current best practices suggest isolating IoT devices from user devices

Are these traditional network setups sufficient?

Case Study – Mr. Coffee





Scream for your morning coffee with this smart tech

Coffee makers that officially support Alexa are coming, but you don't have to wait. You can tell Alexa or Siri to brew your coffee right now.

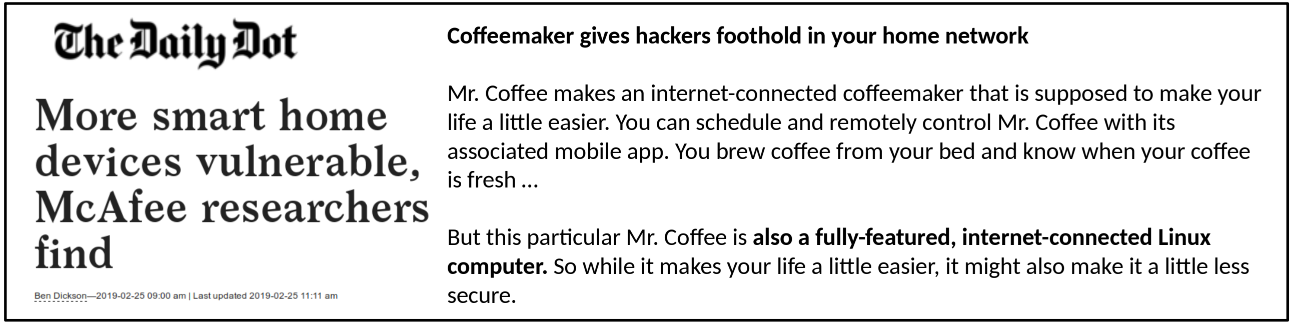
BY TAYLOR MARTIN | OCTOBER 11, 2016 5:06 AM PDT

Let’s take a closer look …

Case Study – Mr. Coffee

- Purpose – Brew coffee based on a schedule or upon being remotely triggered by the user
- Requirements – Connect to home Wi-Fi network, receive commands from a supported WeMo platform

February 25, 2019



The Daily Dot

More smart home devices vulnerable, McAfee researchers find

Ben Dickson—2019-02-25 09:00 am | Last updated 2019-02-25 11:11 am

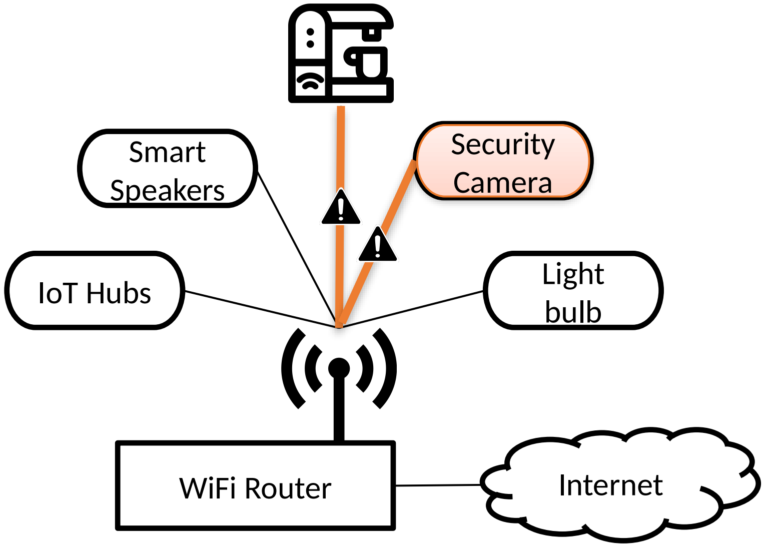
Coffeemaker gives hackers foothold in your home network

Mr. Coffee makes an internet-connected coffeemaker that is supposed to make your life a little easier. You can schedule and remotely control Mr. Coffee with its associated mobile app. You brew coffee from your bed and know when your coffee is fresh …

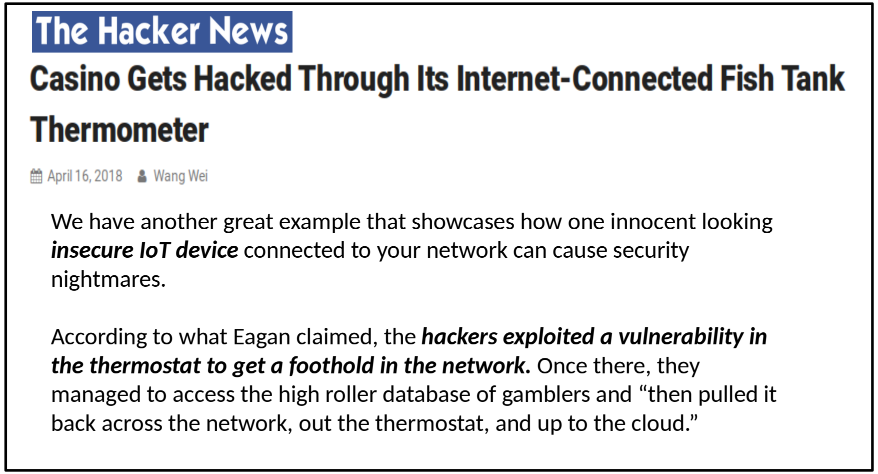
But this particular Mr. Coffee is **also a fully-featured, internet-connected Linux computer.** So while it makes your life a little easier, it might also make it a little less secure.

- Vulnerability – Commands are transferred in plaintext, doesn't validate source before execution
- Threat vector – Shell access, excellent covert pivot point for lateral movement

Attack on Smart Home devices...



Rube Goldberg attack



The Hacker News

Casino Gets Hacked Through Its Internet-Connected Fish Tank Thermometer

April 16, 2018 Wang Wei

We have another great example that showcases how one innocent looking ***insecure IoT device*** connected to your network can cause security nightmares.

According to what Eagan claimed, the ***hackers exploited a vulnerability in the thermostat to get a foothold in the network.*** Once there, they managed to access the high roller database of gamblers and “then pulled it back across the network, out the thermostat, and up to the cloud.”

During deployment users put the same level of trust on all connected devices.

