Bart Busschots
www.bartb.ie

- Host, Editor, and Executive Producer of the Let's Talk Apple & Let's Talk Photography podcasts

- Sysadmin (mostly Linux) at Maynooth University

Making a Place for the IoT in Your Home (and How the Internet Works)

Contents

- What is the Internet of Things (IoT)?
- What's the problem with IoT devices?
- How the Internet Works (the Executive Summary)
- IP Subnets & the Difference between Routers, Switches & Wireless Access Points
- An approach for 'safely' adding IoT devices to you home network

What is the IoT?

- IoT == Internet of Things
- Like Web 2.0 a decade ago, the IoT is a buzzword without a robust definition
- The Internet of Things is generally considered to consist of internet connected appliances. Not traditional computers, but other things
- 'Smart' means internet connected
  'dumb' means not internet connected

Examples:

- Smart TVs
- Smart Lightbulbs
- Internet-connected Cameras
- Internet-connected home alarm systems

What's the Problem?

- All computer code is written by humans, and all humans make mistakes - bugs are INEVITABLE
- Most bugs are just annoying, but some create entry points for attackers - i.e. some bugs create VULNERABILITIES
- The longer a code-base has been around, the more time there has been to find and fix bugs, so the more stable and secure it is
- Conversely, new code-bases inevitably contain more bugs, and are less stable and less secure

What's the Problem?

- Good security is built with a product, not bolted on afterwards
- A security architecture should be an integral part of the initial design - it should guide the design of the code from the very first day
- No amount of rushed patches can ever make up for a poorly designed security architecture
- Retro-fitting a new security architecture into an existing product is like trying to put a modern LAN into a medieval castle - expensive and difficult!

What's the Problem?

- The IoT is a nascent market - there are no established market leaders yet - there is no Microsoft of the IoT
- The IoT market can be thought of as being in the 'land-grab' phase - everyone is racing to market with what they hope will be the coolest new features in the hope of locking people into their platform before someone else does
- In this kind of 'wild west' market, security is often given a lower priority than features - it is often an after-thought
- The end result - the IoT is a security mine-field!

How the Internet Works

The Executive Summary

It's ALL TCP/IP

- The internet, our home networks, university networks, corporate networks, they are all powered by TCP/IP
- TCP/IP allows streams of data to be sent between two (internet connected) computers anywhere in the world
- This is HARD - so TCP/IP breaks the problem into four manageable pieces (layers - each building on the one below)
- TCP/IP breaks data into small bite-sized chunks called packets

4 Layers

- Layer 1 - Link (Ethernet & MAC addresses)
  - Send a single packet within a single network
- Layer 2 - Internet (IP & IP addresses)
  - Get one packet across the internet
- Layer 3 - Transport (TCP, UPD & port numbers)
  - Get a stream of data across the internet
- Layer 4 - Application (HTTP, SMTP, IMAP, POP3, FTP etc.)
  - The protocols that drive the apps we humans use

Addressing

- Ethernet uses MAC addresses
- IP, TCP, UDP, and all the Application Layer protocols use IP addresses
- Humans use domain names (like www.bartb.ie)
- Who or what does the translations?

Addressing

- ARP (Address Resolution Protocol) translates between MAC and IP addresses within an Ethernet network
- DNS (Domain Name System) translates between domain names and IP addresses

The Fly in the Ointment

- ARP is a very old protocol, and lacks authentication - ARP is naive, and all packets are believed

- This can be exploited by a malicious device ON THE SAME ETHERNET NETWORK (ARP cannot cross a router)

- DNS is a little less naive, but not much - a malicious device on a network can alter DNS responses sent to others on the same network

The Bottom Line

- A malicious device on an Ethernet (including WiFi) network can abuse ARP to be come a man-in-the-middle (MiTM)
- A MiTM can read and edit all unencrypted network traffic on an Ethernet network
- A MiTM can falsify DNS responses, allowing domains to be hijacked - go to www.bankofamerica.com, get to a website run by the bad guys!

How the Internet Works

The TCP/IP Network Model in More Detail (Bonus Material)

TCP/IP is King

- In the early days of home networking, there was no defacto-standard
- The Major OS vendors had their own network protocols - DEC had DECnet, Microsoft had NetBIOS, and Apple had Apple Talk
- Thankfully we live in a world where the TCP/IP network stack won - the internet, our corporate networks, and our home networks are all TCP/IP networks
- If you understand TCP/IP you understand the internet, your work network, and your home network

It's All About Layers

- The problem TCP/IP solves is a huge one - allowing two computers anywhere in the world to exchange any amount of data

- To make that problem manageable, it was broken into four smaller problems - the four layers of the TCP/IP network stack

- Each layer is responsible for its own thing, and the only thing set in stone is the interface between the layers

The Packet is King

- TCP/IP is a packet-switching technology
- Big chunks of data are broken into small manageable pieces, and each piece, known as a packet, is sent separately
- Each packet fends for itself on the network

4 TCP/IP Layers

- 1 - Link Layer (1 Packet within the LAN)
- 2 - Internet Layer (1 Packet across the world)
- 3 - Transport Layer (a stream of data across the world)
- 4 - Application Layer (the protocols that power the apps we humans use - HTTP for the web, SMTP for email, etc.)