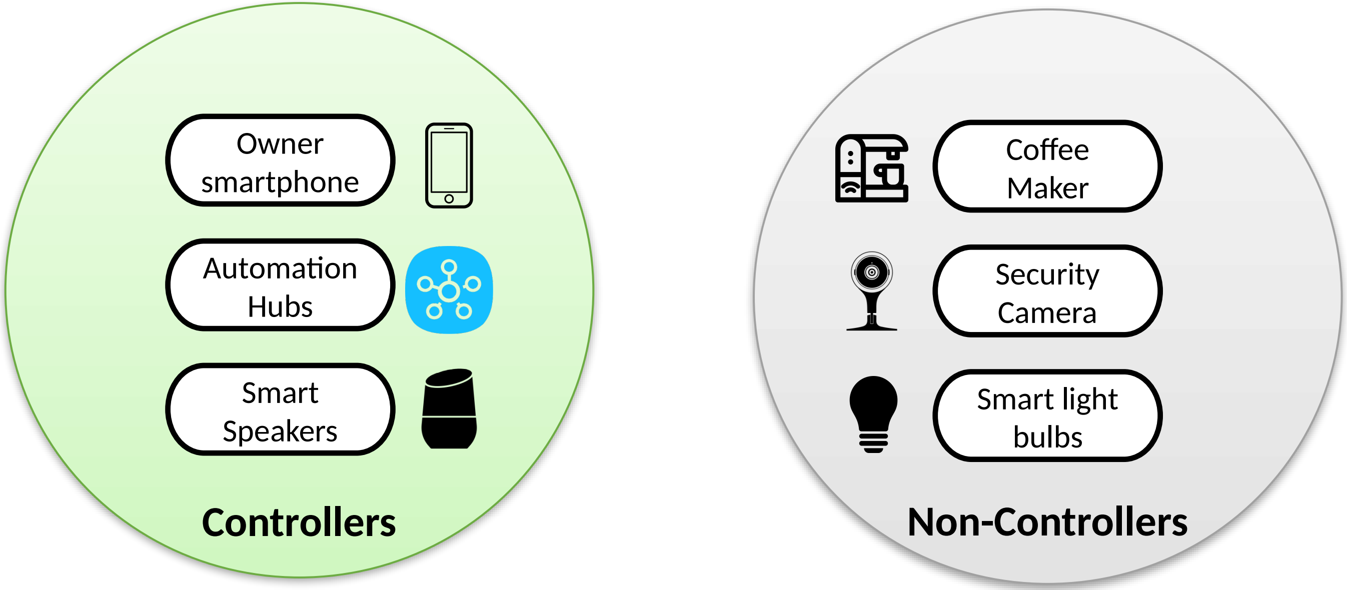
Research Statement

- Most smart home devices exhibit limited and predictable communication paths on a local network
  - Access services on the Internet
  - Receive control commands from automation hubs, smartphones
- IoT devices are designed for specific purposes
  - Thermometer reports temperature readings to a web service or users’ smartphone
  - Coffee Maker brews coffee on receiving a command from a WeMo automation hub
- Network communication paths should justify their purpose

Can we simplify this to a least privilege policy?

Device categories

- ❑ We define a dichotomy of smart home devices



Owner smartphone

Automation Hubs

Smart Speakers

Controllers

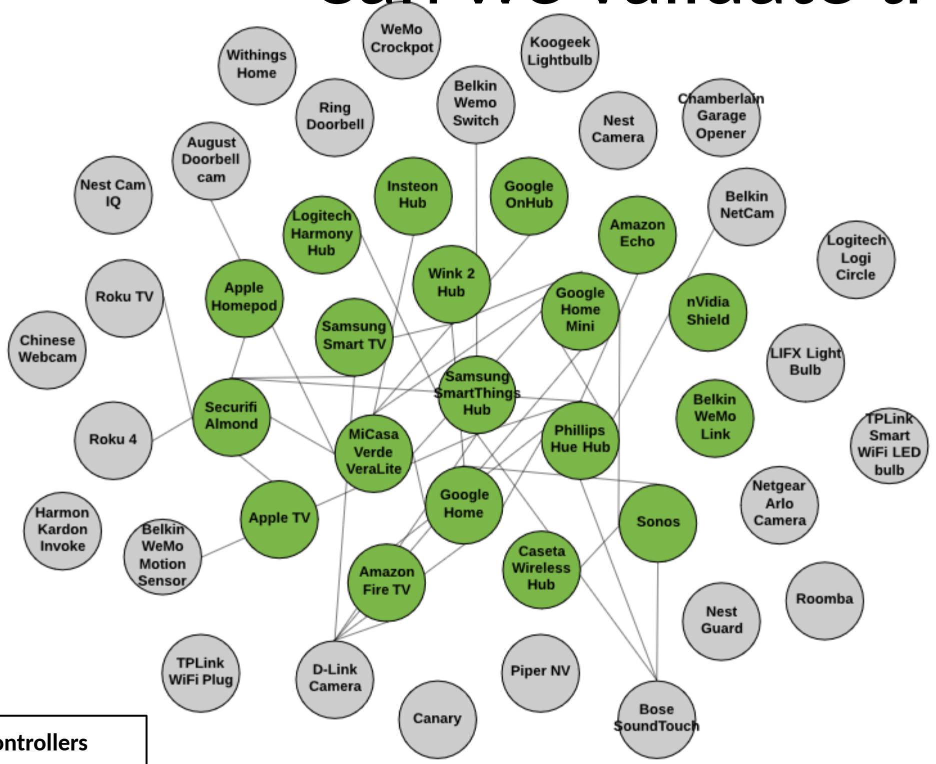
Coffee Maker

Security Camera

Smart light bulbs

Non-Controllers

Can we validate this hypothesis?



🟩 Controllers

⬜ Non-Controllers

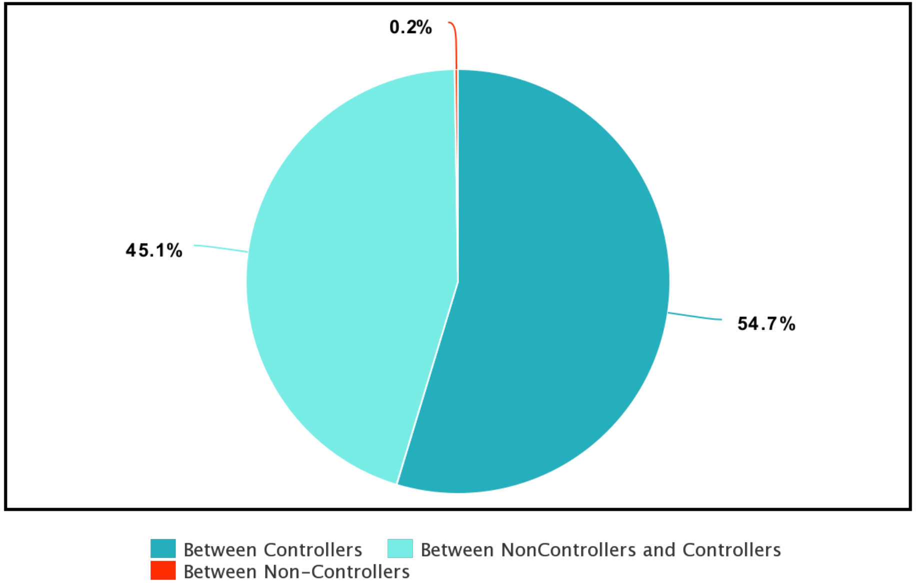
- ❑ YourThings data set of smart home deployment
  - ❑ Network captures for 10 days
  - ❑ 46 labeled smart home devices

Categorizing the YourThings devices

Distinguishing Feature	No. of Devices	Category
None	26	Non-Controllers
Voice Assistant	10	Controllers
Remote Control Hub	9	Controllers
Home Router	1	Controllers

Non-Controllers do not interact with each other

Findings



D2D interactions for all 10 days

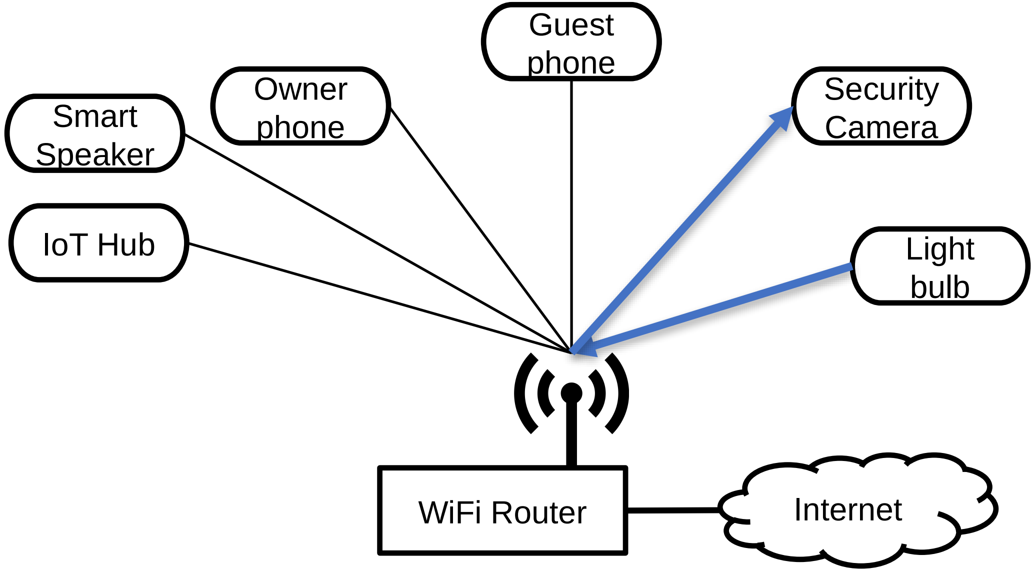
- ❑ Found 1 instance of deviation from our hypothesis
  - ❑ 2 (out of ~15.5 million) packets were exchanged between two non-controllers
  - ❑ D-Link Camera requesting device details of a Belkin Netcam (UPnP discovery)
  - ❑ While innocuous, similar to known UPnP injection attacks on Belkin Netcam
- ❑ Least privilege policy – Non-Controller device should only be able to interact with a Controller device

So, let's implement it!

Research Challenges

- ❑ **R1 :** Existing network access control mechanisms do not mediate between devices on the same LAN.
  - ❑

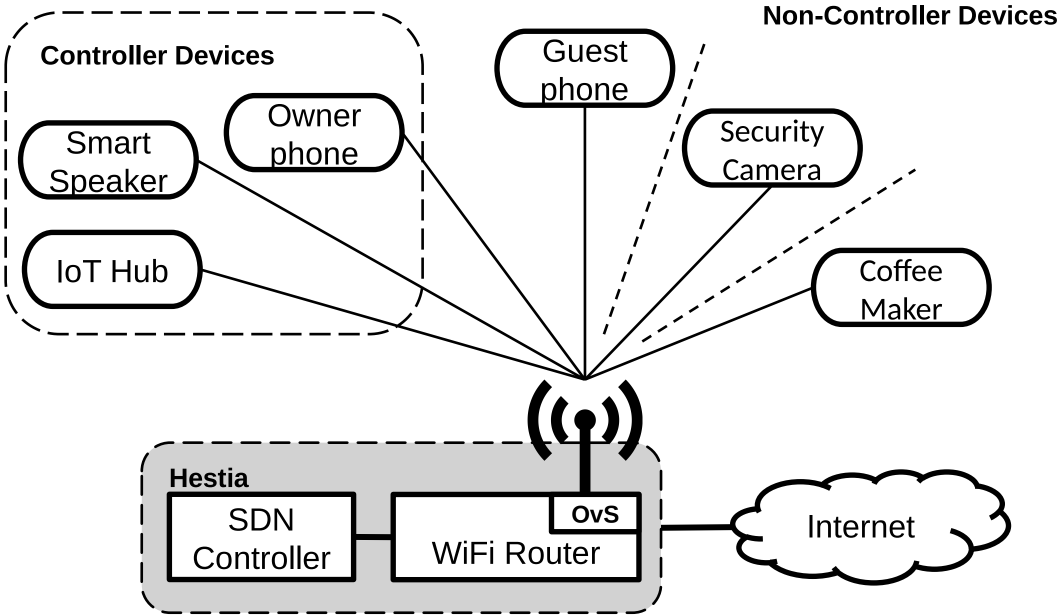
Standard Access Point

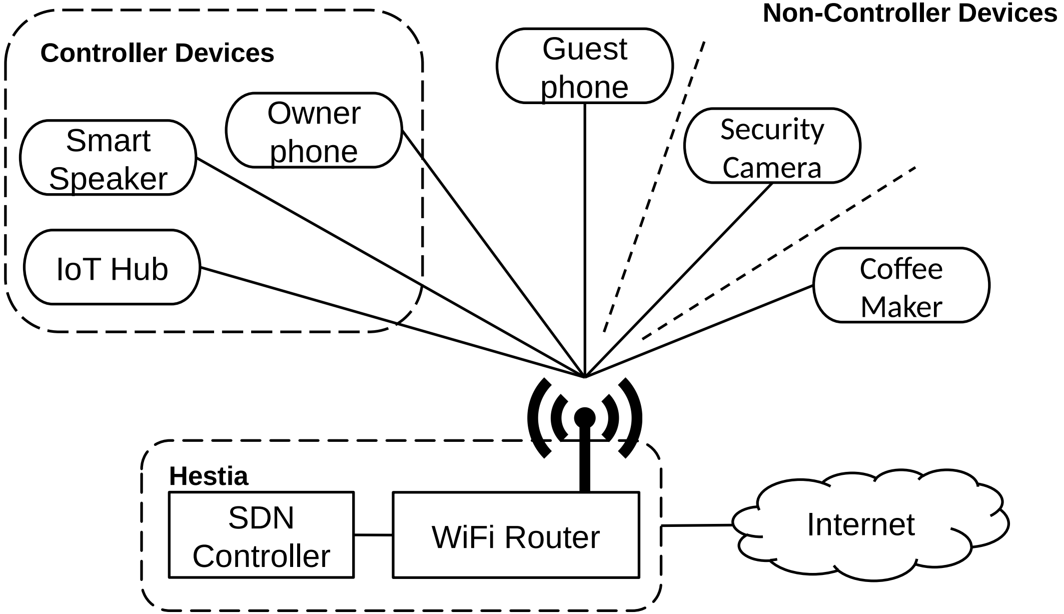


- ❑ WiFi interface acts as a bridge
  - ❑ Packets between wireless clients are forwarded directly, without going through the entire network stack