Layer 1 - Link

- The job of the Link Layer is to get one single packet between two devices that are directly connected to each other
- The Layer 1 protocol used on our home and corporate networks is Ethernet (either through copper wires, or over the air when we call it WiFi)
- Our cable modems or DSL modems would use a different Layer 1 protocol to connect our homes to our ISPs
- The fiberoptic cables that connect the work together would use yet another Layer 1 protocol to get one packet from one end of the fiberoptic cable to the other

Ethernet

- The Ethernet protocol uses Media Access Control addresses (MAC addresses) to identify the communicating parties
- Ethernet sends single packets (called 'Ethernet frames') from one MAC address to another
- Every network card has a unique MAC address
- The Ethernet protocol is trapped within a single network or LAN - it CANNOT cross a router

Layer 2 - Internet

- The Internet layer is responsible for moving a single packet between two devices on any set of connected networks - it is for 'internetwork' communication
- Today, our homes, offices, schools, etc. are all connected into one massive web of connected networks which we call The Internet
- The protocol used for layer 2 is IP - the Internet Protocol

The Internet Protocol (IP)

- IP routes single packets of data from one IP address to another through an interconnected web of IP networks (like The Internet)
- The work-horses that power IP are routers - devices which connect two or more networks together - forming an 'internetwork' (like The Internet)
- Routers co-operate with each other to figure out paths between IP addresses (this is HARD - we won't go into how they do it)
- An IP packet moves across the internet from router to router by crossing a single network at a time. To get across any given network, IP uses the appropriate Layer 1 protocol. A single IP packet can make use of many Layer 1 protocols as it moves through the Internet

Layer 3 - Transport

- At layers 1 and 2, each packet is independent, and worse, each packet is free to be dropped if needed - if a router is too busy, the spec says it is free to just ignore incoming packets for a bit
- Because each packet is independent, there is no guarantee that the order of packets will be preserved
- Layer 3's job is to paper over this chaos and present a coherent stream of data to the receiver
- There are two important layer 3 protocols - TCP and UPD
- Both TCP and UDP use IP to transmit their packets

TCP (Transmission Control Protocol)

- TCP is a connection-oriented protocol that guarantees that all data sent will be received, and will be reassembled in the order it was sent
- Communication over TCP starts with a hand-shake that establishes a two-way connection
- Once the connection is established, each end can send data through the connection to the other end
- TCP uses buffers to reassemble the data, putting it back into the correct order, and, requesting any missing packets be resent
- The connection establishment, resending, and buffering all require resources, and add latency

UDP (User Datagram Protocol)

- UDP is TCP's leaner but less powerful cousin
- UDP is connectionless - no setting up of a connection before data can be sent
- UDP does not resend missing packets
- UDP has much lower latency than TCP, and uses much fewer resources (no buffers for reassembling the data perfectly)
- UDP is perfect for time-critical data that is allowed to be a bit lossy - e.g. Skype!
- UDP is also good for simple single-packet request/response protocols (like DNS which we'll learn about shortly)

Layer 4 - Application

- Layer 4 contains the protocols that drive the apps we humans use
- Layer 4 protocols use TCP or UDP to transmit data streams
- There are more Layer 4 protocols than you can shake a proverbial stick at, but here are a few examples:
  - Web browsing: HTTP & HTTPS
  - Email: SMTP, IMAP & POP3
  - File Transfers: FTP (obsolete), SFTP, TFTP & Rsync
  - Remote Logins: Telnet (obsolete), SSH, VNC, ARD & RDP

MAC Addresses & IP Addresses

- We know that within our home networks, all packets move as Ethernet frames, and that Ethernet frames are sent from one MAC address to another
- But - our apps use application layer protocols that use UDP and TCP, which use IP to send packets of data between IP addresses
- How are the IP addresses mapped to the MAC addresses?
- Enter ARP - The Address Resolution Protocol

ARP - The Address Resolution Protocol

- Used by IP to figure out what MAC address a given IP can be reached at within an Ethernet network
- ARP uses Ethernet broadcasts to simply ask every device on the network if they know the MAC address for a given IP
- An Ethernet broadcast is an Ethernet frame sent to the special MAC address FF:FF:FF:FF:FF - every device on an Ethernet network listens to broadcasts
- When IP needs to know the MAC address for an IP address on the LAN, it broadcasts an ARP WHO HAS request
- Every device receives this request, and if it is configured to be that IP, it replies with an ARP response containing its MAC address

The Problem with ARP

- ARP is surprisingly simplistic - you can watch it in action with the terminal command: `sudo tcpdump -nneq arp`
- ARP has no security model - ANYONE can answer any ARP request, and all answers are believed without question
- This lack of any security at all allows for a technique called "ARP Poison Routing" or APR
- A hostile device on your Ethernet network can use APR to become a Man-in-the-middle, intercepting ALL network traffic into and out of your LAN
- The silver lining is that because ARP uses Ethernet, and because Ethernet cannot cross routers, APR can only be abused by a malicious device connected to your LAN

Names & IP Addresses

- We humans like nice memorable names - we're not good at remembering IP addresses
- TCP and UDP use IP to transmit data, and IP sends packets between IP addresses, so, both UDP and TCP send streams of data from one IP to another
- When you type http://www.bartb.ie/ into your browser, it opens an HTTP connection to my web server at IP 46.22.130.125 - how did it know to connect to that IP?
- Enter DNS - the Domain Name System

The Domain Name System (DNS)

- DNS is a hierarchical system, with servers delegating responsibility for sub-sets of the name space to other servers
- The hierarchy goes from right to left
- The root name servers delegate control of Top Level Domains (TLDs) like .com, .org, .ie, etc.
- The TLD name servers will then delegate control of sub-domains of the TLDs to name servers run by people who own domains under those TLDs
- E.g. the .ie name servers delegate control of barth.ie to my name servers, i.e. 85.233.160.78, 85.233.160.79 & 85.233.164.72

DNS Resolution

- Only server administrators need to worry about running their own name servers and registering domain names
- What we all need is DNS name resolution - the ability to turn a domain name into an IP address
- This is done by DNS resolvers. Every ISP will run a DNS resolver, and there are publicly available ones like Google's two at 8.8.8.8 and 8.8.4.4
- When you configure a DNS server on your computer or router, you are actually configuring a DNS resolver, not a DNS name server (I know - confusing right!)

DNS Resolution...

- To turn a domain name like www.bartb.ie into an IP address, a DNS resolver starts by asking the DNS root servers if they know the answer. They will not - but they do know that all .ie domains are managed by the IEDR, so they reply telling the resolver to ask one of the IEDR name servers
- The resolver then asks one of the IEDR'S name servers if they know the IP address for www.bartb.ie. They will also not know, but they will reply telling the resolver to ask one of my name servers
- Finally, the resolver will ask one of my name servers, and it will answer that www.bartb.ie is at 85.233.164.72

DNS Resolution ...

- You can see this for yourself with the terminal command:
  `dig +trace www.bartb.ie`

- You can watch the DNS queries on your network with the command:
  `sudo tcpdump udp port 53`

DNS Caching

- The only thing you need to remember about DNS resolution is complex and time-consuming - lots of "I dunno - ask that guy" before you finally get the answer
- This is why the DNS protocol allows for DNS responses to be cached heavily
- Your browser will cache DNS results, your computer will, your router will, and your resolver will

The Problem with DNS

- DNS uses UDP - a DNS request is a single UDP packet, and so is a DNS response
- Regular DNS is NOT secured - no digital signatures to validate the data - the answer is just believed
- DNS has some basic protections from spoofing in that it uses random port numbers and adds random IDs to requests, then only believes responses with the expected random ID and port number
- A Man-in-the-middle can read the random ID and random port from any DNS query, and reply with a forgery that will be believed
- Spoofed DNS entries are CACHED!
- APR + DNS spoofing and caching == BIG PROBLEMS

IP Subnets

And the difference between Routers, Switches & Wireless Access Points

An IP Subnet Is ...

- A logical network
- Every device on an IP subnet must be on the same Ethernet network
- You can have many IP Subnets within your house - and to safely deploy IoT deices - you SHOULD!
- IP addresses within a subnet are related to each other

The Big Three

- An IP subnet is defined by three things:
  - A network address (the first IP in the range of related IP addresses)
  - A subnet mask, or netmask (determines how many IP addresses will exist within the subnet)
  - At least one gateway IP address - the IP address of a router that allows IP packets to enter and exit the network
- E.g. my primary home network has a net address of 192.168.10.0 with a netmask of 255.255.255.0 and a gateway address of 192.168.10.1

Netmasks

- Netmasks have an ability to confuse people like very little else on this planet - as a result, I'm NOT going to describe them in detail - you can get that description in the Taming The Terminal series at www.bartb.ie/ttt

- Instead of describing netmasks in general, I'm going to describe the one netmask that is almost universally and exclusively used for home networks - the so-called 'Class C' netmask