Hestia

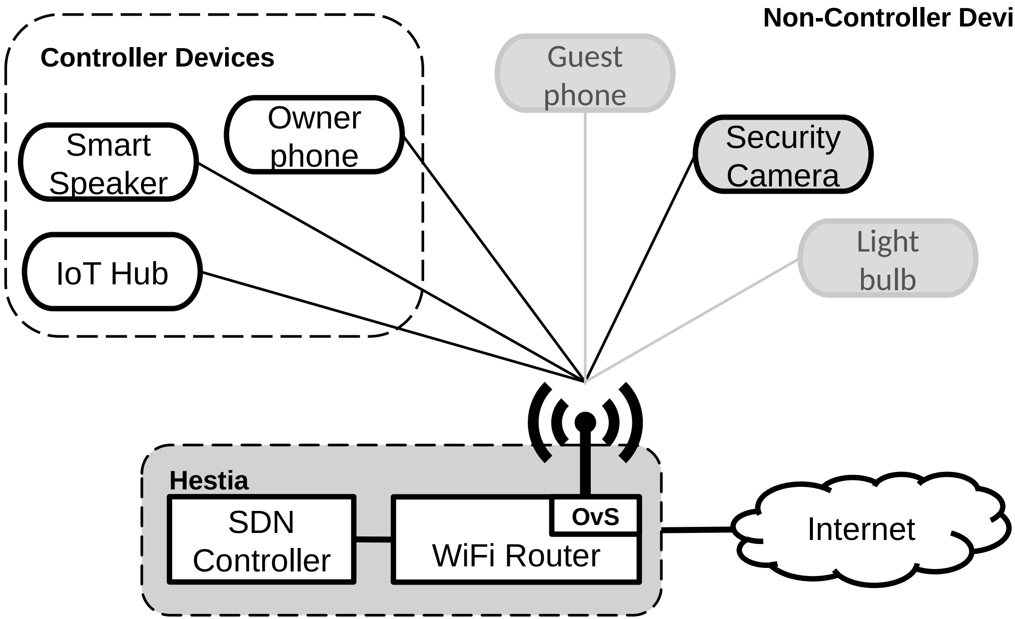
- ❑ Modified router firmware that includes OvS (Open vSwitch) soft switch
  - ❑ Relays packets to the SDN controller
- ❑ Allows us to define granular policies to control device communication
  - ❑ All devices can connect to Internet
  - ❑ Non-Controllers can interact with Controller devices only




Controller Devices
Smart Speaker
Owner phone
IoT Hub
Guest phone
Non-Controller Devices
Security Camera
Coffee Maker
Hestia
SDN Controller
WiFi Router
Internet

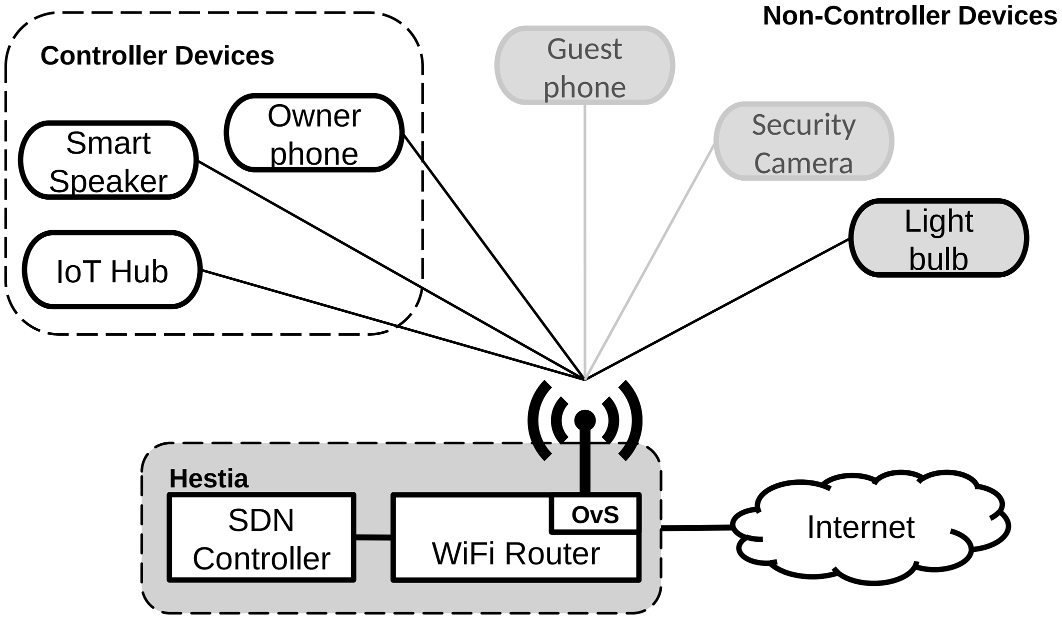
Hestia

- ❑ Modified router firmware that includes OvS (Open vSwitch) soft switch
  - ❑ Relays packets to the SDN controller
- ❑ Allows us to define granular policies to control device communication
  - ❑ All devices can connect to Internet
  - ❑ Non-Controllers can interact with Controller devices only