Class C Networks

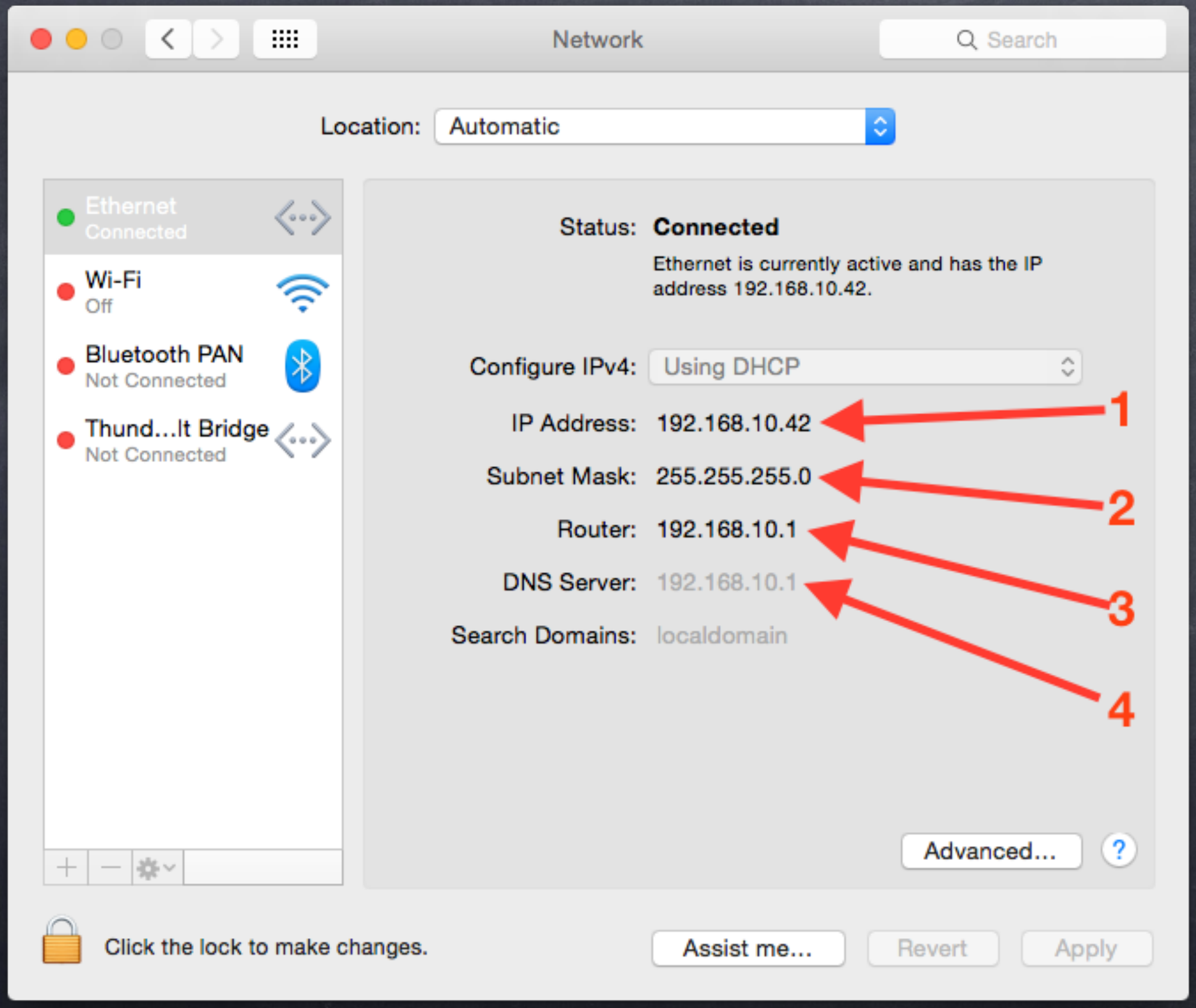
- An IP subnet with a Class C netmask is referred to as a class C network
- Netmasks can be expressed in MANY ways, all the following are different notations for the class C netmask: 255.255.255.0, /24 & fffffoo
- In a class C network, the first three parts of all IP addresses are the same, so they are often written as those first three parts, and then a *, so I could write my main network as 192.168.10.*
- The first an last IP addresses in a subnet are reserved (net address & broadcast address), so there are 254 usable IP addresses in a Class C network - last part of the IP going from 1 to 254 inclusive

IP Address Ranges

- So the Class C netmask we'll be using keeps the first three parts of the IP address the same - can I just use anything for those first three?
- Nope!
- ICANN (the people who mange the allocation of IP addresses) hand out IP addresses (and have almost run out!)
- There are a number of ranges set aside for use on private networks (like home networks)
- The two common ones are: 10.*.*.*, and 192.168.*.* - STICK TO THESE FOR YOUR HOME NETWORKS! (there are literally thousands of class C subnets to choose from within those ranges)

The Big Four

- We've already said that a network is defined by three settings - net address, netmask, and gateway address
- But those are not the setting you put into your Mac/iPhone/Play Station/...
- The following are the FOUR settings each device on your network needs:
    1. An IP address within your subnet
    2. Your network's netmask (always use class C)
    3. Your network's gateway address
    4. The IP address of a DNS resolver (your router usually)


Network
Search
Location: Automatic
Ethernet
Connected
Wi-Fi
Off
Bluetooth PAN
Not Connected
Thund...lt Bridge
Not Connected
Status: Connected
Ethernet is currently active and has the IP address 192.168.10.42.
Configure IPv4: Using DHCP
IP Address: 192.168.10.42
Subnet Mask: 255.255.255.0
Router: 192.168.10.1
DNS Server: 192.168.10.1
Search Domains: localdomain
1
2
3
4
Advanced...
?
Click the lock to make changes.
Assist me...
Revert
Apply

The Dynamic Host Configuration Protocol (DHCP)

- You COULD do all your IP configuration manually - you would just have to enter the big 4 settings into EVERY device, and you would have to keep track of what IP addresses in your range you have already assigned to devices
- But when you move to another network, none of those settings would work! BAD for portable devices!
- DHCP solves this problem - it uses broadcasts to allow devices to discover your network's settings. Home routers come with DHCP built in - use that to configure your networks

What is a Switch?

- A switch is a piece of hardware for connecting Ethernet cables to form an Ethernet network

- A switch does NOT have an IP address, nor a MAC address - it is transparent (invisible on the network)

- You can connect multiple switches together to build out your Ethernet network

What is a Wireless Access Point?

- A wireless access point is like a switch that connects radio waves to an Ethernet cable
- A wireless access point can extend your wired Ethernet network into the air as a WiFi network
- A wireless access point does not have an IP address or a MAC address - it is transparent, just like a switch

What is a Router?

- A router is a device for connecting two or more IP subnets to each other
- A router will have multiple IP and MAC addresses, one of each for each network it is connected to
- Routers are not transparent
- Ethernet broadcasts cannot cross a router

What is a Home Router?

- Devices like those we get from our ISP, or perhaps buy from Apple (AirPort Extreme/Airport Express) are CALLED routers, but that name is misleading - they are multi-function devices, and each function can be switched on or off
- Home Routers should be thought of as being all the following in a single, configurable, box:
  - A router
  - A switch
  - A wireless access point (usually)
  - A caching DNS resolver
  - A DHCP Server

Public IPs

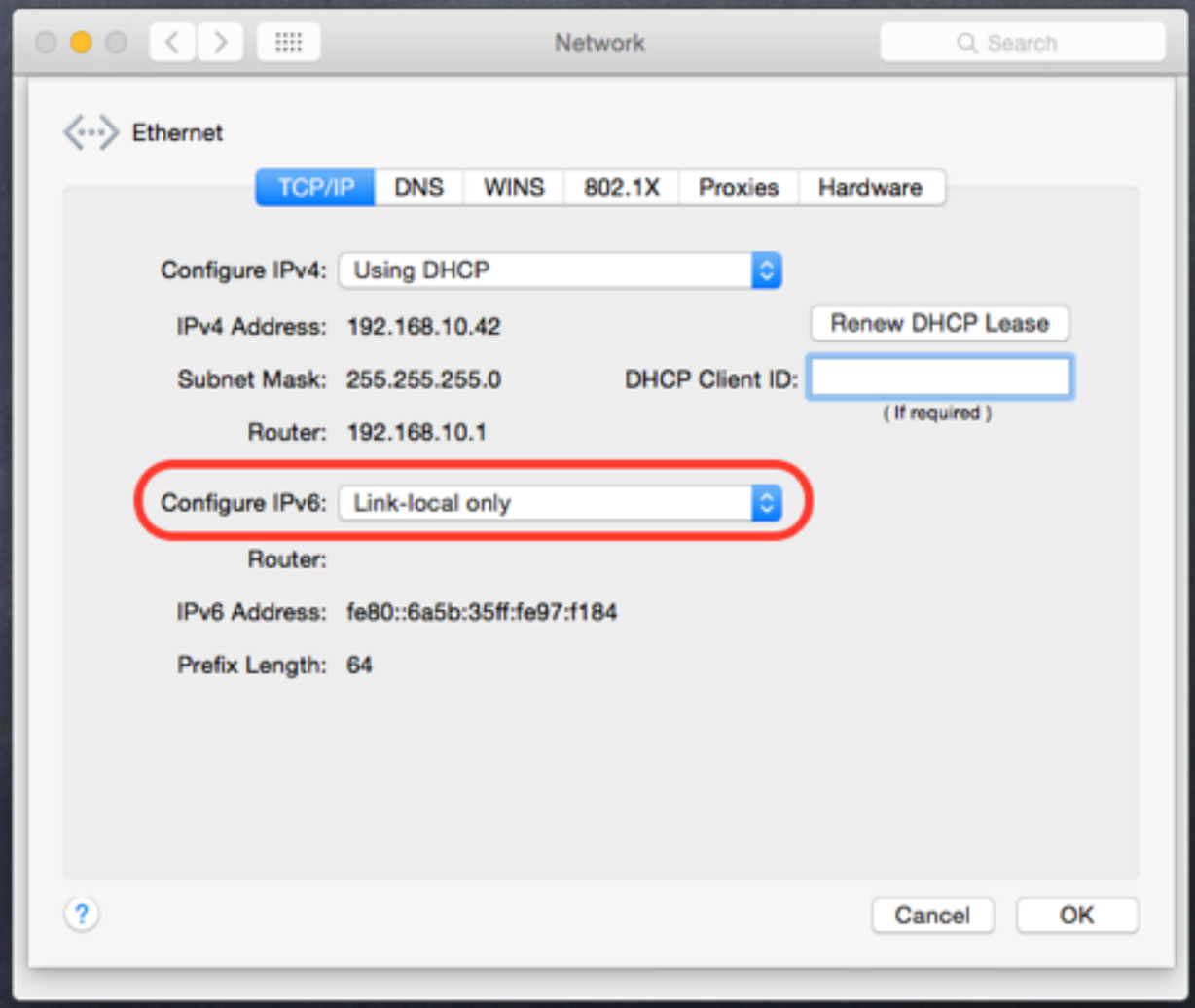
- Your ISP has bought a finite number of public IP addresses from ICANN - those are a valuable, and limited, resource that needs to be conserved
- Your ISP COULD give you lots of their expensive public IP addresses, one for each of your devices - but they do not!
- To save valuable IPs, your ISP gives you ONE of their public IP addresses, and rely on your router to share that IP with all the devices in your home
- Network Address Translation, or NAT, makes this possible

NAT

- NAT is a function that can be performed by a router as it passes traffic between networks
- A non-NAT router does not alter packets as it moves them from one network to another
- A NAT router rewrites the packets so that their 'from' address becomes the shared IP. When return packets come in, the reverse rewrite is done
- NAT needs to keep a table to remember what out-bound TCP and UDP traffic should be rewritten to which internal IP address
- NAT has the added bonus of acting like a one-way valve for network traffic - conversations can only flow if they were started by the device on the inside of the NAT - outsiders CANNOT initiate a flow of network traffic