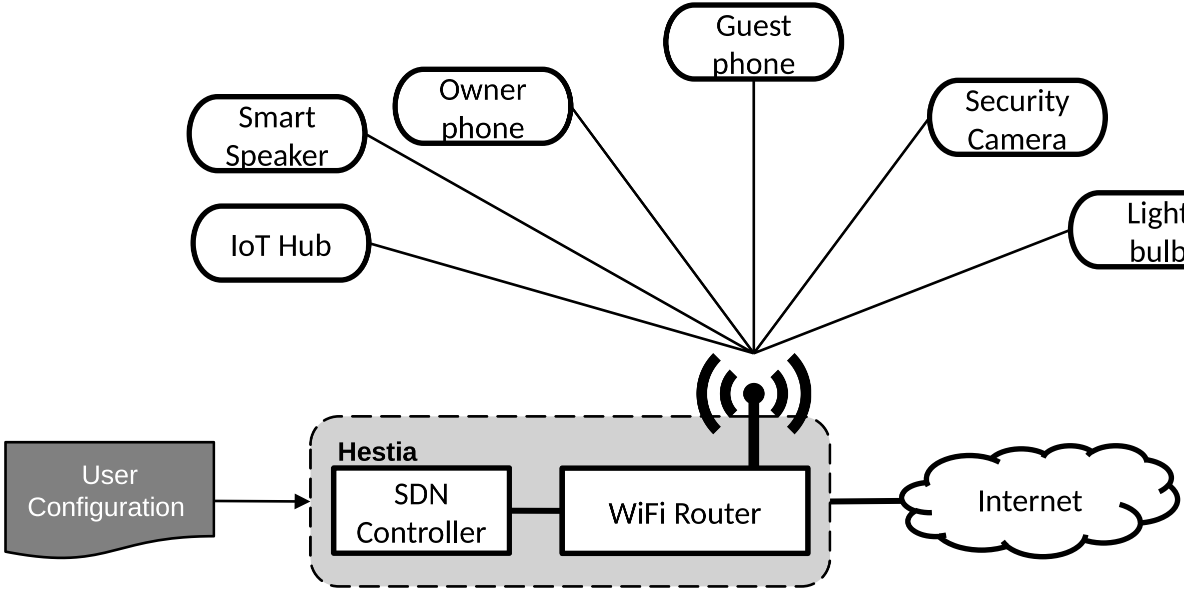


Hestia

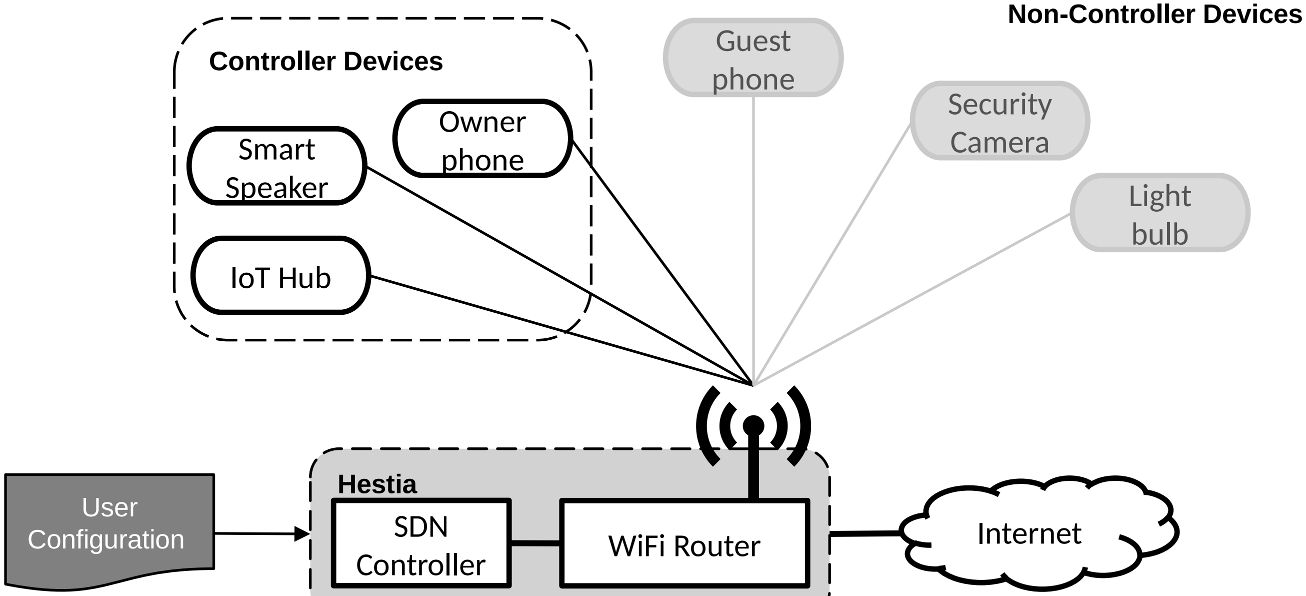
- ❑ Modified router firmware that includes OvS (Open vSwitch) soft switch
  - ❑ Relays packets to the SDN controller
- ❑ Allows us to define granular policies to control device communication
  - ❑ All devices can connect to Internet
  - ❑ Non-Controllers can interact with Controller devices only