IPv6

- We have completely ignored IPv6 in this talk, and will continue to do so
- For now, you almost certainly want to limit IPv6 to your LAN by setting it to 'link-local only'



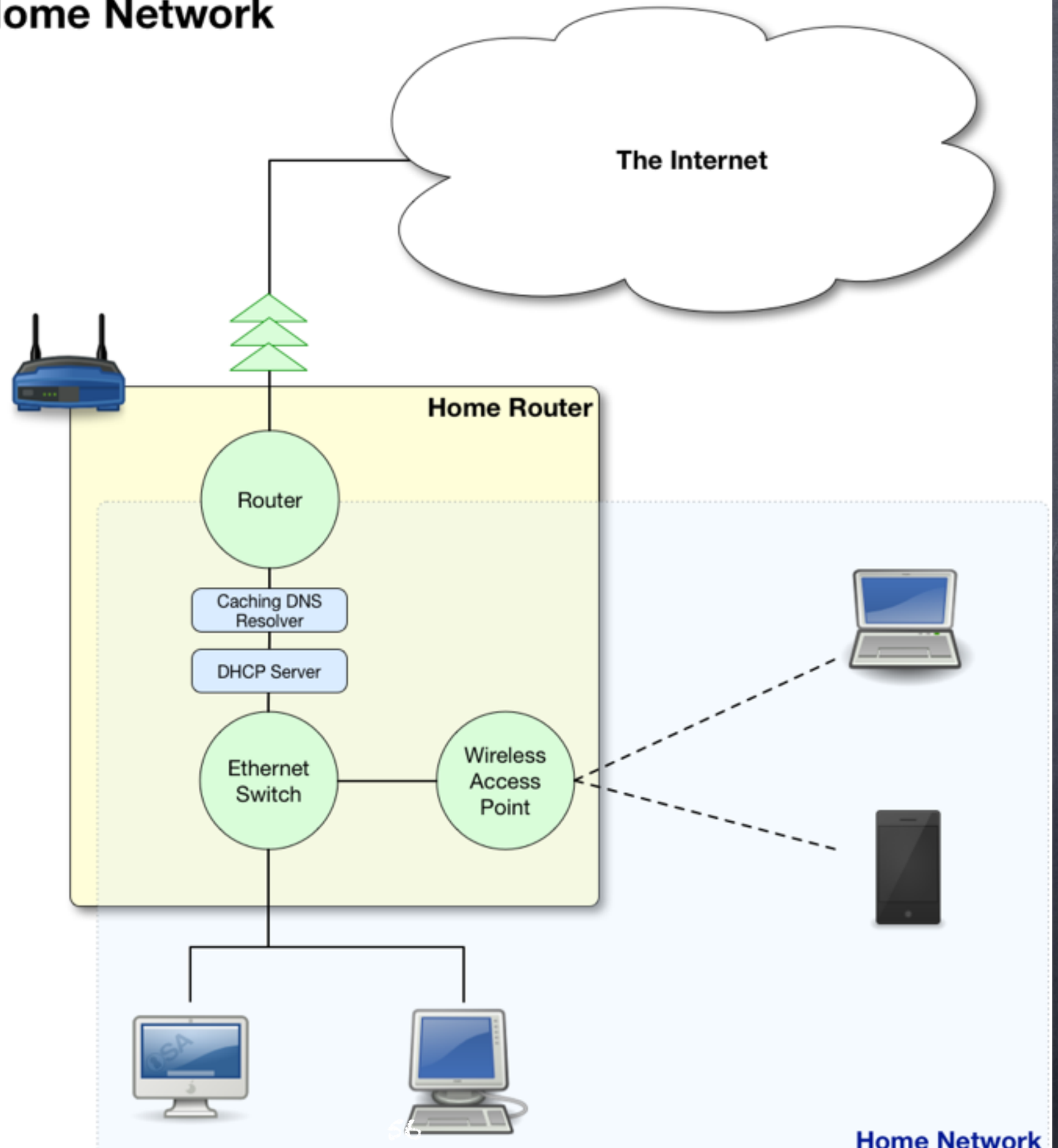
For More ...

- If you crave more details, check out episodes 23 to 35 of Taming the Terminal (http://bartb.ie/ttt)

IoT Devices & Your Home Network

What's the Problem? How do I Fix it?

A Typical Home Network



What Could Possibly Go Wrong?

- By default, your entire home network will be one large Ethernet network, used as a single large IP subnet
- Because of how ARP was designed (without any security) any devices plugged into your home network can perform a man-in-the-middle attack and READ AND ALTER ALL TRAFFIC ENTERING AND LEAVING YOUR HOUSE
- Because of how DNS was designed (without proper security) a man-in-the-middle can alter DNS responses, hence creating fraudulent mappings between human-friendly domain names like bankofamerica.com and IP addresses - enabling all kinds of dangerous attacks

What is the Solution?