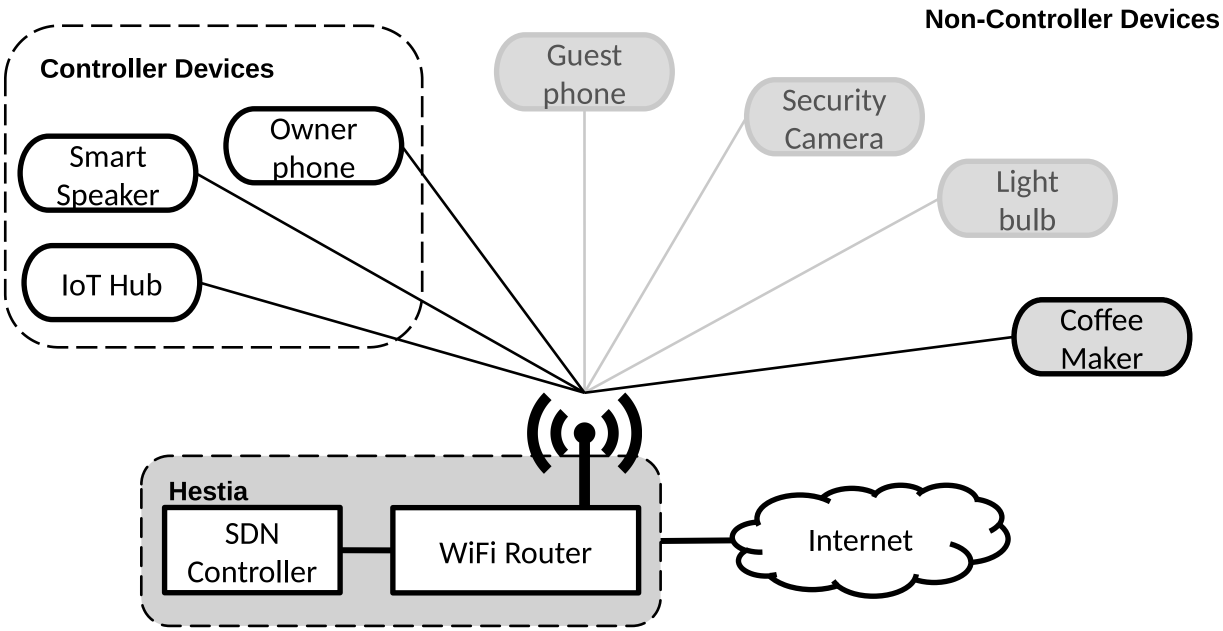
Identifying controller devices



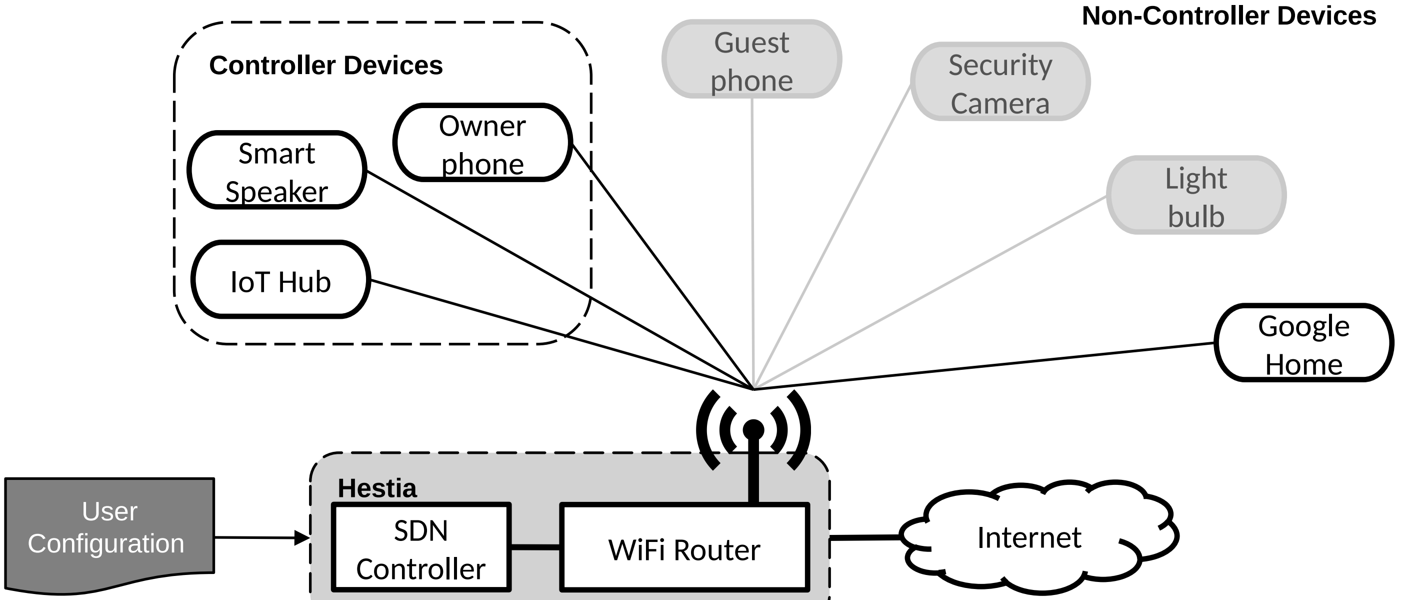
Identifying controller devices



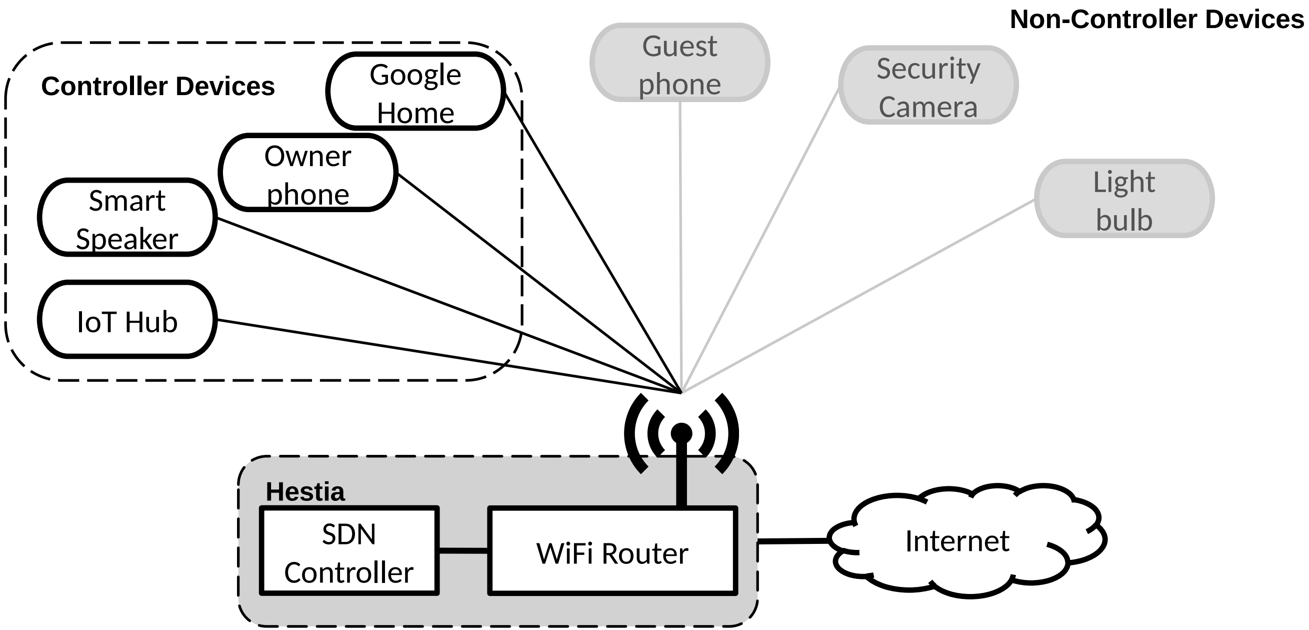
Connecting a new device to Hestia



Connecting a new device to Hestia



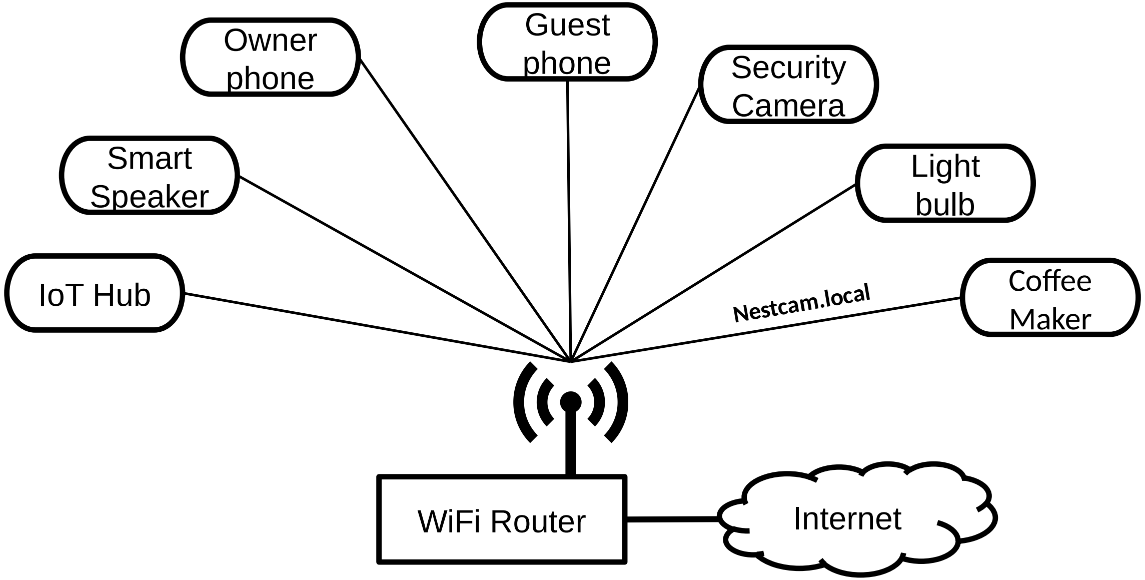
Connecting a new device to Hestia



Research Challenges

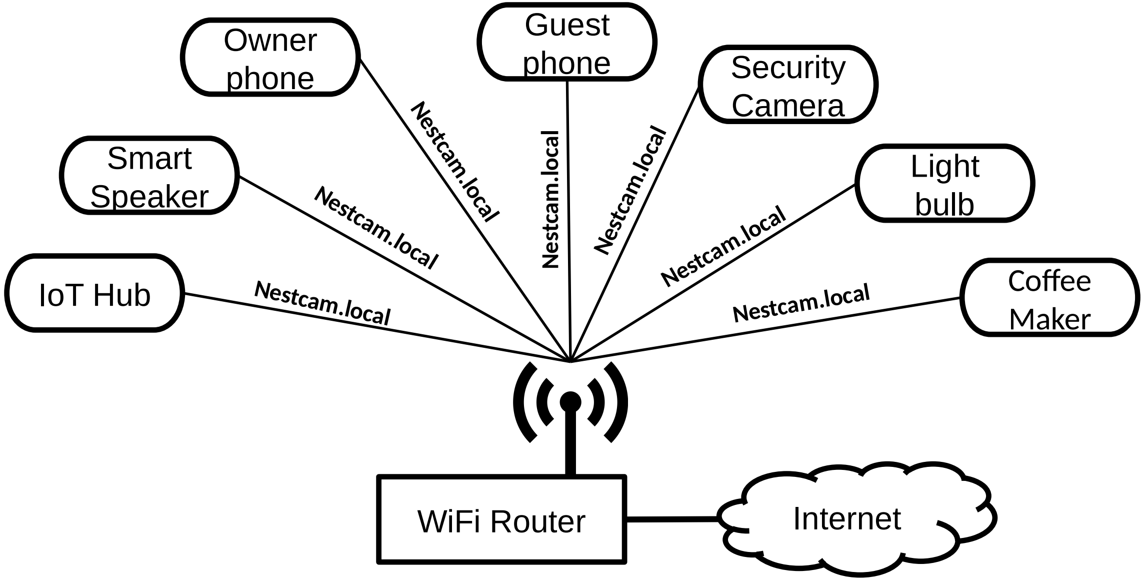
- **R1 :** Existing network access control mechanisms do not mediate between devices on the same LAN.
  - Traffic between local devices stays within the WiFi network, making firewall policies and ACLs ineffective
- **R2 :** Handling multicast discovery packets
  -

Standard Access Point



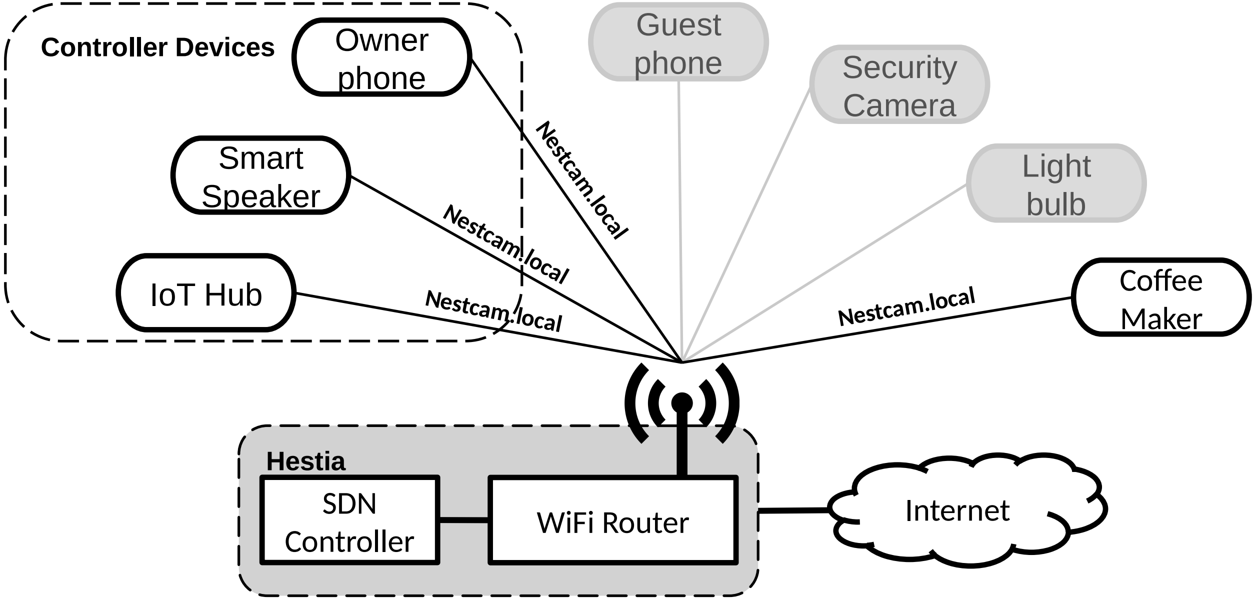
- ❑ Multicast packets are duplicated by the AP and broadcasted to all connected clients
  - ❑ Can be used for network reconnaissance and discovery protocol based attacks

Standard Access Point



- ❑ Multicast packets are duplicated by the AP and broadcasted to all connected clients
  - ❑ Can be used for network reconnaissance and discovery protocol based attacks

Selective Device Discovery



- ❑ Hestia provides a protocol agnostic way of handling discovery packets
- ❑ Non-Controller devices are only discoverable by the Controller devices

Deploying Hestia in a smart home

- ❑ Hestia is designed to replace the standard WiFi access points in a smart home
- ❑ Deployed on a commodity home router
  - ❑ Lightweight SDN app in python using RYU framework

Can Hestia effectively replace a standard router without any performance overhead?

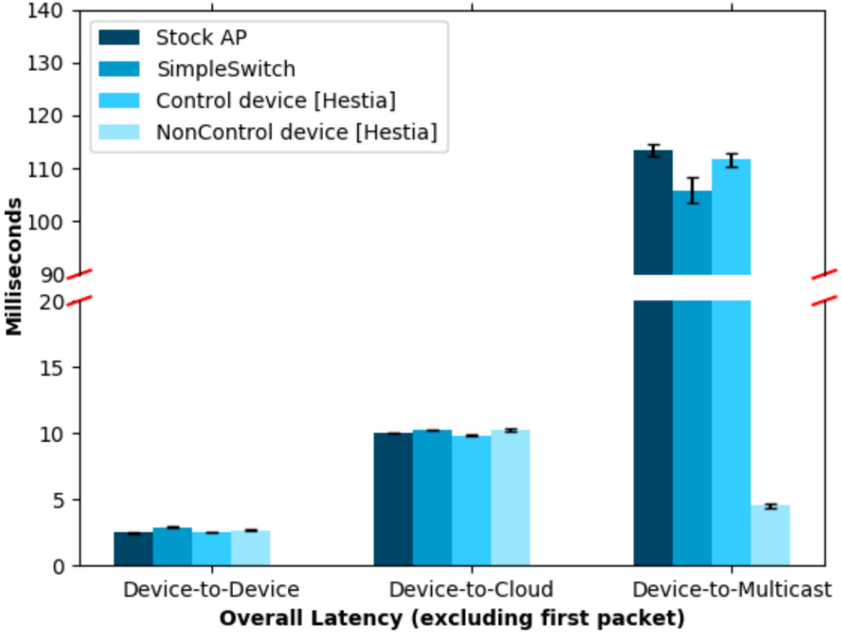
Network Performance Evaluation

- We explore a total of 12 different experimental conditions
  - Impacts on all communication types
  - Impacts due to device categorization
- We measure “three” key variables
  - First packet latency
    - SDN systems treat the first differently to make a routing decision
  - Average (without first packet) latency
  -

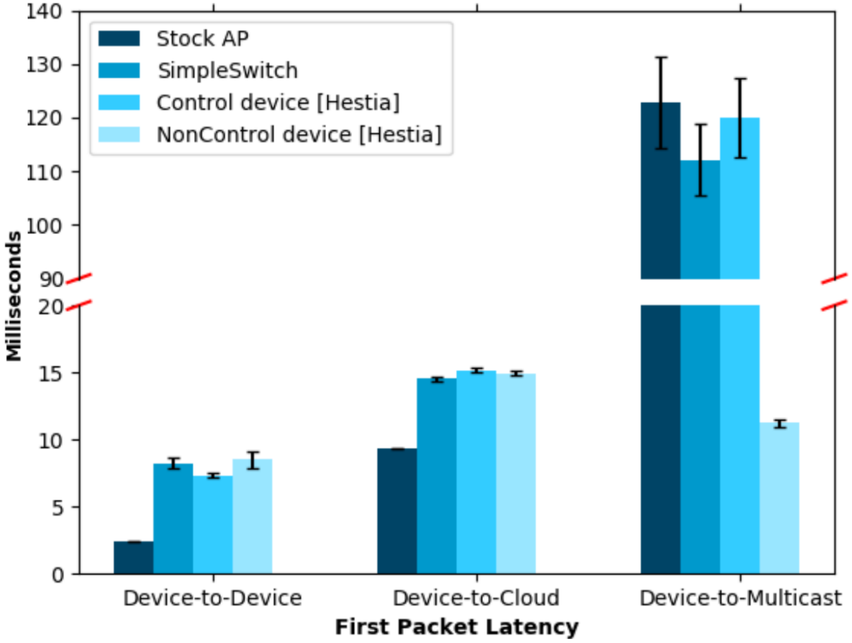
Experimental Setup

- Desktop as the SDN controller
- Macbook Air acting as the client generating traffic
- At least 7 additional devices connected as controllers
    - Including smartphones, tablets, eBook readers, etc.
- Developed a latency measurement tool for multicast communication, as most available tools (including ping) do not support multicast
-

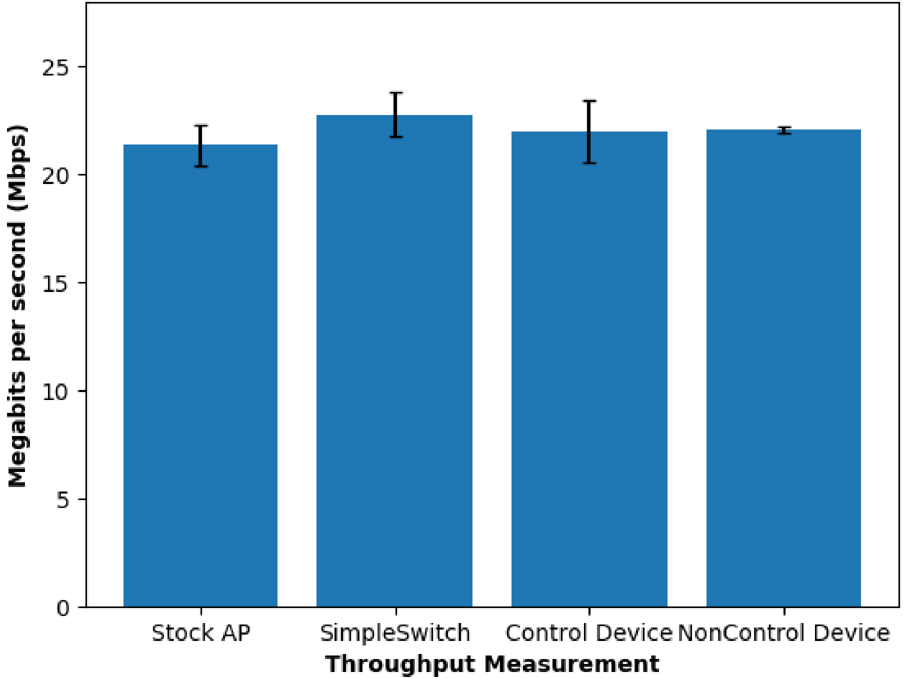
Average latency is largely constant



First packet latency is slightly higher for SDN systems



Hestia does not negatively impact throughput



Takeaway

- ❑ Current network deployments and access control policies are not ready for the smart home ecosystem
- ❑ We provide a practical approach to this problem, in the form of a least privilege network access policy
- ❑

Questions?

BACKUP SLIDES

YourThings data set

- ❑ Recorded device interactions in 5 minute intervals over 10 days
- ❑ Devices, and their network configurations varied from day-to-day
- ❑ Created unique src-dst mappings on a per-day basis
  - ❑ To understand which devices interact across different setups
- ❑ Total of 426 instances of device-to-device interactions
  - ❑ Single exception: 2 packets exchanged between non-controllers