- ARP is the culprit, and ARP confined to a single Ethernet network
- The two key ingredients in the solution are separation into separate networks, and NAT
- Keep your IoT devices out of your main Ethernet network where all your data is, and set up a sequence of one-way-valves (using NAT) that allows your IoT devices to access the internet, but not your computers and laptops

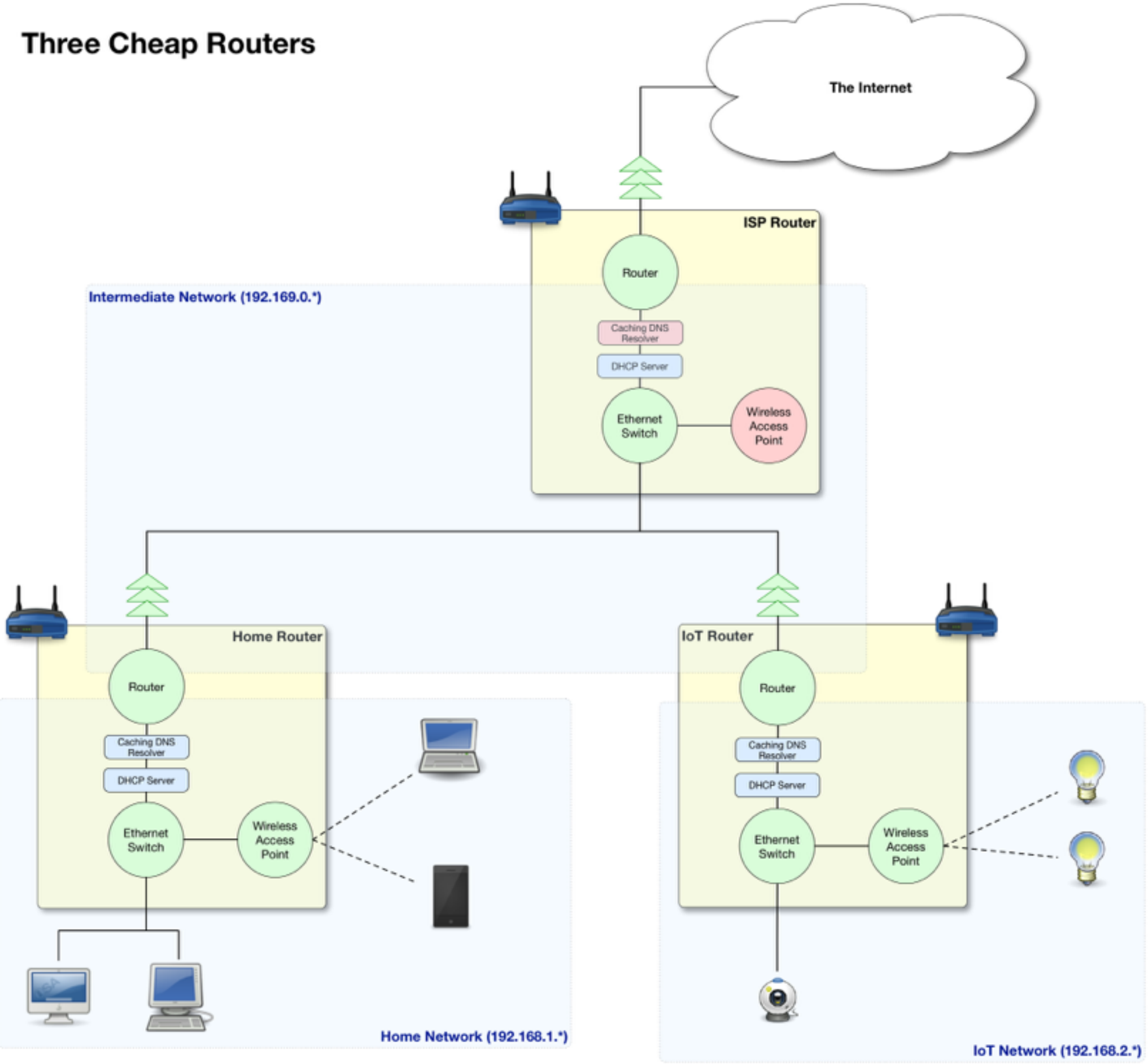
Three Cheap Routers

- If money were no object, you could solve the problem by buying a single enterprise router - enterprise routers can connect many networks together, not just two like a home router, so they can be configured to create two fully isolated networks - the problem with this solution is PRICE!
- Thankfully, you can achieve the same result with three typical home routers. You almost certainly already have one, so you just need two more, which you can easily get for under $100 total

Three Subnets to Rule Them All

- Before you start, you'll need to choose three different class C subnets within the allowed private ranges (10.*.*.* and 192.168.*.*) Pick any three, and name them "intermediate", "home", and "IoT", e.g.:
  - Intermediate: 192.168.0.*
  - Home: 192.168.1.*
  - IoT: 192.168.2.*

Three Cheap Routers



The Intermediate Network

- This network's job is to connect the three routers together - we'll call the one that connects to the Internet the ISP router, the one behind which your home network will be the Home Router, and the one behind which al your IoT devices will live the IoT Router

- This network will be controlled from the ISP router's control panel. The ISP router will be doing NAT, translating the intermediate network to the public IP assigned to you by your ISP

- You should enable DHCP on the ISP router - that will hand out IP addresses in the intermediate network - be sure to configure DHCP on this router to use your chosen subnet for the intermediate network (192.168.0.* in my example)

The Home Network

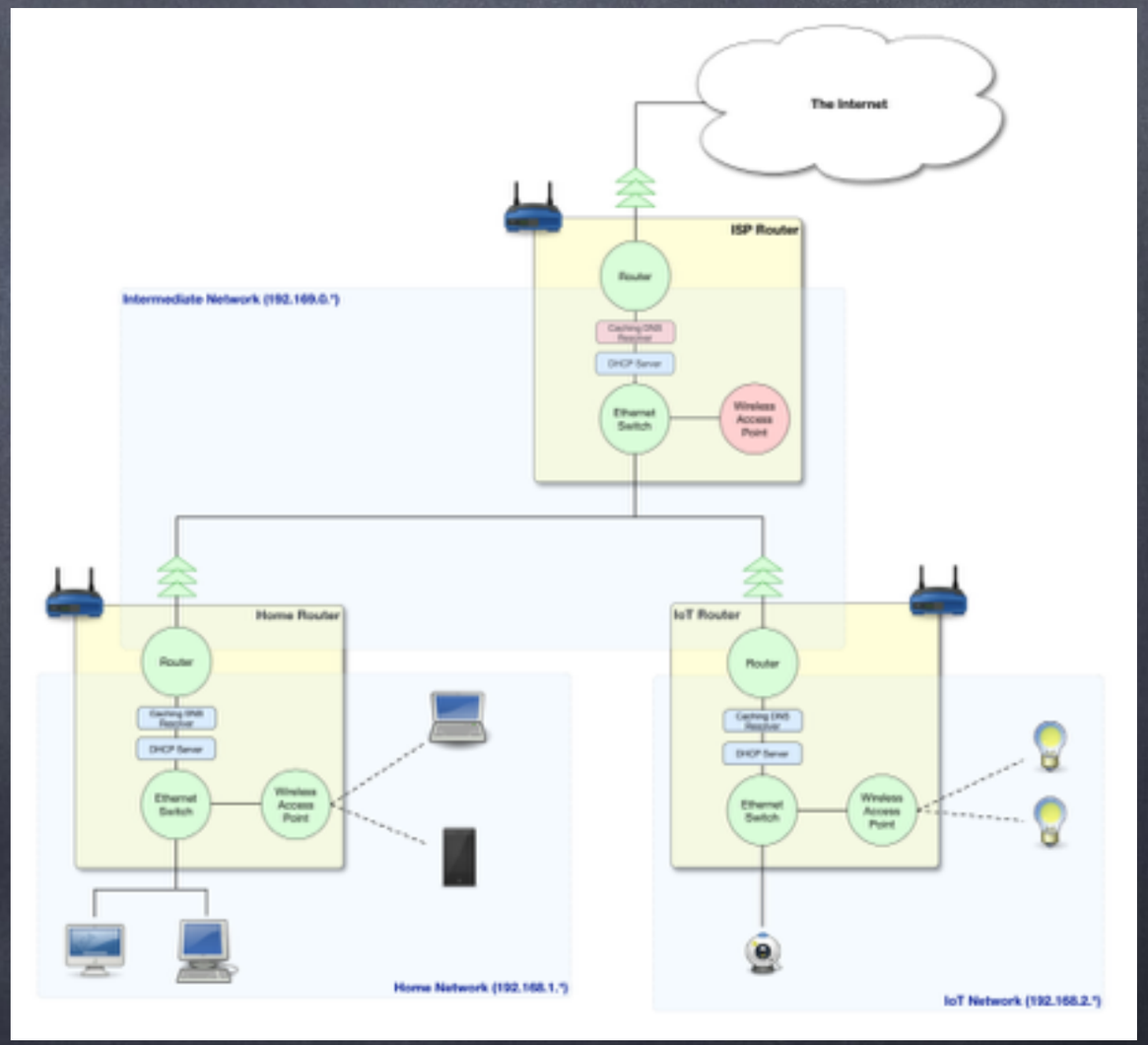
- This network should contain all your computers, tablets, phones, printers, TV boxes etc. (your current home network basically)
- This network is controlled by the Home Router - configure it as follows:
  - Make sure it is in routing mode (not bridging mode)
  - Turn on wifi and give the network a name that makes it clear that it is your home network, and not your IoT network
  - Enable DHCP, but be sure to configure it for the chosen home network range (192.168.1.* in my example)

The IoT Network

- This network should contain all your IoT devices
- This network is controlled by the IoT Router - configure it as follows:
  - Make sure it is in routing mode (not bridging mode)
  - Turn on wifi and give the network a name that makes it clear that it is your IoT network, and not your home network
  - Enable DHCP, but be sure to configure it for the chosen IoT network range (192.168.2.* in my example)

Key Points

- The IoT devices are in their own Ethernet network, so they cannot use ARP to intercept traffic from your home network
- There is a one-way valve (NAT router) between the IoT devices and your data



Some Caveats

- This setup can result in routers warning that you have "double NAT"
  - The warnings exist because in the past, double NAT was usually an indication of an accidental misconfiguration
  - We are double-NATing with intent, and for a good reason, so we can ignore
- Router auto-configuration protocols like UPnP and NATPnP will not work with this setup (a security bonus IMO)
- Port-forwarding becomes more complex (but can be done)

Questions?

The Slides are Available at:
http://bartb.ie/talk-1603-ctmac